USB Proceedings

The 20th International Conference on

Modeling Decisions for Artificial Intelligence

MDAI 2023, Umeå

Vicenç Torra, Yasuo Narukawa

Image from wikipedia.org

USB Proceedings

The 20th International Conference on

Modeling Decisions for Artificial Intelligence

MDAI 2023, Umeå, Sweden
19 - 22 June, 2023

Editors:

Vicenç Torra
Umeå University
Umeå, Sweden
E-mail: vtorra@ieee.org

Yasuo Narukawa
Department Management Science
Tamagawa University
Tokyo
Japan
E-mail: nrkwy@eng.tamagawa.ac.jp

ISBN: 978-91-527-7293-5

Preface

This volume contains papers that had to be presented at the 20th International Conference on Modeling Decisions for Artificial Intelligence (MDAI 2023) celebrated in Umeå, Sweden, 19 - 22 June, 2023. The rest of papers as well as invited papers have been separately published in the Lecture Notes in Artificial Intelligence, Vol. 13890 (by Springer).

This conference followed MDAI 2004 (Barcelona), MDAI 2005 (Tsukuba), MDAI 2006 (Tarragona), MDAI 2007 (Kitakyushu), MDAI 2008 (Sabadell), MDAI 2009 (Awaji Island), MDAI 2010 (Perpinyà), MDAI 2011 (Changsha), MDAI 2012 (Girona), MDAI 2013 (Barcelona), MDAI 2014 (Tokyo), MDAI 2015 (Skövde), MDAI 2016 (Sant Julià de Lòria), MDAI 2017 (Kitakyushu), MDAI 2018 (Mallorca), MDAI 2019 (Milano), MDAI 2020, MDAI 2021 (Umeå), and MDAI 2022 (Sant Cugat).

The aim of MDAI is to provide a forum for researchers to discuss different facets of decision processes in a broad sense. This includes model building and all kinds of mathematical tools for data aggregation, information fusion, and decision-making; tools to help make decisions related to data science problems (including, e.g., statistical and machine learning algorithms as well as data visualization tools); and algorithms for data privacy and transparency-aware methods so that data processing procedures and the decisions made from them are fair, transparent, and avoid unnecessary disclosure of sensitive information.

The MDAI conference included tracks on the topics of (a) data science, (b) machine learning, (c) data privacy, (d) aggregation functions, (e) human decision-making, and (f) graphs and (social) networks.

The conference celebrates this year the 50th anniversary of graded logic, introduced by Jozo Dujmović in a paper in 1973. In such paper, he also introduced the concept of andness, a key concept to define adjustable aggregators with a variable conjunction degree.

The conference was supported by Umeå University, the European Society for Fuzzy Logic and Technology (EUSFLAT), the Catalan Association for Artificial Intelligence (ACIA), the Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT), and the UNESCO Chair in Data Privacy.

Vicenç Torra, Yasuo Narukawa
June, 2023

General Chairs

Vicenç Torra, Umeå University

Program Chairs

Vicenç Torra, Umeå University, Sweden
Yasuo Narukawa, Tamagawa University, Japan

Advisory Board

Didier Dubois, Institut de Recherche en Informatique de Toulouse, CNRS, France
Jozo Dujmović, San Francisco State University, USA
Lluis Godo, IIIA-CSIC, Spain
Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences, Poland
Cengiz Kahraman, Istanbul Technical University, Turkey
Sadaaki Miyamoto, University of Tsukuba, Japan
Pierangela Samarati, Università degli Studi di Milano, Italy
Sandra Sandri, Instituto Nacional de Pesquisas Espaciais, Brazil
Michio Sugeno, Tokyo Institute of Technology, Japan
Ronald R. Yager, Machine Intelligence Institute, Iona Collegue, USA

Program Committee

Kayode S. Adewole, Umeå University, Sweden
Laya Aliahmadipour, Shahid Bahonar University, Iran
Cláudia Antunes, Universidade de Lisboa, Portugal
Eva Armengol, IIIA-CSIC, Spain
Edurne Barrenechea, Universidad Pública de Navarra, Spain
Gloria Bordogna, Consiglio Nazionale delle Ricerche, Italy
Humberto Bustince, Universidad Pública de Navarra, Spain
Alina Campan, North Kentucky University, USA
Francisco Chiclana, De Montfort University, UK
Susana Díaz, Universidad de Oviedo, Spain
Josep Domingo-Ferrer, Universitat Rovira i Virgili, Spain
Yasunori Endo, University of Tsukuba, Japan
Vladimir Estivill-Castro, Griffith University, Australia
Zoe Falomir, Universitat Jaume I, Spain
Javier Fernandez, Universidad Pública de Navarra, Spain
Katsushige Fujimoto, Fukushima University, Japan
Joaquin Garcia-Alfaro, Institut Mines-Télècom and Institut Polytechnique de Paris, France

Michel Grabisch, Université Paris I Panthéon-Sorbonne, France
Yukihiro Hamasuna, Kindai University, Japan
Tove Helldin, University of Skövde, Sweden
Enrique Herrera-Viedma, Universidad de Granada, Spain
Aoi Honda, Kyushu Institute of Technology, Japan
Van-Nam Huynh, JAIST, Japan
Masahiro Inuiguchi, Osaka University, Japan
Simon James, Deakin University, Australia
Aránzazu Jurío, Universidad Pública de Navarra, Spain
Yuchi Kanzawa, Shibaura Institute of Technology, Japan
Ali Karaşan, Yildiz Technical University, Turkey
Hiroaki Kikuchi, Meiji University, Japan
Petr Krajča, Palacky University Olomouc, Czech Republic
Marie-Jeanne Lesot, Université Pierre et Marie Curie (Paris VI), France
Giovanni Livraga, Università degli Studi di Milano, Italy
Jun Long, National University of Defense Technology, China
Beatriz López, University of Girona, Catalonia, Spain
Jean-Luc Marichal, University of Luxembourg, Luxembourg
Radko Mesiar, Slovak University of Technology, Slovakia
Andrea Mesiarová-Zemánková, Slovak Academy of Sciences, Slovakia
Anna Monreale, University of Pisa, Italy
Pranab K. Muhuri, South Asian University, India
Toshiaki Murofushi, Tokyo Institute of Technology, Japan
Guillermo Navarro-Arribas, Universitat Autònoma de Barcelona, Spain
Shekhar Negi, Umeå University, Sweden
Jordi Nin, Esade, Universitat Ramon Llull, Spain
Miguel Nunez-del-Prado, Universidad del Pacífico, Peru
Anna Oganyan, National Institute of Statistical Sciences (NISS), USA
Gabriella Pasi, Università di Milano Bicocca, Italy
Oriol Pujol, University of Barcelona, Catalonia, Spain
Maria Riveiro, Jönköping University, Sweden
Julian Salas, Universitat Oberta de Catalunya, Catalonia, Spain
Robyn Schimmer, Umeå University, Sweden
H. Joe Steinhauer, University of Skövde, Sweden
László Szilágyi, Sapientia-Hungarian Science University of Transylvania, Hungary
Aida Valls, Universitat Rovira i Virgili, Spain
Paolo Viappiani, Université Paris Dauphine, France
Zeshui Xu, Southeast University, China

Local Organizing Committee Chair

Vicenç Torra, Umeå University, Sweden

Additional Referees

Sergio Martinez Lluis, Najeeb Moharram Salim Jebreel, Rami Haffar

Supporting Institutions

Umeå University
The European Society for Fuzzy Logic and Technology (EUSFLAT)
The Catalan Association for Artificial Intelligence (ACIA)
The Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT)
The UNESCO Chair in Data Privacy

Table of Contents

Regular Papers

Hand Pose Recognition through MediaPipe Landmarks

Manuel Gil-Martín[1[0000-0002-4285-6224]], Rubén San-Segundo[1[0000-0001-9659-5464]] and Ricardo de Córdoba[1[0000-0002-7136-9636]]

[1] Speech Technology and Machine Learning Group (T.H.A.U. Group),
Information Processing and Telecommunications Center, E.T.S.I. de Telecomunicación,
Universidad Politécnica de Madrid, 28040, Madrid, Spain

Abstract. This paper proposes a framework to recognize hand poses using a limited number of landmarks from images. This Hand Pose Recognition (HPR) system is composed of a signal processing module that extracts and processes the coordinates of specific points of the hand called landmarks, and a deep neural network module that models and classifies the hand poses. These specific points or landmarks are extracted automatically through MediaPipe software. Detecting hand poses from these points has two main advantages compared to traditional computer vision approaches: the information sent to the recognition module is smaller (points' coordinates vs. a full image) and the classification is not affected by additional information included in the images (like the background). The experiments were carried out over two different datasets using the experimental setups of previous works. The proposed framework was able to obtain better performance than the best results reported in previous works. For example, in case of using the Tiny Hand Gesture Recognition Dataset, we obtained classification accuracies of 98.74 ± 0.08 % and 98.22 ± 0.06 % with simple or complex backgrounds, while the best reported accuracies in previous works (using the whole image) were 97.10 % and 85.30 % respectively. The proposed solution is able to provide high recognition performance independently of the background where the image is taken.

Keywords: Hand Pose Recognition, MediaPipe, Hand landmarks, Deep learning, Convolutional Neural Networks.

1 Introduction

Hand Pose Recognition consists in detecting the posture or pose that people perform using their hands. This technology could be useful to develop human computer interaction systems and could improve the user experience across a wide variety of different domains. For example, it could be seen as the basis for sign language understanding and hand gesture control applications. For instance, a person could ask for taking a picture using the front camera of a smartphone by opening and closing the hand palm. In these applications, it is crucial to accurately recognize the hand pose or gesture to perform specific actions with smart devices, a computer or an automatic transmission machine. In this context, computer vision based approaches have been applied reaching promising results.

However, computer vision based approaches are usually based on feeding the systems by raw images that include sensitive information like the face or the background that people would like not to share. In addition, these images could have large sizes and cause strain on bandwidth in real applications. This way, it could be great to study solutions that extract the strictly necessary information from the images in order to develop lighter systems that could respect the individual's privacy.

This paper aims to propose a framework to detect hand poses using a limited number of landmarks from images. The main contributions of the paper are:

- The proposal of a framework to detect hand poses from specific points of images that is not affected by the background of the images nor the people who perform the pose.
- The evaluation of the proposal using two datasets and a comparison to previous works that used the whole images as input using the same experimental setups.

This paper is organized as follows. Section 2 reviews the related work on hand pose recognition. Section 3 describes the material and methods used, including the datasets, the system architecture including the signal processing and deep learning approaches and the evaluation details. Section 4 discusses the experiments and the obtained results. Finally, Section 6 summarizes the main conclusions of the paper.

2 Related work

Multiple previous works have been focused on Human Activity Recognition in order to optimize the physical activity classification using wearables or cameras [1-4] that could be traditionally applied to sports monitoring purposes [5-7] such as fitness tracking, personal incentivizing, or rehabilitation [8]. However, there exist a lower number of works focused on detecting hand poses or gestures. Most of these works use images as inputs of their systems and follow a hand localization step as first stage. Afterwards, they extracted handcrafted features or descriptors [9] from the hand and feed an inference algorithm that classifies the different hand poses. As mentioned in the introduction, most hand detection systems are based on computer vision approaches [10-12] which often use raw images.

For example, Wang et al. [13] developed a hand pose recognition system where they first obtained a segmented hand map using Kinect software development kit, then they extracted a volumetric shape descriptor using the line between the center of hand and wrist as polar axis of polar coordinates and finally they used a Support Vector Machines classifier to perform hand pose recognition.

Another previous work [14] proposed a Convolutional Neural Network based system to model ten different hand poses from ten different people. They pre-filtered the images using a Gabor filter and used skin color distribution as descriptor of the hand

to feed the deep learning architecture. They obtained an accuracy of 97% in the person-independent test.

In the same way, a previous work [15] segmented the image into the hand in different regions and obtained the Histogram of Oriented Gradients and a Local Binary Pattern from each region. Afterwards, they combined k-means and Support Vector Machines in order to classify the hand poses, obtaining an F1 score near 96% using data from 25 subjects and 16 different hand poses.

Similarly a previous work [16] feed a deep Convolutional Neural Network to directly classify hand poses in images without any previous segmentation. They classified the hand pose with average accuracy of 97.1% in the images with simple backgrounds and 85.3% in the images with complex backgrounds. They used a dataset with data from 40 subjects and seven different hand poses.

To summarize, existing methods combined a hand segmentation step with a handcrafted features extraction step to obtain a descriptive hand pose representation. Afterwards, they fed a machine learning module to model and classify the hand poses. The aim of this work is to use a powerful and effective library to directly extract representative relevant points from the hand images (landmarks) to model and classify hand poses through a deep learning solution. In this sense, it is hypothesized that hand pose recognition task could be performed by the combination of landmarks extraction and a deep learning architecture offering higher performance without handcrafted methods to process the input images.

3 Material and methods

This section includes information about the datasets used in this work, the proposed system architecture including the signal processing and the deep neural network and the evaluation of the system.

3.1 Datasets

For this work, we have used two publicly available hand pose datasets: Multi-modal Leap Motion dataset for Hand Gesture Recognition [15] and Tiny Hand Gesture Recognition Dataset [16].

Multi-modal Leap Motion dataset for Hand Gesture Recognition includes data from 25 subjects (8 women and 17 men) that performed 16 different hand poses. Each subject was placed in front of a computer with the Leap Motion located on a table between the subject and the computer for image collection. Each subject was free to move the right hand over the device inside the Leap Motion field of view. The hand poses included in this dataset are: L, fist moved, index, ok, C, heavy, hang, two, three, four, five, palm, down, palm moved, palm up, and up. This dataset contains a total number of frames of 65,156 related to hand poses.

Tiny Hand Gesture Recognition Dataset contains data from 40 subjects (14 women and 26 men) that performed seven different hand poses. Half of the subjects were recorded with gray simple background and the rest with complex background. The considered complex backgrounds are highly cluttered and the illumination undergoes large variations. The hand poses included in this dataset are: fist, L, ok, palm, pointer, thumb down and thumb up. This dataset contains a total number of frames of 260,796.

3.2 System architecture

Fig 1 shows a diagram module of the system: a data acquisition step where the images are collected, a signal processing module where landmarks are extracted from the images and processed, and a deep learning network to model and classify the hand poses.

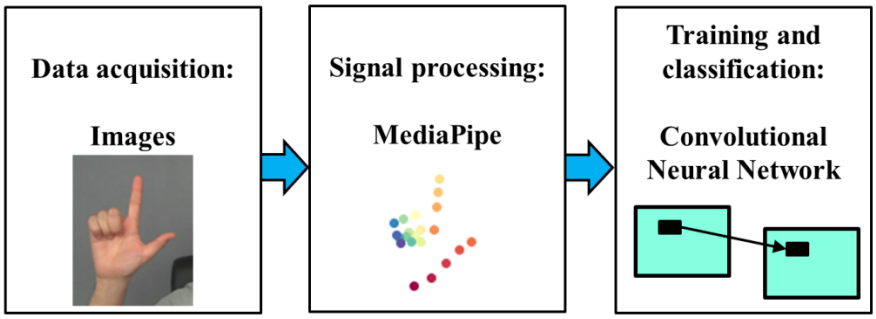


Fig 1. System architecture for hand poses recognition using MediaPipe.

Signal processing module

MediaPipe [17] is a library with the capacity to track hands from input frames or video streams. This framework offers a wide variety of solutions, such as face detection, face mesh, hair segmentation, object detection or pose and hands tracking. In particular, we used the MediaPipe Hands software to extract x and y coordinates of 21 landmarks from the hand. These coordinates are normalized to [0.0, 1.0] interval by the image width and height respectively. The 21 landmarks correspond to different location of the hand area: wrist and four points along the five fingers. Fig 2 shows the landmarks of different hand poses in the datasets used in this work.

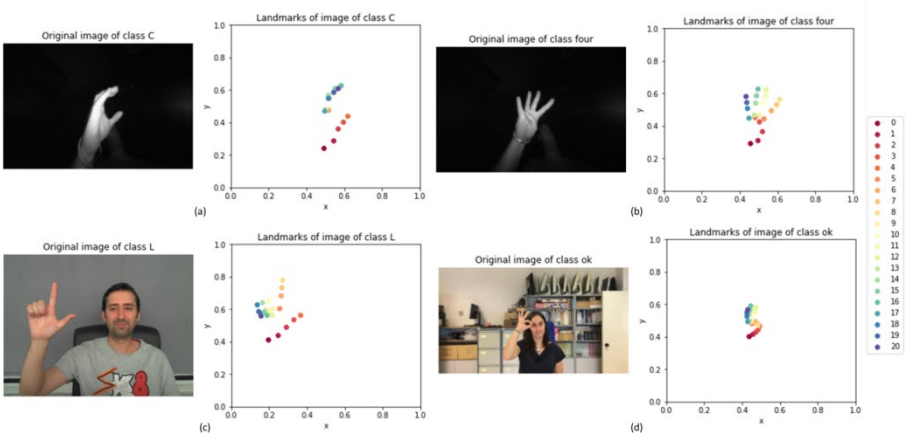


Fig 2. Original images and landmarks of different examples of the datasets used in this work (a) and (b) from Multi-modal Leap Motion Dataset and (c) and (d) from Tiny Hand Gesture Recognition Dataset.

The proposed framework extracts the landmarks from the images using the MediaPipe library. After obtaining the landmarks, a specific normalization of the coordinates is applied in order to help the neural network to model the hand poses. This normalization consists in using the lower landmark of the palm (wrist) as reference and subtracting their coordinates to the rest of landmarks. **Fig 3** shows the original and normalized landmarks of an example of class four, where it is possible to observe that the reference of the different poses becomes the coordinate origin instead of the wrist landmark.

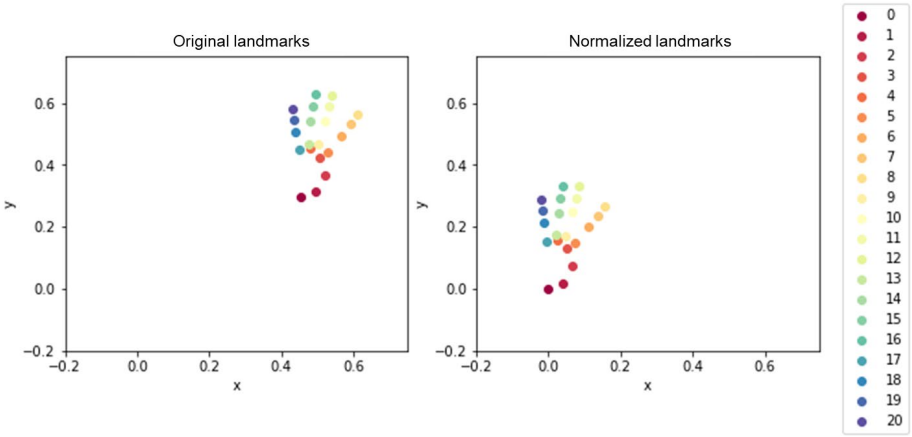


Fig 3. Original and normalized landmarks of an example.

Deep learning approach

The deep learning architecture used in this work was composed of two main parts: a feature learning subnet and a classification subnet. The first subnet learnt features from the x and y coordinates of the different landmarks, using two convolutional layers. The second subnet used fully connected layers to classify the learned features as a predicted hand pose. The architecture included dropout layers (0.3) after convolutional and and fully connected layers to avoid overfitting during training. The last layer used a softmax activation function to offer the predictions of each class for every analysis frame, while intermediate layers used ReLU for reducing the impact of gradient vanishing effect. We used categorical cross-entropy as loss metric and the root-mean-square propagation method as optimizer. We adjusted the epochs and batch size of the deep learning structure for each dataset: 300 and 500 for the Multi-modal Leap Motion dataset for Hand Gesture Recognition and 5 and 500 for the Tiny Hand Gesture Recognition Dataset. The difference between the numbers of epochs in each configuration is related of the number of examples to train the network, which is higher in the second dataset. Fig 4 represents the architecture used in this work to model and classify the hand poses of the datasets, where C indicates the number of recognized hand poses.

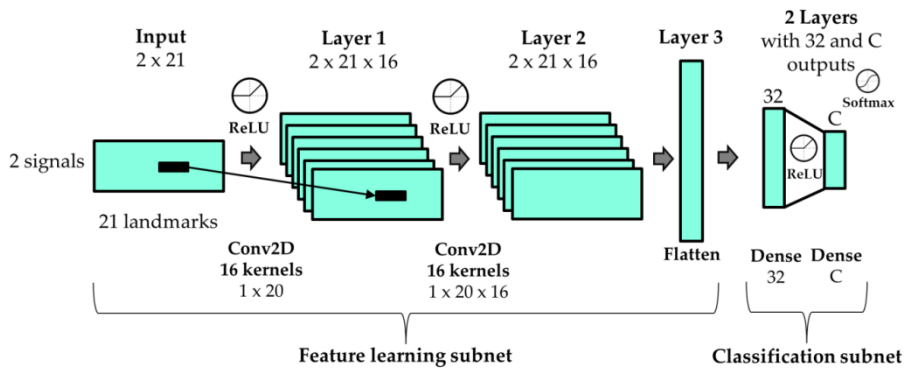


Fig 4. Convolutional Neural Network Architecture used in this work for all the datasets.

3.3 Evaluation setup

In this work, we considered the data distributions of the previous works: specific train and test subsets for Multi-modal Leap Motion dataset for Hand Gesture Recognition and a cross-validation strategy for Tiny Hand Gesture Recognition Dataset.

In case of training and testing subset, data from the same subject were included in both subsets. This methodology provided an optimistic scenario where the system was evaluated with recordings from subjects who were processed during the training step. This methodology was used for the Multi-modal Leap Motion dataset for Hand Gesture Recognition to follow the same experimental setup of a previous work [15] using

this dataset, where the training subset contained 48,436 frames and the testing subset contained 16,720 frames.

In case of the cross-validation experimental setup, 25 people were used for training, 5 subjects for validation, and 10 people for testing. In these experiments, it was assured that all the recordings from the same subject are included only in a subset. Once the system model is fitted on the training subset, the validation subset was used for optimizing the model hyperparameters. Finally, the system was evaluated with the testing subset. This process was repeated several times leaving different subjects for testing in each iteration. The results were averaged along all trials. This methodology simulated a difficult scenario because the system was evaluated with recordings from subjects different to those used for training. This methodology was used for the Tiny Hand Gesture Recognition Dataset to follow the same experimental setup of a previous work [16] using this dataset.

As evaluation metrics, we used accuracy, which is defined as the ratio between the number of correctly classified samples and the number of total samples. Considering a classification problem with N testing samples and C classes, accuracy is defined in Equation (1).

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{C} P_{ii} \tag{1}$$

Considering R_i as the sum of all examples in a column of the confusion matrix, and S_i as the sum of all examples in a row, precision, recall and F1 score metrics are defined as follows:

$$\text{precision} = \frac{1}{C}\sum_{i=1}^{C} \frac{P_{ii}}{R_i} \tag{2}$$

$$\text{recall} = \frac{1}{C}\sum_{i=1}^{C} \frac{P_{ii}}{S_i} \tag{3}$$

$$\text{F1 score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

Confidence intervals are used to show statistical significance values and provide confidence about the results reliability. These intervals include plausible values for a specific metric. We will assure that there exists a significant difference between results of two experiments when their confidence intervals do not overlap. Equation (5) represents the computation of confidence intervals attached to a specific metric value and N samples when the confidence level is 95%.

$$\mathrm{CI}(95\%) = \pm 1.96 \sqrt{\frac{\text{metric} \cdot (100 - \text{metric})}{\mathrm{N}}} \tag{5}$$

4 Experiments and Discussion

We firstly analyzed the effect of normalizing the coordinates using the lower landmark of the palm (wrist) as reference and subtracting their coordinates to the rest of landmarks. In this sense, the reference of the different poses becomes the coordinate origin. We observed that we could increase the recognition accuracy from 96.45 ± 0.28 % to 97.25 ± 0.25 % for Multi-modal Leap Motion dataset for Hand Gesture Recognition and from 98.52 ± 0.09 % to 98.74 ± 0.08 % for simple backgrounds of Tiny Hand Gesture Recognition Dataset. This normalization offers a slight increment of performance. However, it is fair to say that this improvement of performance is significant even in the difficult situation when performance is high. One of the reasons of this improvement is that thanks to this normalization, the representation of examples of the same pose become similar independently of the location of the hand in the image. For example, the representation of a hand pose consisting in pointing a screen with one finger could differ when the person performs the pose at right of left side of the image. However, thanks to normalizing using the wrist landmark, both representations become similar since both use the coordinate origin as reference. Regarding computational cost, the normalization does not heavily increase the processing time thanks to simple operations that are used.

Second, we compared our solution to previous works using the same datasets and their data distributions. A previous work [15] using the Multi-modal Leap Motion dataset for Hand Gesture Recognition segmented the image into the hand in different regions and obtained the Histogram of Oriented Gradients and a Local Binary Pattern from each region. Afterwards, the system combined k-means and Support Vector Machines in order to classify the hand poses, obtaining an F1 score near 96% using specific training and testing subsets. Another previous work [16] used the Tiny Hand Gesture Dataset and increased the number of samples by performing synthetic translations over the whole images, reaching a total number of 500,000 hand gesture samples for training. This system used a deep convolutional neural network (composed by 9 convolutional layers, 4 pooling layers, 3 fully connected layers, interlaced with ReLU and dropout layers) to directly classify hand poses in images without any previous segmentation. This previous work classified the hand pose with average accuracy of 97.1% in the images with simple backgrounds and 85.3% in the images with complex backgrounds following a cross-validation experimental setup, where 25 people were used for training, 5 subjects for validation, and 10 people for testing. Table 1 includes the comparison of these previous works and our work using the mentioned normalization of landmarks.

Table 1. Results considering baseline experimental setups for the datasets.

Dataset	Work	Accuracy (%)	F1 score (%)
Multi-modal Leap Motion Dataset for Hand Gesture Recognition	[15]	-	96.00
	This work	97.25 ± 0.25	97.23 ± 0.25
Tiny Hand Gesture Recognition Dataset – Simple background	[16]	97.10	-
	This work	98.74 ± 0.08	98.74 ± 0.08
Tiny Hand Gesture Recognition Dataset – Complex background	[16]	85.30	-
	This work	98.22 ± 0.06	98.23 ± 0.06

As observed, we obtained better performance than these previous works. One of the interesting aspects of using our approach is that the system does not suffer a decrement of performance when dealing with complex backgrounds. One of the reasons is that once the landmarks are extracted, the process is the same independently of the context and background where the image was taken. Additionally, the system can process images of different dimensions: in previous works, the deep neural architecture that directly processes the images requires a specific input image dimensions and uses specific convolutional kernel sizes to learn relevant features from them. Nevertheless, as the extraction of landmarks does not depend on the neural architecture in the proposed approach, it is not restricted to specific image dimensions.

However, the proposed approach could have some limitations that should be address in future works: managing blurred images or images without complete hands. In these situations, the MediaPipe tool can have problems extracting the landmarks.

5 Conclusions

This paper proposes an alternative framework to detect hand poses using a limited number of landmarks from images. This approach for Hand Pose Recognition automatically extracts 21 MediaPipe landmarks (x and y coordinates of specific points) from the hand and feeds a deep neural architecture to model and recognize different hand poses. This solution obtained better results than previous works using the same datasets. For example, in case of using the Tiny Hand Gesture Recognition Dataset, classification accuracies of 98.74 ± 0.08 % and 98.22 ± 0.06 % with simple or complex backgrounds, respectively. Moreover, detecting hand poses from these points or landmarks has two main advantages compared to traditional computer vision approaches: the information sent to the recognition module is smaller (coordinates from the points vs. a full image) and the classification is not affected by additional information included in the images (like the background). In this sense, the proposed system could offer a high performance without handcrafted methods to process the input images.

As future work, it would be interesting to recognize gestures as a sequence of frames using landmarks, study the effect of changing the brightness and contrast of

the recorded images before extracting the landmarks, study the effect of including a non-gesture class to distinguish when the system is not able to extract landmarks, detect which hand is processed to avoid errors when both hands appear in the image, perform the hand pose recognition using a Leave One Subject Out Cross Validation methodology and automatically compute the remaining landmarks when only a part of the hand appears in the original image. In addition, it would be interesting to apply this framework for other datasets with wider variety of backgrounds and/or related to sign language recognition with a higher number of classes.

Acknowledgements

The work leading to these results was supported by the Spanish Ministry of Science and Innovation through the project AMIC-PoC, and BeWord (PDC2021-120846-C42 and PID2021-126061OB-C43, funded by MCIN/AEI/10.13039/501100011033 and by the European Union "NextGenerationEU/PRTR"). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

1. M. Gil-Martin, R. San-Segundo, F. Fernandez-Martinez, and J. Ferreiros-Lopez, "Time Analysis in Human Activity Recognition," *Neural Processing Letters,* 2021, doi: 10.1007/s11063-021-10611-w.
2. M. Gil-Martín, R. San-Segundo, F. Fernández-Martínez, and R. de Córdoba, "Human activity recognition adapted to the type of movement," *Computers & Electrical Engineering,* vol. 88, p. 106822, 2020/12/01/ 2020, doi: https://doi.org/10.1016/j.compeleceng.2020.106822.
3. M. Gil-Martin, R. San-Segundo, F. Fernandez-Martinez, and J. Ferreiros-Lopez, "Improving physical activity recognition using a new deep learning architecture and post-processing techniques," *Engineering Applications of Artificial Intelligence,* vol. 92, Jun 2020, Art no. 103679, doi: 10.1016/j.engappai.2020.103679.
4. S. Zhang, Z. Wei, J. Nie, L. Huang, S. Wang, and Z. Li, "A Review on Human Activity Recognition Using Vision-Based Method," *Journal of Healthcare Engineering,* vol. 2017, 2017 2017, Art no. 3090343, doi: 10.1155/2017/3090343.
5. Y. Hsu, S. Yang, H. Chang, and H. Lai, "Human Daily and Sport Activity Recognition Using a Wearable Inertial Sensor Network," *IEEE Access,* vol. 6, pp. 31715-31728, 2018, doi: 10.1109/ACCESS.2018.2839766.
6. Z. Zhuang and Y. Xue, "Sport-Related Human Activity Detection and Recognition Using a Smartwatch," *Sensors,* vol. 19, no. 22, Nov 2019, Art no. 5001, doi: 10.3390/s19225001.
7. D. Hendry, K. Chai, A. Campbell, L. Hopper, P. O'Sullivan, and L. Straker, "Development of a Human Activity Recognition System for Ballet Tasks," *Sports Medicine - Open,* vol. 6, no. 1, p. 10, 2020/02/07 2020, doi: 10.1186/s40798-020-0237-5.

8. M. Gil-Martin, W. Johnston, R. San-Segundo, and B. Caulfield, "Scoring Performance on the Y-Balance Test Using a Deep Learning Approach," *Sensors,* vol. 21, no. 21, pp. 7110-7110, Nov 2021.
9. P. Trindade, J. Lobo, and J. P. Barreto, "Hand gesture recognition using color and depth images enhanced with hand angular pose data," in *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 13-15 Sept. 2012 2012, pp. 71-76, doi: 10.1109/MFI.2012.6343032.
10. M. Oudah, A. Al-Naji, and J. Chahl, "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques," *Journal of Imaging,* vol. 6, no. 8, p. 73, 2020.
11. G. Zhang, L. Wang, L. Wang, and Z. Chen, "Hand-raising gesture detection in classroom with spatial context augmentation and dilated convolution," *Computers & Graphics,* vol. 110, pp. 151-161, 2023/02/01/ 2023, doi: https://doi.org/10.1016/j.cag.2022.11.009.
12. P. Trigueiros, F. Ribeiro, and L. P. Reis, "Hand Gesture Recognition System Based in Computer Vision and Machine Learning," in *Developments in Medical Image Processing and Computational Vision*, J. M. R. S. Tavares and R. Natal Jorge Eds. Cham: Springer International Publishing, 2015, pp. 355-377.
13. Y. Wang, R. Yang, and Ieee, "Real-Time Hand Posture Recognition based on Hand Dominant Line using Kinect," in *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, San Jose, CA, 2013, Jul 15-19 2013, in IEEE International Conference on Multimedia and Expo Workshops, 2013. [Online]. Available: <Go to ISI>://WOS:000335245800022
14. D. Núñez Fernández and B. Kwolek, "Hand Posture Recognition Using Convolutional Neural Network," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Cham, M. Mendoza and S. Velastín, Eds., 2018// 2018: Springer International Publishing, pp. 441-449.
15. T. Mantecon, C. R. del-Blanco, F. Jaureguizar, and N. Garcia, "A real-time gesture recognition system using near-infrared imagery," *Plos One,* vol. 14, no. 10, Oct 3 2019, Art no. e0223320, doi: 10.1371/journal.pone.0223320.
16. P. Bao, A. I. Maqueda, C. R. del-Blanco, and N. Garcia, "Tiny Hand Gesture Recognition without Localization via a Deep Convolutional Network," *Ieee Transactions on Consumer Electronics,* vol. 63, no. 3, pp. 251-257, Aug 2017, doi: 10.1109/tce.2017.014971.
17. C. Lugaresi *et al.*, "MediaPipe: A Framework for Building Perception Pipelines," *ArXiv,* vol. abs/1906.08172, 2019.

Data Augmentation For Small Object using Fast AutoAugment

DaeEun Yoon[1,2[0000−0003−2299−191X]], Semin Kim[1,2[0000−0003−3746−0863]], SangWook Yoo[1,2[0000−0002−5404−4397]], and Jongha Lee[1,2[0000−0002−1568−6733]]

[1] AI R&D Center of Lululab Inc., Seoul, Republic of Korea
[2] {de.yoon, sm.kim, sangwook.yoo, jongha.lee}@lulu-lab.com

Abstract. In recent years, there has been tremendous progress in object detection performance. However, despite these advances, the detection performance for small objects is significantly inferior to that of large objects. Detecting small objects is one of the most challenging and important problems in computer vision. To improve the detection performance for small objects, we propose an optimal data augmentation method using Fast AutoAugment. Through our proposed method, we can quickly find optimal augmentation policies that can overcome degradation when detecting small objects, and we achieve a 20% performance improvement on the DOTA dataset.

Keywords: Object Detection · Small Object · Optimal Data Augmentation

1 Introduction

Through the recent development of deep learning technology, various computer vision tasks have been solved and studied. Among them, object detection is a very important task in computer vision. Object detection has been applied in many areas, including robot vision, autonomous vehicles, satellite image analysis, and medical image analysis, and there have been many advances. However, despite these advances, the problem of detecting small objects has emerged. As shown in Fig. 1, detecting a small object tends to be more difficult than detecting large object or medium object. even in the top submission for the MS COCO [1] Object Detection challenge, the performance of detecting small objects is significantly lower than that of detecting large objects. However, detecting small objects is often a more critical task than detecting large objects. For example, if a small forest fire is detected on a real-time mountain CCTV, the spread of a large forest fire can be prevented early, and in the case of self-driving cars, small pedestrians or traffic signs must be detected. Satellite data taken at high altitude should detect small objects, and small defects and objects should be detected in image analysis automation equipment at industrial sites. Moreover, medical images should be able to detect small-sized malignant tumors. Thus, objects

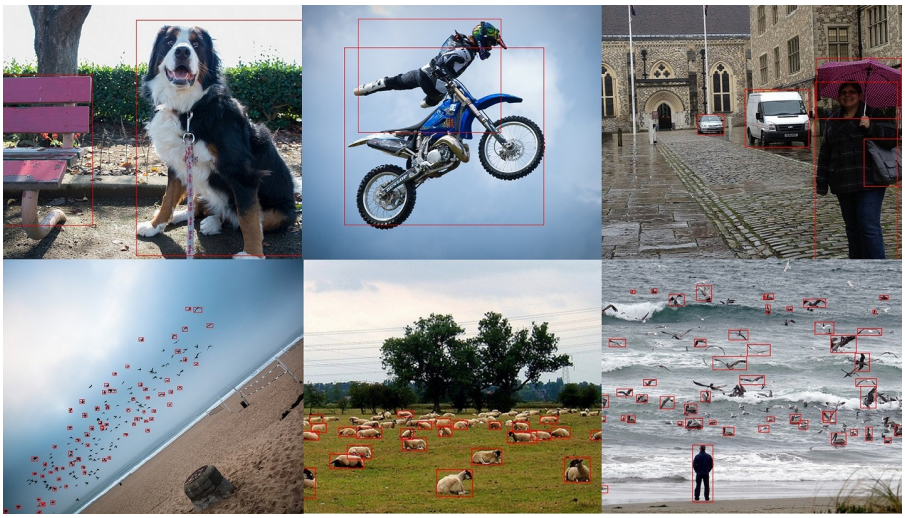


Fig. 1. An inference sample of Faster R-CNN in MS COCO. The first row is an image consisting of a large object and a medium object instance, and the second row is an image consisting of a small object instance. Despite its clear visibility, small objects have lower detection performance compared to large object detection performance.

that should be detected in the real world are often represented by small pixels in the image. In this paper, there are three perspectives on the degradation of small object detection performance. First, the area of a small object pixel differs from a large object by several times to several tens of times. This data imbalance problem can cause object detection models to be biased towards large objects during training. Second, most data augmentation methods are not effective on small objects. Data augmentation can create models that prevent overfitting and improve generalization performance by adding diverse distributions to training datasets, and consequently contribute significantly to improving the performance of the models. Various augmentation techniques have also been studied in object detection. Pixel-Level transform authorization, which changes pixel values such as RGBSshift, Blur, Random Contrast, and Random Brightness, and geometry transform authorization such as Flip, Shift, and Rotate improve classification or large object detection performance, but not small object detection performance. RandomErase [2] and Cutout [3] erase or fill parts of the image with specific values, contributing significantly to performance improvement, allowing the model to predict only parts of the image without looking at the entire part of the image, but applying it to small objects is problematic. This is because the operation of erasing a part of the image or filling it with a certain value may be applied to the whole, not to a part of a small object. MixUp [4] improves training performance by blending two images, but does not contribute to improving small object detection performance. CutMix [5] cuts and pastes the image

Fig. 2. Samples from DOTA. It consists of Google Earth, satellite, and aerial images.

to another image patch. Similarly, it improves overall training performance but does not contribute to improving small object detection performance. The third is the absence of an optimal augmentation policy. Research on the augmentation method of small objects is being conducted steadily. However, most studies do not apply optimal augmentation policies. We propose a novel optimal small object augmentation search method based on the above three perspectives. To evaluate the performance on small object detection, we use the prestigious Fast R-CNN [6] for object detection and perform a quantitative analysis on DOTA [7] Dataset. We have improved the small object detection performance by 20% compared to before.

2 Related Works

2.1 Object Detection

The object detection framework of previous studies is a two-step detector structure, consisting of a region proposal stage that is presumed to have an object and an object classification stage that classifies which category the object is. R-CNN [8] and Fast R-CNN were proposed based on a two-stage detector structure, and later a one-stage detector structure that performs region proposal and object classification at once in a convolution network, representatively YOLO [9], SSD [10], and RetinaNet. Usually, in a two-stage detector structure, the recognition is performed on an ROI with a specific object, so the accuracy is high but the speed is slow. Conversely, the one-stage detector structure has the advantage of

Table 1. GPU hours comparison of Fast AutoAugment and AutoAugment, PBA. AutoAugment measured computation cost using an NVIDIA Tesla P100, while PBA measured computation cost using a Titan XP, and Fast AutoAugment estimated computation cost using an NVIDIA Tesla V100.

Dataset	AutoAugment[14]	PBA[15]	Fast AutoAugment[16]
CIFAR-10	5000	5	3.5
SVHN	1000	1	1.5

low accuracy but high speed. because region proposal and object classification are performed on the entire image with multiple objects.

2.2 Small Object Detection

Several methods have been proposed to improve the performance of Small Object Detection. Scale-Transferrable Object Detection [11] proposed a method to generate high-resolution feature maps using the Pixel Shuffler method, which is commonly used in Image Super-Resolution, for small object detection. STDnet [12] proposed a Region Context Network (RCN) that enhances the detection of small objects in high-resolution feature maps. Augmentation for Small Object Detection [13] improved small object detection performance by proposing an algorithm that copies and pastes small objects.

2.3 Small Object Detection in Aerial Images

The DOTA dataset includes images from Google Earth, GF-2, and aerial(see Fig. 2). DOTA-v2.0 contains 18 common categories, with a total of 11,268 images and 1,793,658 instances. The dataset is divided into four subsets: train, valid, test-dev, and test-challenge. The train subset consists of 1,830 images and 268,627 instances, while the valid subset includes 593 images and 81,048 instances. The test-dev subset has 2,792 images and 353,346 instances, and the test-challenge subset has 6,053 images and 1,090,637 instances. However, ground-truth annotations are not provided for the test-dev and test-challenge subsets.

2.4 Optimal Augmentation

Data augmentation has become essential in most machine learning fields. However, determining the appropriate augmentation for dataset is a difficult problem. Although the developer determines the augmentation based on Manual Search or Grid Search, it is not the optimal augmentation suitable for dataset. As a result, active research is being conducted to find the optimal augmentation policy. AutoAugment [14] based on reinforcement learning, explored the optimal augmentation policy by giving the child model a test set loss according to the augmentation policy as a reward and achieved state-of-the-art in the classification field. However, this method is time-consuming and costly because the child

model must be repeatedly trained to update the policy searching controller (using RNN in AutoAugment). On the other hand, Population Based Augmentation [15] is based on the Population Based Training (PBT) algorithm among hyperparameter optimization techniques. Population Based Augmentation(PBA) train several models with different augmentation at the same time, and compare the performance of each model in the middle of training to replicate the parameters of the high-performance model to the parameters of the low-performance model and give some variations of the applied augmentation technique. As shown in Table 1, Unlike AutoAugment, time was reduced by 1/1000 because repetitive re-training was not required. Also, Fast AutoAugment [16] uses a trained model without augmentation to obtain an augmentation data loss according to the augmentation policy. It obtains an optimal policy by reducing the density between the original data and the augmented data. Since policy search is conducted using the trained model without repeating re-training, the time is also reduced by 1/1000 compared to AutoAugment.

3 Method

In this section, we describe augmentation algorithms for small objects and propose methods and implementations for finding optimal policies.

3.1 Augmentation Algorithm

The augmentation algorithm for small objects is based on copy-pasting strategies used in [13]. The algorithm is to copy a small object and paste it to another location. There are three types of methods.

Copy and paste a single object Select one small object from the image and paste it to a random location.

Copy and paste multiple objects Select two or more small objects from the image and paste them to a random location.

Copy and paste all objects Select all small objects from the image and paste them to a random location.

The copy-pasting algorithm ensures that the pasted object does not overlap with any existing objects. However, the edge of the copied object may appear unnatural against the background. According to [13], they tested using Gaussian blurring on the edge, but the performance actually declined and the unnatural appearance was still maintained.

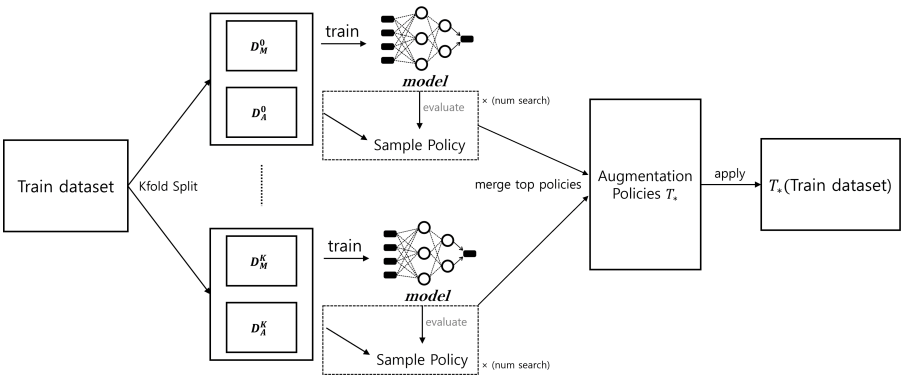


Fig. 3. An overall procedure of augmentation policy search by Fast AutoAugment algorithm.

3.2 Searching Policies

Searching for the optimal augmentation policy is based on Fast AutoAugment. Fast AutoAugment is a method of searching for an augmentation policy that is most suitable for the characteristics of Dataset by estimating density similarity between original data and augmented data. The methodology proposed by Fast AutoAugment for density similarity estimation is that if the augmented data applied with the augmentation policy for the model trained with the original data has a low loss, the optimal augmentation policy. In other words, the lower the loss for the model trained without augmentation, the more similar the density to the original data, and the most appropriate augmentation policy for the characteristics of the dataset.

3.3 Searching Small Object Augmentation Policies

SOA [13] found the optimal algorithm policy in a way close to Manual search or Grid search by changing the parameter coefficient to improve the performance of small object detection. In this paper, the policy for three copy-pasting algorithms is explored with the Bayesian Optimization TPE [17]. As a result, the optimal copy-pasting policy for small object detection is searched.

3.4 Implementation

Using the Kfold method in the Sklearn [18], the train data is split into D_M^K and D_A^K(see line 1 in Algorithm 1). After that, the model is trained in parallel on each D_M^K without augmentation (line 3). After training, augmentation policies are searched for using the HyperOpt function in Ray [19], which is a library for hyperparameter optimization (line 5-7). The search method is based on the

Algorithm 1 Implementation pseudo code

Require: Train Dataset D, $numSearch$, K, N

1: Split D into Kfold data D_M^K, D_A^K
2: **for** $k = 1, \ldots, K$ **do**
3: Train Model M^k on D_M^k
4: **for** $t = 0, \ldots, numSearch - 1$ **do**
5: T_t = Search *operation*, p, m
6: BayesianOptim(T_t, $Loss(M^k|T_t(D_A))$)
7: $T_t^k = T_t^k \cup T_t$
8: $T_* = T_t \cup$ (select top N policies in T_t^k)
9: Train Model M on $T_*(D)$

Bayesian optimization TPE [17], and the searching parameters are operation, p, and m(line 5-7). Operation is the copy-pasting algorithm explained in Subsection 3.1, p is the copy-pasting probability, and m is the parameter of how many times to paste. In the case of m, the parameter is optimized based on the number of times 1 to 3. Then, the searched policy is applied to D_A to obtain the loss for the augmentation policy(lines 6-7). For each search, the top N policies with the lowest loss are added to the final policies T_* (line 8), and one of the T_* policies is randomly chosen and applied as the augmentation policy for each iteration during the final model training. Finally, model is trained by applying the searched policies T_*(line 9). Fig. 3 shows the overall procedure.

4 Experiments and Results

In this section, we conduct experiments to compare the performance of our proposed methods with the Baseline and SOA, in DOTA-v2.0 valid. Here, baseline is the result of training with the setting and DOTA setting proposed in the object detection model papers, and SOA is the result of applying the best augmentation policy in the SOA paper. The comparison method is Average Precision (AP). The AP scores are calculated separately for four categories: All, Small (object size 0 to 32×32), Medium (object size 32×32 to 96×96), and Large (object size 96×96 or larger). Our proposed method demonstrated a significant improvement in mAP performance, as shown in Tables 2 and 3. Compared to the baseline, our method achieved a 9% increase in mAP performance in Table 2, and a 11% increase in Table 3. Furthermore, for small objects, our method showed a substantial 20% improvement in mAP performance in Table 2, and a 17% improvement in Table 3. These results highlight the effectiveness of our proposed method in object detection tasks, particularly for detecting small objects. The searched optimal policies are depicted in Fig. 4, showing the distribution of probability(p) and magnitude(m). As can be seen from the distribution, probability and magnitude tend to be inversely proportional.

Table 2. Results of our experiments using Faster R-CNN based on RPN(resnet50 backbone). Experimental results are based on AP(Average Precision) metric.

	mAP	mAP_L	mAP_M	mAP_S
baseline [6]	0.491	0.591	0.543	0.402
SOA [13]	0.517	0.579	0.572	0.461
Ours	**0.538**	0.573	**0.578**	**0.485**

Table 3. Results of our experiments using RetinaNet(MobileNetV3 backbone). Experimental results are based on AP(Average Precision) metric.

	mAP	mAP_L	mAP_M	mAP_S
baseline [20]	0.323	0.555	0.374	0.122
SOA [13]	0.346	0.544	0.433	0.132
Ours	**0.359**	**0.586**	**0.447**	**0.143**

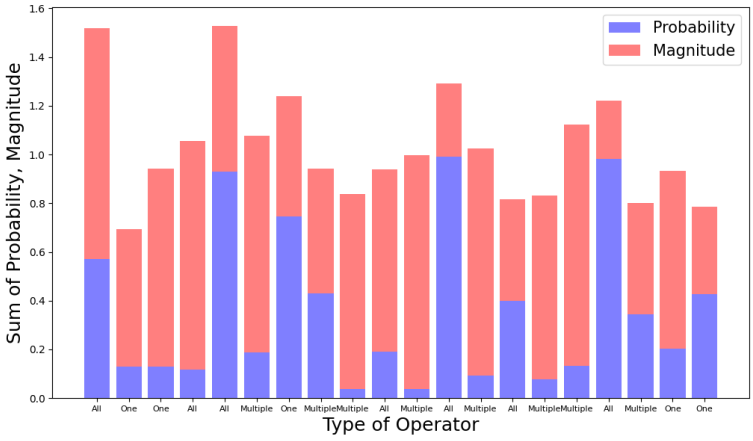


Fig. 4. The distribution of probability and magnitude of the top 20 policies using Faster R-CNN, where the x-axis represents the type of copy-pasting and the y-axis represents the sum of the parameters p and m. Examining the parameter values, it can be observed that they exhibit an inverse relationship, and that optimal performance is achieved when the values are inversely proportional.

5 Conclusion

We investigated the problem of small object detection. On most datasets, small objects are much smaller than large or intermediate objects, which negatively affected small object detection performance. We introduced a small object augmentation strategy to address this problem and proposed a method to improve small object detection performance by finding an optimal augmentation policy. Our experiments show a mAP 9% and small object mAP 20% performance improvement of the proposed method in DOTA-v2.0.

References

1. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
2. Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.
3. Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.
4. Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.
5. Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.
6. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.
7. Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.
8. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
9. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
10. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.
11. Peng Zhou, Bingbing Ni, Cong Geng, Jianguo Hu, and Yi Xu. Scale-transferrable object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 528–537, 2018.
12. Brais Bosquet, Manuel Mucientes, and Víctor M Brea. Stdnet: A convnet for small target detection. In *BMVC*, page 253, 2018.

13. Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho. Augmentation for small object detection. *arXiv preprint arXiv:1902.07296*, 2019.
14. Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019.
15. Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *International Conference on Machine Learning*, pages 2731–2741. PMLR, 2019.
16. Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *Advances in Neural Information Processing Systems*, 32, 2019.
17. James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.
18. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
19. Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging {AI} applications. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 561–577, 2018.
20. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

Advancing Text Summarization through the Utilization of Arbitrary Aspect Learning

Ziwei Hou, Bahadorreza Ofoghi, Nayyar Zaidi, Musa Mammadov, Shamsul Huda, and John Yearwood

School of Information Technology, Deakin University, Burwood Victoria 3125, Australia
{houziw,b.ofoghi,nayyar.zaidi,musa.mammadov,shamsul.huda,john.yearwood}@deakin.edu.au

Abstract. Over the past decades, there has been a surge in the volume of textual data derived from various sources. As the abundance of text carries valuable information and knowledge, summarizing it is extremely desirable. Text summarization is one of the extensively studied and current topics in Natural Language Processing (NLP). Many text summarization techniques have been developed to correctly extract significant and focused information from long text documents such as news articles. Pre-trained language models are among the most effective methods that automatically filter out redundant information from text. However, most existing models do not summarize textual documents with respect to the key aspects of focus. In this paper, we propose a novel aspect-based text summarization model based on learning arbitrary, non-pre-designed aspects from data and with no reference to any auxiliary, external resources. We studied the effectiveness of the incorporation of the aspects learned from data when augmented into the baseline transformer-based and other summarization methods. Three benchmark datasets were used to validate the effectiveness of our model. Our experiments show improvements over the baseline methods when arbitrary aspects are added to the learning of the text summarization process.

Keywords: Text summarization · Deep learning · Aspect analysis.

1 Introduction

Text summarization is vital in natural language understanding and it has applications in information retrieval [15]. The process of text summarization is implemented using both text extraction and generation [3, 23]. Effective text summarization can benefit a wide range of downstream natural language tasks. Moreover, it can achieve difficult tasks while reducing expert efforts and avoiding inconsistencies in the summaries [24]. Current research focuses on reducing an overwhelming amount of text to a shorter, more digestible form while retaining the important information and key aspects. One of the challenges of text summarization is that the token embeddings learned from the documents give

biased representations of sentences in terms of their real semantic content when averaging high-frequency words [11]. In addition, the availability of domain-specific corpora is limited [21]. The lack of resources is a considerable obstacle to fine-tuning. Moreover, as the raw data from social media contain abundant information, the unrelated information can be combined with influential data which carry the evidence parts of news like aspects and topics. This irrelevant information will disturb the attention to the evidence in news [9].

In this paper, we propose a new text summarization approach based on arbitrary aspects, which are unknown and learned from data. Aspects are usually a word or a few terms within sentences that contain some key information. For instance, in the sentence "*He made the comments in a statement disassociating himself from a new exhibition of his artistic creations that have been removed from walls.*", there are at least two aspects, "`comments`" and "`artistic creations`". These aspects carry important information of the sentence which will contribute to the creation of a summary that will represent the sentence. The approach is designed as an unsupervised model to collect the words related to the content of the current document collection. Aspects are extracted using neural word embeddings to map words that appear in similar contexts to similar positions in the embedding space. Then, the attention mechanism is used to generate word embeddings in the sentences. The aspect embedding training process is similar to an autoencoder which uses dimension reduction to extract the common factors in the embedded sentences, and then uses a linear combination of the aspect embeddings to reconstruct the sentences. The main contribution of this work is that the new proposed architecture extracts summaries while integrating arbitrary aspects into the model. The aspects provide latent key information that is extracted from within given documents without reference to any external data.

2 Related Work

Text summarization works can be categorized into two major groups, extractive summarization and abstractive summarization. Extractive summarization approaches [25, 13] focus on scanning the whole document, filtering major words, and concatenating important elements in the source text. Meanwhile, abstractive summarization approaches [16, 2] target modifying the original text by generating new summaries, which include paraphrases of the primary text. The first category of approaches can ensure grammatical correctness and achieve high accuracy. However, the second category requires the models to have the ability to represent the semantic content of the original text and use this information for generating summaries. This is a challenging task as it requires that models improve their ability to use words creatively or make inferences from the source text. With recent advances in NLP and the availability of large pre-trained language models, research related to text summarization has progressed significantly [23, 22]. Most of them rely on attention mechanisms. GPT [16] from OpenAI and

Bidirectional Encoder Representations from Transformers (BERT) [1] are two well-known models in the community.

GPT [16] is a pre-trained language model based on the decoder of a transformer. It is widely used for many text summarization and generation tasks. As an autoregressive model, GPT will only encode the forward information and hence will lose some of the information from the input. In addition, GPT requires an encoder to capture the dependency relationship encoder to the news claim and the retrieved facts. A Pseudo-self-attentive (PSA) [26] is proposed to capture bi-directional information among the inputs. PSA incorporates the conditioning input as a pseudo-history to a pre-trained transformer.

Apart from GPT-related models, there are some works using BERT proposed by Devlin et al. [1]. A variant of BERT named BERTSUM for summarization tasks was proposed by Liu. [13]. BERTSUM is built on top of BERT with additional summarization layers. Lewis et al. [10] proposed a pre-training sequence-to-sequence model, BART, which can be applied to the text summarization task. Compared to GPT and BERT, the BART model adopts the overall structure of the Transformer.

The aspect-based analysis is widely used in sentiment analysis, text generation, recommendation, etc. It provides a fine-grained analysis through text. The identified aspects can be used to generate a summary of long documents more accurately and comprehensively. In the work of Frermann et al. [4], an aspect-focused summarization is proposed. It produces a one-to-one summary of the target aspect based on the given documents. Tan et al. [22] designed a weakly supervised summarization model by using external aspects from Wikipedia or other external sources based on BART. In contrast to these previous works, the proposed architecture focuses on learning aspects from the internal content of existing documents.

Related to aspect-based summarization, there are some works based on topic-based summarization [14, 18]. In the work of Ma et al. [14], a topic-related text summarization model, called T-BERTSUM, is built. The authors built T-BERTSUM with three parts, Representation, Neural Topic Model (NTM), and Summarization. Inspired by the previous works on text summarization, we developed an aspect-based summarization technique that takes advantage of discovered aspects from documents to generate textual summaries. Different from the works in the literature, in this work, aspect information is learned and extracted from word embeddings in an unsupervised way. Besides, there is no data from external resources used.

3 Aspect-Based Text Summarization

The following proposed architecture is designed to address the problem of finding the aspects from the input news. To verify the effectiveness of introducing aspects, three state-of-art summarization models are used, PSA, BERTSUM, and BART. These architecture are shown in Fig. 1 and Fig. 2 which will be introduced in Sections 3.2 and 3.3. It can be seen that the proposed architec-

ture are built of mainly two components, an external aspect extraction, and a language model:

- The aspect extractor is built based on a self-attention mechanism which can give more weight to aspect information.
- Pre-trained language models PSA [26], BERTSUM [13], and BART [10] are trained to filter out redundant information from documents.

Unlike the baseline models, the proposed architecture learns and extracts latent aspect embeddings at the word level from each sentence. With an attention-based aspect extractor, latent aspect embeddings are obtained for a more nuanced understanding of the document. The pre-trained language model is fed with original input and latent aspect embeddings, and trained for the downstream text summarization task.

3.1 Aspect Learning and Extraction

The aspect extractor is used to learn a set of aspect embeddings so that each aspect can be explained by representative words in the embedding space [6]. Firstly, each word w is represented by a feature vector $e_w \in R^d$. Word embeddings are used to construct feature vectors, and map words that often occur together in the context to closer points in the embedding space. The feature vector associated with the word corresponds to the row of the word embedding matrix $E \in R^{V \times d}$, and V represents the number of words in the vocabulary. The embeddings of aspects matrix $M_A \in R^{K \times d}$ can be learned as aspects, and words share the same embedding space. Here, K represents the number of aspects, and it is smaller than V. Aspect embeddings are used to approximate the aspect words in the vocabulary, and are filtered by the attention mechanism. To remove or pay less attention to irrelevant words and improve the coherence of the filtered aspects, reconstructing sentences with aspects can be achieved with the following steps:

- Map to feature vector e_{w_i} for each word w_i, $i = 1, ..., n$;
- Construct sentence embedding Z_s to capture the most relevant information at the sentence level: $Z_s = \sum_{i=1}^{n} a_i e_{w_i}$, where a_i is the weight of w_i.

Next, we calculate p_t and reconstruct sentence embedding r_s. Here, p_t are the weight vectors of K aspect embeddings. It represents the probability that the sentence belongs to this aspect. It can be computed by reducing Z_s from d dimension to K dimension, and then softmax is used to normalize the outputs, $p_t = \text{Softmax}(W \cdot Z_s + b)$, where W is the weighted matrix parameter, b is bias, and r_s is the reconstruction vector which can be considered as a linear combination of aspect embeddings, $r_s = M_A^\top \cdot p_t$.

3.2 Aspect-Based PSA

Fig.1 shows the structure of the proposed Aspect-Based PSA. Once the relevant words are filtered with aspect extraction in Section 3.1, PSA language model [26]

is launched to capture the bi-directional information. PSA can be formulated as: $\text{PSA}(Y, X, A) = \text{Softmax}(Q_Y K_Y K_X K_A^\top) V_Y V_X V_A$, where Q is the query, K is the key, and V is the value in the self-attention mechanism. $Y \in T \times D$ represents the input sentence, $X \in S \times D$ is the length of input sentence S, and A is the extracted aspects. The objective function of PSA [19] is $L_{\text{PSA}} = -\sum_{i=1}^{M}(\log \text{P}(y_i|y_1, ...y_{i-1}; X, A))$, where $y_i, i = 1, .., M$ is the generated content based on X and A.

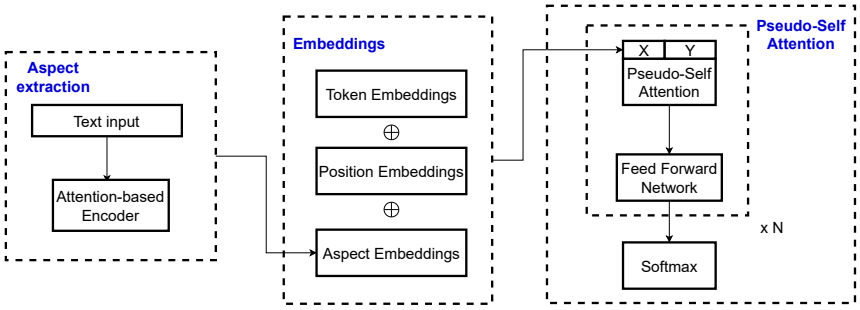


Fig. 1. The overall architecture of aspect-based PSA for text summarization.

3.3 Aspect-Based BERTSUM

To better understand the benefits of aspects of text summarization tasks, we consider another state-of-the-art method, BERTSUM [13]. Fig. 2 illustrates the proposed architecture. The multi-sentence input is split by adding the [CLS] token at the beginning of each sentence and adding the [SEP] token at the end of each sentence. Segment embeddings, EA and EB, are used to discriminate the order of the sentence in the text. Position embeddings, E_0, E_1 ... , E_n, show the positions of the token in the input. Token, segment, and position embeddings are summed and treated as the input into BERT to generate sentence embeddings T_n. The learned aspect embeddings AT_n from aspect extraction in Section 3.1 are concatenated with sentence embeddings and fed into the summarization layer to predict the probability of each sentence in the original document being a part of the extracted summary, and finally, the optimal top-n sentences are selected as the document summary.

3.4 Aspect-Based BART

As discussed earlier, BART adopts an encoder-decoder structure, where the input at the encoder side is a noise-added sequence, the input at the decoder side is a right-shifted sequence, and the target at the decoder side is the original sequence. Aspect-Based BART takes the aspect embeddings from aspect extraction in Section 3.1 as the additional inputs. It can preserve the autoregressive properties while exploiting the bi-directional modeling capability of the encoder side for generative tasks.

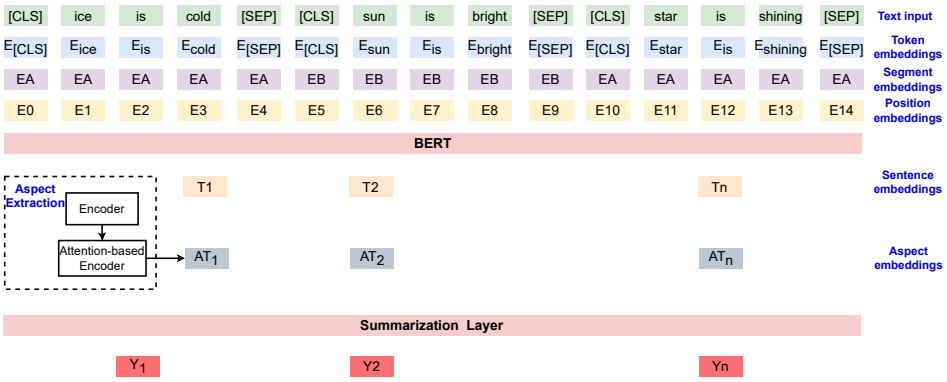


Fig. 2. The architecture of aspect-based BERTSUM.

4 Experiments

4.1 Experimental Settings

For all experiments, we used the original, published language models of GPT-2, PSA, BERTSUM, and BART. For aspect extraction, the number of aspects was 500 for the CNN DailyMail and MA-News datasets, and 140 for the GossipCop dataset. An analysis was conducted to find the optimal number of aspects per dataset, which will be discussed in Section 4.4. We adopt the standard parameter settings of the methods which are listed in Table 1.

Models	Optimizer	Learning Rate	Epochs	Layers	Heads per Layer	Hidden Size
GPT-2	Adam	1e-3	5	12	12	768
PSA	Adam	1e-3	5	12	12	768
BERTSUM	Adam	1e-4	5	12	12	768
BART	Adam	3e-5	5	12	12	768

Table 1. Experimental parameter settings.

4.2 Datasets

Following previous works [17, 13, 26], we evaluate proposed architectures on the CNN DailyMail [7], MA-News [4] and GossipCop [20] datasets.

– CNN DailyMail ismade up of online news articles from CNN and Daily Mail websites. There are 93,000 articles that are collected from CNN[1] from April 2010 to April 2015 and 220,000 articles from Daily Mail[2] between June 2010 to April 2015 [7]. The whole dataset contains 287,266 examples for training, 13,368 examples for validation, and 11,490 examples for testing.

[1] https://edition.cnn.com/
[2] https://www.dailymail.co.uk/auhome/index.html

Model	R-1(%)	R-2(%)	R-L(%)	JSD	KLD
GPT-2	29.34	8.27	26.58	0.0984	0.0598
PSA	40.23	**17.73**	37.27	0.1035	0.0653
PSA+aspects*	**40.35**	15.01	**38.34**	**0.1048**	**0.0667**
BERTSUM	43.31	20.15	**39.58**	0.1536	0.0760
BERTSUM+aspects*	**43.89**	**20.70**	39.51	**0.1603**	**0.0892**
BART	44.16	**21.28**	40.90	0.1543	0.0754
BART+aspects*	**44.59**	21.14	**41.03**	**0.1634**	**0.0826**

Table 2. Summarization performances of the baseline and aspect-based models on the `CNN DailyMail` dataset. Note: * denotes aspects learned through the proposed model. The number of aspects*=500.

- MA-News is a synthetic dataset from CNN DailyMail [4]. Documents are pre-learned into 6 different aspects. The dataset contains 280,000 training samples, 10,000 validation samples, and 10,000 testing samples.
- GossipCop is one of the datasets from FakeNewsNet. It contains news articles from the GossipCop website. We adopt the same settings with Shu et al. [20]. The dataset contains 7,331 training samples, 1,459 validation samples, and 974 testing samples.

4.3 Evaluation

For evaluating the performance of the proposed architecture, ROUGE [12] is applied which is one of the standard evaluation metrics for text summarization. We considered the coverage of one word, two words, and the longest term (ROUGE-1, 2, L) between the gold summary from the dataset and the summary generated by the models [12]. Additionally, Jensen-Shannon Divergence (JSD) [5] and Kullback-Leibler Divergence (KLD) [8] were considered for performance evaluation. JSD and KLD compute divergence between ground-truth summary with the generated summary at the token level.

Table 2 summarizes the results on the CNN DailyMail dataset. The proposed models outperform the baseline models while achieving higher ROUGE and JSD/KLD scores for text summarization. Aspect-based PSA outperforms GPT-2, and PSA. Basides, aspect-based BERTSUM shows improvements compared to BERTSUM for ROUGE-1, ROUGE-2, JSD, and KLD respectively. In the last section of Table 2 where BART is used as the language model, the aspect-based BART achieves 44.59, 41.03, 0.1634, and 0.0826 in ROUGE-1, ROUGE-L, JSD, and KLD that outperform the baseline BART model.

From Table 3, it can be observed that on the GossipCop dataset, the aspect-based models outperform their counterpart models in terms of ROUGE-L, JSD, and KLD.

Table 4 shows the experimental results on the MA-News dataset. The experiments on the baseline models are without the pre-defined aspects in the dataset. For both BERTSUM and BART, we designed four types of experiments: i) language model without any aspects, ii) language model with given aspects in the dataset only, iii) language model with our aspects learned from the dataset only, and iv) language model with both given aspects and our aspects learned from the dataset. In Table 4, PSA with aspects achieves improvements on Rouge-1, 2,

Model	R-1(%)	R-2(%)	R-L(%)	JSD	KLD
GPT-2	15.58	5.37	17.23	0.1253	0.0651
PSA	20.84	**9.71**	18.89	0.1327	0.0802
PSA+aspects*	**21.05**	9.37	**20.01**	**0.1384**	**0.0815**
BERTSUM	**21.41**	9.87	20.17	0.1439	0.0853
BERTSUM+aspects*	21.32	**9.91**	**20.44**	**0.1592**	**0.0862**
BART	**23.57**	10.05	22.89	0.1485	0.0875
BART+aspects*	23.45	**10.07**	**23.15**	**0.1670**	**0.0883**

Table 3. Summarization performances of the baseline and aspect-based models on the `GossipCop` dataset. Note: * denotes aspects learned through the proposed model. The number of aspects*=140.

Model	R-1(%)	R-2(%)	R-L(%)	JSD	KLD
GPT-2	25.91	6.89	22.01	0.1051	0.0514
PSA	34.89	13.62	33.43	0.1392	0.0539
PSA+aspects*	**35.15**	**14.25**	**33.83**	**0.1457**	**0.0572**
BERTSUM	38.86	18.14	38.07	0.1350	0.0685
BERTSUM+MA-News aspects	40.34	19.51	38.67	0.1395	0.0672
BERTSUM+aspects*	40.18	19.93	39.07	0.1571	0.0678
BERTSUM+MA-News aspects+aspects*	**41.17**	**20.76**	**39.95**	**0.1584**	**0.0715**
BART Sup 280K	40.07	18.78	37.83	0.1505	0.0701
BART Sup 280K+MA-News aspects [22]	41.98	20.65	39.64	0.1538	0.0717
BART Sup 280K+aspects*	42.39	21.28	39.59	0.1611	0.0782
BART Sup 280K+MA-News aspects+aspects*	**42.73**	**21.56**	**40.63**	**0.1615**	**0.0790**

Table 4. Summarization performances of the baseline and aspect-based models on the `MA-News` dataset. Note: * denotes aspects learned through the proposed model. The number of aspects*=500.

L, JSD, and KLD, compared with PSA. For BERTSUM and BART, the learned aspects can improve the summarization performance.

One can conclude from these improved scores that adding aspect embeddings to the model is indispensable. The embeddings of aspect information can facilitate the process of classifying and identifying the text so that an accurate summary can be formed from the original. It should be noted that with our improved model, we not only encode and decode the text but also embed the text from the perspective of text interpretation and inference while incorporating the different contexts (aspects) of the text within the embedding process.

4.4 Analysing the Effect of the Number of Aspects

To investigate the impact of the number of aspects, we designed experiments on the variation of the ROUGE-1, ROUGE-2, ROUGE-L, JSD, and KLD scores with varying numbers of aspects. For these experiments, BERTSUM is used, and the study was conducted on the CNN DailyMail and GossipCop datasets. The results of these experiments are shown in Table 5 and Table 6 with 14, 140, 200, 500, 1000, and 5000 aspects.

In Table 5, with the setting of aspect number as 500, the best performance scores on CNN DailyMail are achieved, except in the case of JSD where the

utilization of 1000 aspects results in a better JSD performance. For the GossipCop dataset, 140 aspects outperform all the other aspect numbers in Rouge-2, JSD, and KLD as shown in Table 6. The other best performances (Rouge-1 and Rouge-L) are at 14 aspects.

#Aspects	R-1(%)	R-2(%)	R-L(%)	JSD	KLD
14	43.27	20.01	38.93	0.1017	0.0756
140	43.57	20.52	39.02	0.1359	0.0855
200	43.38	20.51	39.25	0.1478	0.0863
500	**43.89**	**20.70**	**39.51**	0.1603	**0.0892**
1000	43.52	20.37	39.34	**0.1675**	0.0874
5000	40.15	18.83	35.24	0.1326	0.0880

Table 5. The analysis of the impacts of the number of aspects on the text summarization performance of BERTSUM+aspects on the CNN DailyMail dataset where aspects are learned through the proposed model.

#Aspects	R-1(%)	R-2(%)	R-L(%)	JSD	KLD
14	**21.35**	9.89	**20.47**	0.1439	0.0811
140	21.32	**9.91**	20.44	**0.1592**	**0.0862**
200	21.33	9.56	20.35	0.1485	0.0857
500	21.15	9.84	20.19	0.1327	0.0803
1000	20.95	9.79	20.07	0.1384	0.0810
5000	20.58	9.23	20.01	0.1253	0.0715

Table 6. The analysis of the impacts of the number of aspects on the text summarization performance of BERTSUM+aspects on the GossipCop dataset where aspects are learned through the proposed model.

The aspect-based BERTSUM model on CNN DailyMail dataset performs better with a larger number of aspects while on GossipCop, a smaller number of aspects leads to better performance. Given the size difference between the CNN DailyMail and GossipCop datasets (312,124 text documents vs. 9,764, respectively), the different optimal number of aspects is attributed to the number of text documents in the dataset.

4.5 Case Study and Discussion

In Table 7, there is an example of summaries from CNN DailyMail. It contains the generated summaries of the proposed architecture, PSA + Aspect and baseline models, GPT-2, and PSA as compared to the ground-truth summary. It is encouraging to see that the summary generated through our aspect-based model contains the keywords related to the document aspects, e.g., 'Italian frigate'. This aspect is extracted from the Aspect Extraction stage. Then, it is concatenated with input embeddings from the document and inserted into the following summarization model. The extracted aspect is one of the keywords in the document that should be contained the generated summary. This keyword while present in the ground-truth summary is missing within the summary texts generated by the two baseline models GPT-2 and PSA. Besides, the word 'Wednesday' describe the date of the reported sighting of a suspected missile. This word is

Original Article	LRB- CNN -RRB- A Libyan government spokesman said Thursday that forces loyal to embattled leader Moammar Gadhafi did not fire a missile at NATO warships, as he had told reporters a day earlier. The Italian frigate Bersagliere reported spotting a suspected missile about 19 kilometers -LRB- 12 miles -RRB- off the port city of Zlitan on Wednesday. The weapon had fallen harmlessly into the sea, according to Italy's defense ministry and NATO. Government spokesman Moussa Ibrahim told reporters that government forces were responsible for the attack, but said Thursday that his comment was a " misunderstanding. " Ibrahim told CNN that military officials later told him they hadn't launched a missile at the NATO fleet currently off Libya. " We did not fire any missiles into the sea against the ship, "Ibrahim told CNN. Gadhafi's forces have been battling a nearly six-month-long revolt that has split Libya roughly in half. NATO planes and warships joined the fight in March, pounding government troops under a U.N. mandate to protect civilians from reprisals by Gadhafi loyalists. Though most of the allied campaign has been conducted from the air, NATO warships reported fighting at least two brief sea engagements with pro-Gadhafi forces off the port city of Misrata in May. CNN's Jomana Karadsheh contributed to this report.
Ground-Truth Summary	Libya now says it didn't fire a missile at an Italian frigate. The warship reported spotting the missile on Wednesday. The weapon landed harmlessly.
GPT-2	Libyan military says it did not launch missiles at NATO warships. The weapon fell harmlessly into the sea, according to the Italian navy. NATO planes and warships joined the fight in March, pounding government troops under a U.N. mandate to protect civilians
PSA	Libyan says not launch a missile at the NATO on Wednesday. Planes and warships protect civilians.
PSA + aspects*	Libyan government says it did not fire a missile at NATO warships. Spokesman told reporters that his comments were a misunderstanding. The Italian frigate reported spotting a suspected missile off Libya on Wednesday.

Table 7. Sample generated summaries of the proposed architecture and baseline models compared with the ground-truth summary on a CNN DailyMail document. Note: * denotes aspects learned through the proposed model. The number of aspects*=500.

missing in the generated summary from GPT-2. When compared with the output summaries of the baseline models, the proposed aspect-based model is able to learn aspects from the documents that carry the important information. The proposed model does not lose sight of the important information within the document. Our model is able to capture the main aspects and even dispersed aspects within the long document.

5 Conclusion

In this paper, we proposed a novel aspect-based text summarization model that leverages textual aspects to improve upon the state-of-the-art text summariza-

tion effectiveness. Our approach does not rely on any external data or any predesigned set of aspects; instead, arbitrary aspects are directly learned from the data. With the experimental results, it was demonstrated that this arbitrary aspect learning process can improve the quality of the generated contents in text summarization tasks. Our aspect-based text summarization models outperformed both extractive and abstractive baseline, state-of-the-art summarization models on several benchmark datasets with varying numbers of textual documents. Our analyses of the number of aspects to be learned from the data showed that the larger datasets with longer documents (and larger vocabularies) tend to have a larger number of aspects that can optimally enhance the summarization effectiveness. The latter finding may require further analysis with other datasets in future work. We also plan to study the transferability of aspects among different datasets and to further confirm the need for within-domain aspect learning.

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (2018)
2. Dou, Z.Y., Liu, P., Hayashi, H., Jiang, Z., Neubig, G.: Gsum: A general framework for guided neural abstractive summarization. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4830–4842 (2021)
3. Fang, L., Zeng, T., Liu, C., Bo, L., Dong, W., Chen, C.: Outline to story: Fine-grained controllable story generation from cascaded events. arXiv preprint arXiv:2101.00822 (2021)
4. Frermann, L., Klementiev, A.: Inducing document structure for aspect-based summarization. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 6263–6273 (2019)
5. Fuglede, B., Topsoe, F.: Jensen-shannon divergence and hilbert space embedding. In: International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings. p. 31. IEEE (2004)
6. He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: An unsupervised neural attention model for aspect extraction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 388–397 (2017)
7. Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P.: Teaching machines to read and comprehend. Advances in neural information processing systems **28** (2015)
8. Hershey, J.R., Olsen, P.A.: Approximating the kullback leibler divergence between gaussian mixture models. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07. vol. 4, pp. IV–317. IEEE (2007)
9. Kozyreva, A., Lewandowsky, S., Hertwig, R.: Citizens versus the internet: Confronting digital challenges with cognitive tools. Psychological Science in the Public Interest **21**(3), 103–156 (2020)
10. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for

natural language generation, translation, and comprehension. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020)
11. Li, B., Zhou, H., He, J., Wang, M., Yang, Y., Li, L.: On the sentence embeddings from pre-trained language models. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 9119–9130 (2020)
12. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out. pp. 74–81 (2004)
13. Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3730–3740 (2019)
14. Ma, T., Pan, Q., Rong, H., Qian, Y., Tian, Y., Al-Nabhan, N.: T-bertsum: Topic-aware text summarization based on bert. IEEE Transactions on Computational Social Systems **9**(3), 879–890 (2021)
15. Mahalakshmi, P., Fatima, N.S.: Summarization of text and image captioning in information retrieval using deep learning techniques. IEEE Access **10**, 18289–18297 (2022)
16. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
17. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
18. Shenouda, G., Bossard, A., Ayoub, O., Rodrigues, C.: Summvd: An efficient approach for unsupervised topic-based text summarization. In: Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing. pp. 501–511 (2022)
19. Shu, K., Li, Y., Ding, K., Liu, H.: Fact-enhanced synthetic news generation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 13825–13833 (2021)
20. Shu, K., Mahudeswaran, D., Wang, S., Lee, D., Liu, H.: Fakenewsnet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media. Journal of Big Data (2020)
21. Soni, S., Roberts, K.: Evaluation of dataset selection for pre-training and fine-tuning transformer language models for clinical question answering. In: Proceedings of The 12th Language Resources and Evaluation Conference. pp. 5532–5538 (2020)
22. Tan, B., Qin, L., Xing, E.P., Hu, Z.: Summarizing text on any aspects: A knowledge-informed weakly-supervised approach. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2020)
23. Xu, S., Zhang, X., Wu, Y., Wei, F.: Sequence level contrastive learning for text summarization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 11556–11565 (2022)
24. Zhang, M., Zhou, G., Yu, W., Huang, N., Liu, W.: A comprehensive survey of abstractive text summarization based on deep learning. Computational Intelligence and Neuroscience **2022** (2022)
25. Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., Huang, X.: Extractive summarization as text matching. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020)
26. Ziegler, Z.M., Melas-Kyriazi, L., Gehrmann, S., Rush, A.M.: Encoder-agnostic adaptation for conditional language generation. arXiv preprint arXiv:1908.06938 (2019)

Clustering Multivariate Longitudinal Data Application on Disease Progression Modeling*

Loujain Liekah[0000−0003−1910−5403], Haytham Elghazel[0000−0002−6546−1567], Fabien De Marchi[], and Mohand-Saïd Hacid[0000−0002−9591−9591]

LIRIS - University of Claude Bernard Lyon 1, Villeurbanne 69100, France.
loujain.liekah5@gmail.com - firstname.lastname@univ-lyon1.fr

Abstract. Methods for identifying homogeneous groups with varying characteristics in longitudinal data have been receiving increasing attention in recent years, especially in the medical domain. Exploiting electronic health records (EHRs) to infer patient subtypes can support practitioners in improving the decision-making process. In this paper, we propose a dynamic method for clustering multivariate longitudinal data, which constitutes a transparent solution for patient subtyping and modeling disease progression. Based on the assumption that subjects with similar disease trajectories share the same patterns, we subtype patients based on their medical history then learn the disease progression model. We cluster data periodically, and maintain the results and update the deduced subtypes by applying a borrowed approach from the data integration domain, namely entity matching. We test our method on the Alzheimer's Disease Neuroimaging Initiative (ADNI) real-world dataset. We demonstrate how our results can facilitate the early detection of dementia two years on average before the actual diagnosis, and therefore assist in the development of support and prevention procedures for patients. We compare our clustering with the state-of-the-art data stream clustering algorithms, and show that our method exhibits higher effectiveness in terms of both internal and external validation metrics.

Keywords:
Multivariate longitudinal clustering · Subtyping · Disease progression modeling · Alzheimer's disease

1 Introduction

Longitudinal data consists of repeatedly measured observations for the same subjects at multiple time points, hence the existence of a time dimension. Patients with severe medical conditions undergo frequent monitoring. Therefore, medical data is longitudinal. It is often stored in electronic health records (EHRs), which comprise patient demographic and medical information [28].

* This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 875171.

Chronic diseases can be partitioned into different stages, where each stage is characterized by the level of progression. Researchers have been exploring machine learning and data mining methods in order to understand the history of disease, and estimate disease paths [18]. The problem has been modeled with varying objectives, such as predicting the future diagnosis [6], estimating survival rates [7], subtyping patients [3], and modeling disease progression [25].

Patient subtyping aims to derive and detect groups of patients with similar traits and patterns. It can be applied to distinguish between mild and severe stages of a disease [21]. Subtyping patients can be performed by clustering [16]. However, the longitudinal property of medical data necessitates a dynamic approach to avoid reinterpretation and repetition of cluster analysis.

A successive application of subtyping is modeling disease progression. It is concerned with finding an ordered sequence of stages of the progressive disease, and a qualitative estimate of prognosis probabilities between these stages. It can be modeled using a Markov decision process, where a state represents the disease stage, and the transition matrix reflects the probabilities of advancing between stages [15]. Markov model has been used to represent breast cancer, although the stages were predefined and not automatically determined from the data [5].

Alzheimer's disease (AD) is widely studied for the purpose of early detection [20]. Nonetheless, most approaches focus on improving the accuracy of the prediction with disregard for model transparency.

In this work, we propose an approach for clustering multivariate longitudinal datasets, which constitutes an interpretable method for subtyping patients. Our approach divides the dataset using a time window, and optimally clusters each division. Each cluster corresponds to a subtype. We define the subtype as an entity, characterized by the size of the cluster, in addition to the input variables as properties with the values corresponding to the center of the cluster. Afterwards, we borrow the concept of entity matching from the data integration domain in order to find the same subtype entity across different times. Finally, we model the disease progression as a Markov process.

It should be noted that, while our method is generic enough to be applied to any dataset, in this paper, we assess the proposed model on the real-world dataset of Alzheimer's disease. We show that our method outperforms state-of-the-art data stream clustering algorithms in terms of both internal and external cluster validation metrics. In addition, our model was able to perform an early detection of dementia two years on average before the real diagnosis. This early discovery of Alzheimer's, or any other disease, can support prevention and treatment planning and improve patients' lives.

2 Related Work

Clustering algorithms seek groups of instances that share distinct internal similarities. It is intuitively a subtyping approach [16]. When applied to medical data for subtyping patients, clustering divides patients into groups sharing similar characteristics, which can reveal valuable information, and improve the diagno-

sis ability. Clustering has been widely used for subtyping and aiding the diagnosis of different diseases such as breast cancer [29], Parkinson [19], and Alzheimer's disease [10]. However, follow-up information are collected periodically in different time steps, which makes medical data longitudinal. This means clustering should be performed repeatedly with maintenance to infer robust subtypes. This process can be computationally expensive, and will require repetition of cluster analysis and result interpretation.

The research on data streams has flourished with the development of devices that captures measurements over time. Data streams are often unlabeled, which makes it inevitable to resort to unsupervised learning methods for analysis purposes. Moreover, data stream processing needs to address the possible change in the properties of the data instances over time. Authors in [1] argue that holistic clustering over the entire stream is insufficient to capture the change in clusters over time. Therefore, they proposed clustering evolving data in two stages: i. *Micro* clustering: that processes the data online efficiently by grouping data points into micro clusters, and composing statistic summaries at snapshots in time. ii. *Macro* clustering is performed offline using the aforementioned summary together with additional user input in order to produce the final clustering. The authors also state that using a time window allows a better understanding of the underlying evolving patterns of the data.

The evolution of data stream clustering was built on algorithms that process large datasets. BICO [11] is a clustering algorithm designed to process streaming data. It summarizes the data by computing corsets of the stream to rapidly process the arriving points, and then runs K-means++ on the set of corsets. The state-of-the-art algorithm for data stream clustering is evoStream [4]. It summarizes the input online into micro clusters using a variant of DBSTREAM. A micro cluster is described by its center, last update time, and weights. Clusters that fall within a radius threshold r are merged at every time gap. After establishing γ micro-clusters, the algorithm takes samples from γ to generate the macro clusters. The offline clusterer performs evolutionary steps by merging and modifying existing solutions randomly to improve the final clusters. The experimental evaluation produced state-of-the-art quality while reducing the computational cost. However, the algorithm requires six different parameters, namely: radius, decay rate, cleanup interval, initialization threshold, population size, and the number of clusters.

Similar to the data stream clustering procedure, we propose clustering longitudinal data in two steps: i. Clustering at each time window optimally by choosing the algorithm and parameters that optimize the cluster structures. Afterwards, we summarize each cluster as an entity defined by its center and size. ii. For two consecutive time steps; we apply entity matching to find corresponding clusters, merge the matching clusters, and maintain the result by updating the summary representation with the new size and center.

Entity matching (EM) is a fundamental task in data integration. It aims to recognize varying descriptions of the same object. An entity is defined by an identifier and a set of attributes of the form key-value pairs. Unsupervised EM is

usually performed by calculating pairwise matching scores to create a similarity graph, where a node corresponds to an entity, and an edge between two nodes is weighted by their matching score. This graph is then divided into different partitions, such that all nodes in one partition correspond to matching entities [17]. Markov clustering (MCL) for entity matching [8] was evaluated in Stringer duplicate detection system [14] and produced a high-quality performance.

Disease progression modeling is a prosperous topic of research that has been receiving increased attention in the last two decades. It defines the sequence of possible phases or stages and transitions throughout the disease. It is often initiated at the time of diagnosis, and advances to reflect the evolution of the health status of patients with a chronic illness such as Alzheimer's or cancer [2].

A Markov process consists of a set of states and the transition probabilities among them. It can naturally model a disease trajectory, by representing each phase or stage by a state. Multiple works proposed Markov processes to model disease progression. An HMM [15] used *predefined* stages of abdominal aortic aneurysms to estimate the rates of progression between four stages of the disease. In [25], authors discussed the continuous progression of a disease. They used a Markov Jump Process to model transitions between disease states. The method yields possible variables associated with the transition probabilities.

Alzheimer's disease (AD) is an irreversible chronic neurodegenerative disease in the brain, causing a decline in memory, thinking, language, as well as behavioral changes leading to dementia. There is no known treatment or recovery from AD. Furthermore, Mild Cognitive Impairment (MCI) is an intermediate state between age-associated impairment and AD. Distinguishing between stable MCI caused by aging and progressive MCI caused by AD is of critical importance for early care planning and delaying progression [9].

There is an evident gap between research outcomes and their utilization in medical practices. Most studies focus on optimizing accuracy metrics while neglecting explainability issues [23]. Therefore, we propose a comprehensible and transparent method for subtyping patients, modeling disease progression, and early forecasting of disease progression.

3 Methodology

Given a longitudinal medical dataset, we want to subtype patients, and then model disease progression with a Markov process. Our overall approach is illustrated in Figure 1. Before we delve into the details of our approach, we define some notations used in this paper.

3.1 Preliminaries

The input D is a longitudinal medical dataset of a group of patients $Q = \{q_1, q_2, .., q_b\}$, collected at multiple points in time (follow-up visits). The first visit is referred to as baseline bl. The record of a patient ι at time t is denoted $r_{\iota t} = \{\chi_{1t}, \chi_{2t}, \cdots, \chi_{nt}\}$. It consists of a set of values that indicates the medical status using information such as lab test results, or treatment data.

The set of records for all patients in Q at time t is $R_t = \bigcup_{\iota=1}^{b} r_{\iota t}$. The dataset contains multiple follow-up visits with equal intervals (for example every 3 months), $D = \bigcup_t R_t$. It is important to note that some patients might skip some follow-ups, or drop out of the study, i.e., $|R_{bl}| = b$, however, $|R_e| \leq b, \forall e \neq bl$. A time window $w = [t_s, t_f]$ is a time interval employed to divide the data D into multiple batches B, a batch might contain multiple visits $B_r = \bigcup_{e=s}^{e=f} R_e$.

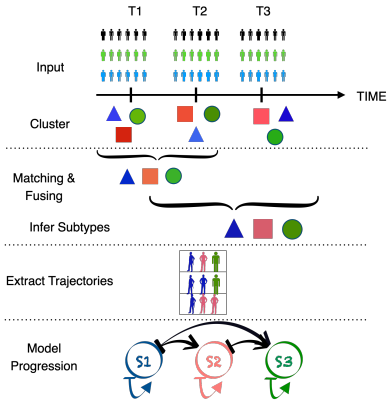


Fig. 1. Cluster Matching for Modelling Disease Progression

An *entity* is a description of an object which consists of an identifier, and a set of properties. Entity matching (`EM`) is the task of finding multiple representations of the same entity. Cluster c_i matches c_j written, $c_i \approx c_j$, means they refer to the same subtype.

3.2 Clustering Longitudinal Data

Our approach to cluster evolving data and infer Markov states that represent patients subtypes can be summarized as follows:

1. Prepare D by imputing missing values and selecting relevant features.
2. Divide D using w into multiple batches $D = \{B_1, B_2, \cdots B_h\}$, such that $h = \frac{\text{study duration}}{w}$. Let $C = \{\}$ be the subtyping result, updated incrementally until the final batch.
3. For each batch B_i:
   - Find optimal clustering algorithm with its parameters by optimizing Silhouette index. Determine k by detecting the knee point using [22].
   - Cluster B_i into $C_i = \{c_{i1}, \cdots, c_{ik}\}$.
   - Calculate the center of each cluster μ_{ik} using average, and its size $ppt_{ik} = \frac{|c_{ik}|}{|B_i|}$. Then, define each cluster as an entity: $c_{ik} = (ppt_{ik}, \mu_{ik})$.
4. For two sequential batches B_i, B_{i+1}, apply `EM` on all pairs of clusters in $C_i \times C_{i+1}$:
   - If matching clusters $c_{ix} \approx c_{jy}$ are found, fuse their points to create a unified cluster entity $c_{ix-jy} = (ppt_{ix-jy}, \mu_{ix-jy})$.

Term	Description
D	Longitudinal dataset
Q	Set of patients
b	Number of patients
bl	First visit
$r_{\iota t}$	Record of patient ι at t
R_t	Set of records at t
w	Time window
B_i	Data batch
C_i	Set of clusters of B_i
ppt_{ik}	$\frac{\lvert c_{ik}\rvert}{\lvert B_i\rvert}$ Size of of c_{ik}.
μ_{ik}	Center of c_{ik}
θ	Graph threshold
τ_ι	Trajectory of q_ι
π	Initial states distribution
P	Transition matrix

Fig. 2. Table of notations

- Give the same label l to all points in c_{ix}, c_{jy}.
- $ppt_{ix-jy} = \frac{|c_{ix}|+|c_{jy}|}{|B_i|+|B_j|}$, $\mu_{ix-jy} = \frac{\mu_{ix}+\mu_{jy}}{2}$.
- $C = C \bigcup \{c_{ix-jy}\}$.

– $C = C \bigcup \{c_{i\vartheta}\}$, for all clusters $c_{i\vartheta}$ with no match.

5. Repeat step 4 on C and the following data batch, until the final batch B_h.

To find matching clusters using `EM` in two consecutive batches B_i, B_{i+1}: **First**, create a similarity graph for cluster entities in $C_i \times C_{i+1}$, where the nodes of the graph are the cluster entities, and the edges are weighted with the matching scores of the adjacent nodes. For a pair of clusters (x, y), the matching score is calculated from euclidean distance using: $m(x, y) = \frac{1}{1+\sqrt{(ppt_x - ppt_y)^2 + \sum_{o=1}^{n}(\mu_{xo} - \mu_{yo})^2}}$

Then, remove the edges with matching scores less than a threshold θ. **Finally,** find matching subtypes using graph partitioning by flow simulation [8].

3.3 Modeling Disease Progression

We formulate disease progression as a longitudinal clustering problem of patients data. Thus, the clusters correspond to the disease stages.

A *Markov process* (MP) is defined by $< S, \pi, P >$ such that: S is a finite set of states. π is the initial state distribution. And P is a state transition probability matrix: $P_{s\hat{s}} = P[S_{t+1} = \hat{s}|S_t = s]$.

We deploy a Markov process to represent disease progression. To define the MP, the states S correspond to the disease stages $S = C$. The transition matrix P approximates the probabilities of progressing from one stage to the subsequent one. Let $|S| = \aleph$, and $n_{\dot{a}\dot{b}}$ the number of observed transitions between states $\dot{a}$, and $\dot{b}$, we deploy the following steps:

1. Give a unique label l_v for each state in S.
2. For each patient, extract the predicted trajectory of labeled states, such that: $\forall q_\iota \in Q, \tau_\iota = [l_1, \cdots, l_e]$.
3. From the set of trajectories $\mathcal{T} = \{\tau_1, \cdots \tau_b\}$, estimate the initial states distribution π, and the transition probability between two states $\dot{a} \longrightarrow \dot{b}$ $\hat{p}_{\dot{a}\dot{b}} = p[S_{t+1} = \dot{b}|S_t = \dot{a}] = \frac{n_{\dot{a}\dot{b}}}{\sum_{x=1}^{\aleph} n_{\dot{a}x}}$.

The resulting Markov process represents the model of disease progression.

4 Experiments

We conduct our experiments on a real-world dataset of Alzheimer's disease. The goals of our experimental evaluation are manifold: 1. Assessing the performance of our approach as a data clustering method over multivariate longitudinal input, and comparing with the state-of-the-art approaches of data stream clustering in terms of internal and external cluster validation metrics. 2. Subtyping Alzheimer's disease patients, and inferring the characteristics of each subtype.

3. Demonstrating the capability of the model to distinguish between stable and progressive patients.

Experimental Setup: Our approach is implemented in Python 3.9[1]. For clustering, we use the implementation in sklearn[2]. For entity matching, we use the original source code of Markov clustering by Stijn van Dongen[3]. For baseline methods, we use the implementation in the R package `"stream"` [12][4].

4.1 Alzheimer's Disease Dataset

We conduct our experiments on the data originating from The Alzheimer's Disease Neuroimaging Initiative(ADNI) [26][5]. The dataset has been collected over 10 years for subjects with inherited risk of developing Alzheimer's disease, with an interval of 6 months between two follow-up visits.

Variable	Number Patients
Age	
$[54, 64]$	175
$]64, 74]$	702
$]74, 84]$	701
$]84, 92]$	143
Gender	
Male	954
Female	767
Diagnosis at Baseline	
Normal	521
MCI	864
Dementia	336

Fig. 3. Demographic & Diagnosis Details

At each visit, the patient's measurements are registered in their record, along with a diagnosis of either Normal, Mild Cognitive Impairment (MCI), or Dementia. Mild Cognitive Impairment (MCI) is an intermediate state between age-related decline in memory and thinking, and the more alarming deterioration caused by Alzheimer's disease. Classifying MCI subjects between stable patients (sMCI), and patients progressing towards Alzheimer's (pMCI) is crucial for early planning of AD treatment [2].

We use the average for imputing the missing values. Normalization is done using min-max. Patients with missing diagnosis were excluded. The number of retained observations after standard data cleaning was 8332 records for 1721 patients, with an average of 5 records per patient. Age, gender, and baseline diagnosis details of the cohort are shown in Figure 3. We select a set of variables known in the literature to be informative for Alzheimer's. The choice of features is crucial for calibrating the performance of the model. The selected features also include a unique identifier for each patient used to extract the trajectories, and a visit code indicating the number of months after the baseline visit used to split the data into batches. Age and gender were only included to report the demographic information. The clustering input features with their descriptions are shown in Table 1.

[1] https://github.com/Loujainl/Longitudinal-Clustering
[2] https://scikit-learn.org/stable/modules/clustering.html
[3] https://github.com/GuyAllard/markov_clustering
[4] https://github.com/mhahsler/stream
[5] For more details, please refer to: https://adni.loni.usc.edu/

Table 1. Selected Features from ADNI Dataset

Variable	Description	Type
Entorhinal	Brain region where AD damage appears	MRI
Hippocampus	Brain complex structure	MRI
CDRSB	Clinical Dementia Rating Scale Sum of Boxes	Cognitive score
ADAS 11& 13	Alzheimer's Disease Assessment Scale	Cognitive score
MMSE	Mini-Mental State Exam	Cognitive test
FAQ	Functional Assessment Questionnaire	Cognitive test
RALVT_immediate	Rey Auditory Verbal Learning Test	Cognitive test
FDG-PET	F-fluroDeoxyGlucose-Positron Emission Tomography	Imaging

Using a time window w of two years, the data was divided into 5 batches. $|B_1| = 6119$, $|B_2| = 1293$, $|B_3| = 511$, $|B_4| = 276$, and $|B_5| = 133$.

To find the optimal clustering for each data batch, we test the following algorithms ; KMeans for partitional clustering, DBSCAN for density, and agglomerative for hierarchical clustering. KMeans centroids were initialized randomly, K was determined by detecting the knee point of the average Silhouette curve obtained for different values of K. For DBSCAN $eps = 0.5$ and $minPts = 5$. Finally, we used *'ward'* linkage criterion in hierarchical clustering. The best performance on Alzheimer's data with respect to silhouette index was consistently obtained by KMeans with K=2 over all batches. We use a graph threshold $\theta = 0.7$.

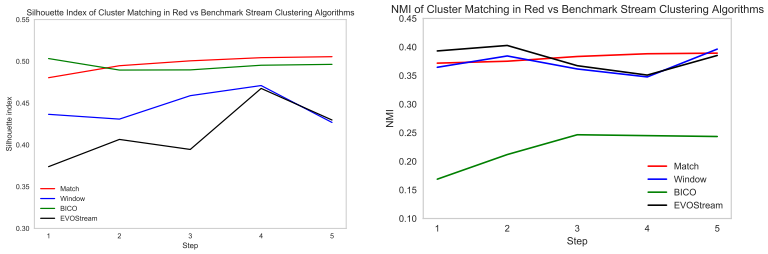
In order to assess the performance of our approach for clustering longitudinal evolving patient data, we compare it with the state-of-the-art data streaming algorithms: evoStream [4] and BICO [11]. We also combine two stages; a sliding window [13] which represents each point as a *micro* cluster with KMeans as a *macro* clusterer. For all baselines, we use the default values of the parameters. We use the same number of clusters found by our approach i.e., K=2.

4.2 Results

We measure the silhouette coefficient as a validity criterion for compactness and separation of resulting clusters, as it showed robustness and better performance against various clustering criteria [24, 27].

In order to assess the performance of the methods, we compare the evolution of the quality while progressing with the consecutive batches. The resulting performance of our approach and all baselines with respect to the silhouette index is illustrated in Figure 4(a). Evidently, our approach outperforms all baselines albeit by just $\sim 2\%$ over BICO.

The external quality validation with Normalized Mutual Information (NMI) is computed using the available diagnosis as the true label. The results of our approach and baselines are shown in Figure 4(b). We can see that the accuracy of our approach improves steadily with every step and achieves similar accuracy to evoStream and window-kmeans. Although BICO yields a good silhouette score, it outputs poor performance with respect to NMI.



(a) Silhouette (b) NMI

Fig. 4. Cluster Quality of Our Approach in Red and Baselines

The two subtypes found by our approach are illustrated with a heat map in Figure 5, with the label 0 referring to No Alzheimer's disease (No-AD), and 1 corresponding to AD. The values represent the cluster mean after normalisation. The most significant feature that distinguishes between the two subtypes is FAQ, which is a questionnaire with an outcome in the range $[0, 30]$, where 0 indicates no impairment, and higher values reflect severe impairment. Similarly, CDRSB ranges between 0 (cognitively normal), and 18 (severe impairment). This indicates that the results of our approach correspond with the scores assigned by the cognitive tests.

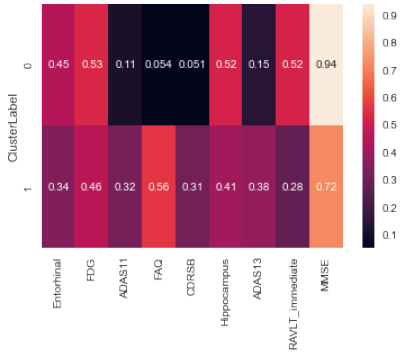


Fig. 5. Heat map illustration of the subtypes results

We report the confusion matrix for all records with the two discovered subtypes: No-AD, and AD, against the actual diagnosis (Normal, MCI, Dementia) in Figure 6.

We calculate the accuracy on Normal and Dementia diagnosis using standard metrics: precision, recall and F1 score: $P = \frac{tp}{tp+fp}, R = \frac{tp}{tp+fn}, F1 = 2 \times \frac{P \times R}{P+R}$.

Diagnosis	No-AD (0)	AD (1)
Normal	2668	0
MCI	3496	436
Dementia	194	1538

Fig. 6. Confusion Matrix

P[Normal] $= \frac{2668}{2668+194} = 0.932$, R[Normal]$= \frac{2668}{2668+0} = 1$, F1[Normal] $= 2 \times \frac{0.932 \times 1}{0.932+1} = 0.965$.

P[Dementia] $= \frac{1538}{1538+0} = 1$, R[Dementia]$= \frac{1538}{1538+194} = 0.888$, F1[Dementia] $= 2 \times \frac{1 \times 0.8879}{1+0.888} = 0.941$.

4.3 Forecasting MCI Progression to Dementia

We study here the ability of our approach to distinguish between stable and progressive MCI, represented by the subtypes No-AD (0), and AD (1) respectively. We retrieve 929 unique patients with MCI diagnosis, and extract the real diagnosis trajectories and the clustering trajectories from the first to the last visit. The trajectories were summarized into length three τ =[`start, middle, final`], such that `start` refers to the baseline visit entry, the `middle` entry refers to the first change in diagnosis/cluster, and the `final` entry denotes the last visit diagnosis/cluster in the original trajectory. We assess our results as follows:

- True positive instances: 733 patients in total. We distinguish three cases:
  - No Alzheimer's, representing stable MCI patients, with no Dementia diagnosis at any visit. For example : the real trajectory is: `[MCI, MCI, MCI]`, and clustering is: `[0,0,0]`. We detected 536 patients.
  - Synchronized forecasting, is when the detection of the subtype AD occurs at the same time of diagnosing Dementia. For example, the real trajectory is: `[MCI, MCI, Dementia]`, and clustering is: `[0, 0, 1]`. We detected 79 patients.
  - Early forecasting, is when the detection of AD subtype occurs before diagnosing Dementia. For example, the real trajectory is: `[MCI, MCI, Dementia]`, and the clustering is: `[0, 1, 1]`. We detected 118 patients. Our model was able to detect Alzheimer's progression before the real diagnosis with an average of 2 years.
- False positive, is when we detect AD subtype, yet the real diagnosis is consistently MCI. For example, real trajectory is: `[MCI, MCI, MCI]`, and the clustering is: `[0, 0, 1]`. We detected 178 patients.
- False negative, is when the forecasting is No-AD, while the actual diagnosis is progressing to Dementia. For example, real trajectory is: `[MCI, MCI, Dementia]`, and the clustering is: `[0, 0, 0]`. We detected 18 patients.

We can calculate the accuracy on MCI: P[MCI] $= \frac{733}{733+178} = 0.805$, R[MCI]$= \frac{733}{733+18} = 0.976$, F1[MCI]$=2 \times \frac{0.805 \times 0.976}{0.805+0.976} = 0.882$.

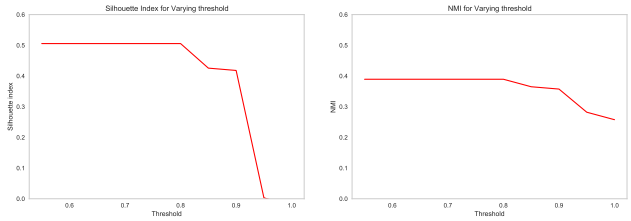
Overall, we conclude that our approach yields high accuracy for forecasting Alzheimer's disease progression.

4.4 Sensitivity Analysis

The only configuration parameter that influences the performance of our method is θ threshold, whose value can impact the graph partitioning of entity matching and consequently, the resulting stages. To evaluate its effect on the results, we test the values in $[0.5, 1]$ with a step of 0.05. The result is shown in Figure 7.

We observe that up to 0.8, the performance is not affected by the value of θ. However, for $\theta > 0.85$, the silhouette index decreases. This is due to the fact that excessive values demand the entities to be almost identical, which is unlikely to be found in clusters of longitudinal data.

Overall, we can conclude that our approach is robust with respect to the threshold, with $\theta = 0.7$ constituting a reliable default value.



(a) Silhouette (b) NMI

Fig. 7. Cluster Quality with Varying Graph Threshold θ

5 Conclusions

We propose a transparent approach to cluster multivariate longitudinal data which can be deployed for patient subtyping and modelling chronic disease progression. Our method is based on optimised clustering to infer subtypes within a time window, then finding matching subtypes in subsequent visits using a borrowed technique from data integration. We conducted experiments on a real-world dataset of individuals at risk of developing Alzheimer's disease, compared with the state-of-the-art data stream clustering methods, and evaluated the capacity of our approach for early forecasting of the disease.

For future work, we plan to test the approach on modeling cancer progression based on real-world data of patients undergoing immunotherapy treatment.

References

1. Aggarwal, C.C., et al.: A framework for clustering evolving data streams. In: VLDB. pp. 81–92. Elsevier (2003)
2. Ansart, M., et al.: Predicting the progression of mild cognitive impairment using machine learning: a systematic, quantitative and critical review. Medical Image Analysis **67**, 101848 (2021)
3. Baytas, I.M., et al.: Patient subtyping via time-aware lstm networks. In: Proceedings of the 23rd ACM SIGKDD. pp. 65–74 (2017)
4. Carnein, M., Trautmann, H.: evostream–evolutionary stream clustering utilizing idle times. Big data research **14**, 101–111 (2018)
5. Chen, H.H., et al.: A markov chain method to estimate the tumour progression rate from preclinical to clinical phase, sensitivity and positive predictive value for mammography in breast cancer. Journal of the Royal Statistical Society:
6. Choi, E., et al.: Doctor ai: Predicting clinical events via recurrent neural networks. In: Machine learning for healthcare conference. pp. 301–318. PMLR (2016)
7. Delen, D., et al: Predicting breast cancer survivability: a comparison of methods. Artificial intelligence in medicine **34**(2), 113–127 (2005)
8. Dongen, S.V.: Graph clustering by flow simulation. PhD thesis, University of Utrecht (2000)

9. El-Sappagh, S., et al.: A multilayer multimodal detection and prediction model based on explainable artificial intelligence for alzheimer's disease. Scientific reports **11**(1), 1–26 (2021)
10. Escudero, J., et al.: Early detection and characterization of alzheimer's disease in clinical scenarios using bioprofile concepts and k-means. In: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. pp. 6470–6473. IEEE (2011)
11. Fichtenberger, H., et al.: Bico: Birch meets coresets for k-means clustering. In: European symposium on Algorithms. pp. 481–492. Springer (2013)
12. Hahsler, M., et al.: Introduction to stream: An extensible framework for data stream clustering research with r. Journal of Statistical Software **76**(14), 1–50 (2017). https://doi.org/10.18637/jss.v076.i14
13. Hahsler, M., Dunham, M.H.: remm: Extensible markov model for data stream clustering in r. Journal of Statistical Software **35**, 1–31 (2010)
14. Hassanzadeh, O., et al.: Framework for evaluating clustering algorithms in duplicate detection. VLDB **2**(1), 1282–1293 (2009)
15. Jackson, C.H., et al.: Multistate markov models for disease progression with classification error. Journal of the Royal Statistical Society: (The Statistician) **52**(2), 193–209 (2003)
16. Lee, S.H., et al: Parkinson's disease subtyping using clinical features and biomarkers: literature review and study of subtype clustering. Diagnostics **12**(1) (2022)
17. Liekah, L., Papadakis, G.: Deduplication over heterogeneous attribute types (d-hat). In: International Conference on Advanced Data Mining and Applications. pp. 379–391. Springer (2022)
18. Miotto, R., et al.: Deep patient: an unsupervised representation to predict the future of patients from the ehrs. Scientific reports **6**(1), 1–10 (2016)
19. Nilashi, M., et al.: Accuracy improvement for predicting parkinson's disease progression. Scientific reports **6**(1), 1–18 (2016)
20. Oxtoby, N.P., et al.: Data-driven models of dominantly-inherited alzheimer's disease progression. Brain **141**(5), 1529–1544 (2018)
21. Saria, S., Goldenberg, A.: Subtyping: What it is and its role in precision medicine. IEEE Intelligent Systems **30**(4), 70–75 (2015)
22. Satopaa, V., et al.: Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In: 2011 31st international conference on distributed computing systems workshops. pp. 166–171. IEEE (2011)
23. Spasov, S., et al.: A parameter-efficient deep learning approach to predict conversion from mci to alzheimer's disease. Neuroimage **189**, 276–287 (2019)
24. Vendramin, L., et al.: Relative clustering validity criteria: A comparative overview. Statistical analysis and data mining: the ASA journal **3**(4), 209–235 (2010)
25. Wang, X., et al.: Unsupervised learning of disease progression models. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 85–94 (2014)
26. Weiner, M.W., et al.: Recent publications from the alzheimer's disease neuroimaging initiative: Reviewing progress toward improved ad clinical trials. Alzheimer's & Dementia **13**(4), e1–e85 (2017)
27. Wiwie, C., et al.: Comparing the performance of biomedical clustering methods. Nature methods **12**(11), 1033–1038 (2015)
28. Yadav, P., et al.: Mining electronic health records (ehrs) a survey. ACM Computing Surveys (CSUR) **50**(6), 1–40 (2018)
29. Yan, Y., Harris Jr, F.C.: A survey of data clustering for cancer subtyping. International Journal for Computers and Their Applications **28**(2), 1–13 (2021)

Learning without real data, a 3D data simulation learning approach applied to ID cards segmentation and text extraction

Edouard Bertrand, Anaïs Druart, Axel Thévenot, and Christophe Rodrigues

Léonard De Vinci Pôle Universitaire, Research Center, 92916 Paris La Défense, France
edouardbertrand100@gmail.com, anais.druart@gmail.com,
christophe.rodrigues@devinci.fr

Abstract. With the multiplication of successful deep learning techniques for the detection of objects in images and since the ImageNet project, the need for massively annotated data has continued to grow. But how to deal with situations in which these annotations are not available, or worse, the input data themselves do not exist, preventing any self-supervision or learning? In this paper, we focus on the particular case of text extraction from French national ID cards. Current Optical Character Recognition (OCR) techniques based on deep learning show great success in the field of text extraction, but their results can be mitigated if the scan is done from a smartphone due to the great variability of angles of view, lighting, camera quality... We propose an approach based on the projection of simulated data into a 3D environment, which allows us to return to a supervised learning framework. However, even if we control all the parameters of the simulation, we cannot guarantee that the generated data is representative of real-life pictures. This is why we guide the creation of examples using active learning to explore the simulation space where card readability is limited. We detail the approach and show empirically its interest with a synthetic evaluation on the simulated datasets and a qualitative evaluation on real ID cards.

1 Introduction

The ImageNet project [1] has allowed for considerable progress in terms of image classification and object identification, thanks to models based on deep learning techniques. However, to achieve these results, a large amount of data had to be manually annotated. Specifically, almost 15 million images were annotated by more than 25,000 crowd workers via Amazon Mechanical Turk. This extraordinary annotation effort is costly and may need to be repeated depending on the task at hand, which is not feasible in most cases.

In this work we are interested in extracting information from structured documents. OCR systems exist to automatically extract text from such documents, but in the case where the scanning is done by a smartphone, the performance of the OCR deteriorates because it is disturbed by several factors such as the quality of the camera, the ambient brightness, the distance, the angle of the shot,

etc. It would be useful to train the OCR systems with real data representing these different difficulties encountered in real cases. It would be useful to train OCR systems on real data representing these different difficulties encountered in real cases. But how can we do this if such data does not exist?

In the absence of data, we propose to create a simulation that allows the projection of any synthetic structured document in a 3D environment. This way we can reproduce and control the different difficulties that would be encountered with a real picture of a document. We choose to focus on the case of image segmentation of French national ID cards, since as far as we know there is no dataset of ID card scans available online, making them a perfect use case to study learning with no real data. Such model could be used for automatic field filling of digital forms or databases for fast identification. Another application is ID authentification on applications offering peer to peer services that require identity verification.

However, by fully simulating the data, we face the problem of the representativeness of the generated examples. In order to minimise this risk, we propose to use an active learning procedure guided by readability to automatically set the different parameters of the simulation and to cover the most realistic places of the simulation space as well as possible. The main contributions of this work are the following: **(1)** Building a pipeline to create a fully annotated realistic dataset of synthetic images to train models for information extraction in structured documents. **(2)** Creation of a model capable of localizing an ID card in an image, allowing the cropping and straightening of the card for improved OCR text extraction. **(3)** Sharing of a reusable public dataset of realistic synthetic French ID cards fully annotated in terms of textual content as well as position information allowing other researchers to train their own models.

The structure of the paper is as follows: we start by discussing the state of the art in Section 2, then we present our approach in Section 3. We then present an experimental protocol in Section 4, discuss our results in Section 5 and end with our conclusions and perspectives.

2 Related works

Close to our work, the DeepFlash paper [2], presents a U-Net model to transform a phone photo with flash into a qualitative portrait photo. Similarly, the edge-sensitive approach of Fan's paper [3] trains its models with a wide range of luminosity and background examples in the synthetic images, as in [4]. To this day there already exist multiple physical systems able to decipher the information on an ID card. These systems are very efficient at localizing the cards different parts and collecting their textual information, as in [5]. Some of them even permit the optical reading of punched or opaque cards like [6]. However, these tools need bulky physical machines that are not accessible to everyone and cannot be easily reproduced. In [7], text is extracted from ID cards using an OCR and NLP methods, but only from perfectly cropped and scanned images. In [8] a model is proposed to reduce the need for real images. Instead of having

several different images of the same scene with different lighting angles, they suggest adding a depth sensor as an input to be able to reconstruct (through learning) the same image from different angles.

Fig. 1. Digital french ID card twin at the generation step and the Blender simulation step. Content segmentation is highlighted with color boxes.

There also exist segmentation algorithms like [10] based on image projection for complex text layout with or without an OCR. However, they are not fitted to be used in real-time and their adaptability is low. At our knowledge, there is no available public dataset of French ID cards. The sensible nature of such personal information does not allow for easy storage, publication or reuse of data. [11] proposes an alternative that uses the Wikipedia illustrative ID cards samples of different countries. The images are printed and laminated to be taken from a smartphone in different situations with variations in distance, angle, light, reflection and background. This dataset has two main problems, the human selection bias and the limited diversity of the dataset. We aim to solve these two problems by using a simulated environment. Machine learning applications involving image datasets commonly use image augmentation [12] to help reduce overfitting. Data augmentation suppose a minimum of data available. Only few works like [13] focuses on the case where no data are available where a real-world robotic object detector is trained by only using synthetic data. Also, [14] are able to generate realistic endoscopic video datasets efficiently for validating surgical vision algorithms, by using the Blender software.

3 Propositions

Here we describe how we generate data synthetically. We then explain how we use active learning to improve the generation process. Finally, we describe how we use neural networks to perform a supervised segmentation task on this data.

3.1 Dataset creation pipeline

The dataset creation pipeline key steps are illustrated on Figure 2. We first construct a blank ID card layout based on a Wikipedia sample, keeping only invariant text as illustrated on Figure 2a. The background around the text is sampled from neighborhood. The profile image is then generated using StyleGAN model [15] able to create randomly realistic faces. The textual fields on the front

of the cards are filled with random characters and numbers. These fields include ID identifier, name, surname(s), sex, birthdate, birthplace, height and bandcodes (two strings at the bottom). This allows us to generate realistic ID cards on demand as illustrated in Figure 2.

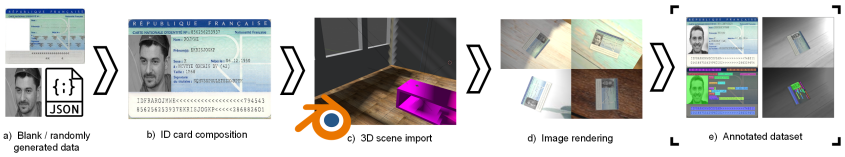


Fig. 2. Dataset creation pipeline key steps

To project the ID cards in a realistic environment we use Blender[1], a free and open-source 3D computer graphics software able to compute rendering on complex scenes. It allows for control of visual effects such as lighting, textures, reflections, backgrounds, blur, materials, translations, rotations, etc. Our 3D scene illustrated in Figure 2c is composed of a camera, an ID card handler containing the ID card and the plastic layer of the card. The room consists of walls, a floor, a roof, a window and two light sources: a sun to simulate natural light and an artificial light. The different parameters involved in the rendering are: camera position, rotation, ID card position, focal length, flash brightness, light orientation, light intensity and desk textures. Ranges of values have been manually defined for all these simulation parameters. Choosing random values within these ranges allows us to generate realistic images of ID cards. Examples of the computed renders are illustrated on Figure 2d. Finally, by controlling each parameter of the 3D scene as well as the card's content, position and content position, we were able to create a fully annotated dataset of 3000 ID cards as illustrated on Figure 1 and Figure 2e. This generated dataset is now publicly available.[2]

3.2 Simulation space exploration with active learning

An important focus of our study is to train our models with data that is as close to reality as possible. By randomly selecting the simulation parameters when controlling the environment, the generated data will be very representative of the diversity of the simulated space. However, these parameters can sometimes turn out to be unrealistic, representing conditions that are either too smooth or too harsh to be comparable to real-world situations. One solution to this problem is to select the simulation parameters through an active learning process guided by the performance of the segmentation models on real data. However, this method requires the generation, training and evaluation of our datasets and

[1] https://www.blender.org/
[2] https://github.com/ResearchPaper0/Learning-without-real-data (anonym)

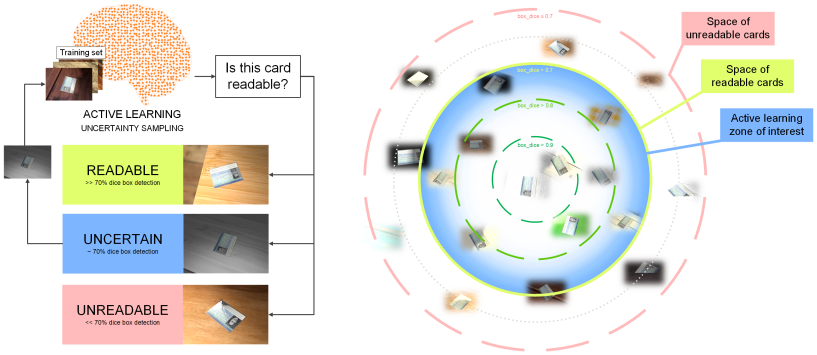


Fig. 3. Active Learning pipeline(left) and Card Readability space(right)

models over multiple iterations, which would take an unreasonable amount of time given the size of our datasets and the number of models. This solution is particularly infeasible in our study, where the real data is extremely sparse.

In order to reduce the cost of active learning, we guide it with the readability of the ID card instead of the model performance, assuming that hardly readable examples might be more interesting for the model if we want to train it on more realistic image conditions. This hypothesis allows us to limit the active learning pipeline to parameter selection only, without seeking feedback from the model training and evaluation. To obtain the card readability value for an example, we first obtain a cropped and straightened image of the ID card in the image using its real coordinates. A dice coefficient is then calculated by calculating the percentage of the text box recognised by our OCR when presented with this image. Below a manually defined threshold (70%), the ID card is considered unreadable. Figure 3 shows the part of the simulation space to explore.

As illustrated on Figure 3, our active learning model follows an uncertainty sampling approach guided by the readability of the ID cards. It is implemented as follows: first, a batch of random pictures is generated using the pipeline described on Figure 2. Then, each ID card picture is labeled as readable or unreadable depending on whether the computed readability value is over 0.7 or not. Each example generated is based on eight adjustable characteristics: accuracy of camera position, distance between camera and ID card, rotation, degree of translation, flash brightness, focal length and ID (x, y) coordinates. These simulation parameters affect the readability of the ID card. Using them as input and the readability labels as output, we are able to train a support vector regression (SVR) model to predict whether or not a set of image parameters will produce a readable ID card image. Using an SVR allows us to see how uncertain the model is in its prediction by looking at how close its output is to 0.5. We use this information to apply uncertainty sampling to a randomly generated set of parameters, selecting the set of parameters whose readability the model is most uncertain about. We then use these parameters to generate interesting, harder-

to-read examples, discard the unreadable ones, and add these new images to the SVR training set for another iteration of the process. We repeat these steps several times, creating a new set of examples that are actively sampled each time. Active learning is repeated until the desired set of readable ID cards is reached. The final SVR training set is then used as the training data set for our segmentation models.

3.3 Supervised machine learning models for image segmentation

To simplify the model and the experimental protocol, we focused on identifying ID cards and text fields. As such, there is no need for extensive training of a dedicated OCR. The first step is to locate the card in the image. Once the corner or edge of the card is detected, we can perform basic image corrections to run the OCR system on the image and recognise the text field. Since we have a dataset of realistic ID card examples that are fully annotated, we can reduce the task to a supervised learning problem. We experimented with three different neural network models to detect the position of the card.

Keypoints Regressor : The first model we use is a supervised Keypoints Regressor model (KPR) as described in [16], whose goal is to learn to retrieve the 4 corners of the ID cards. The model is composed of a stack of convolutional neural networks. The first layer of the model is a ResNet-50 [17] pretrained on ImageNet [1]. The head of the ResNet-50 is replaced by a customized head to get the 8 coordinates predictions. KPR takes as input 448×448×3 RGB images and predicts 8 normalized coordinates for the 4 ID card's corners in the image. KPR model has the advantage of reaching the coordinates even outside of the image which can be especially useful if the ID card image is truncated.

UNet : [18] is a neural network with the shape of the letter U. It was designed for medical image segmentation in a context where few samples were available and data augmentation was performed. Our UNet structure consists of 3 DownBlocks, 2 UpBlocks linked with the symmetrical skip connections from DownBlocks to Upblocks followed by a final HeadBlock. Our UNet model takes as input 128×128×3 RGB images and predicts the 128×128×2 masks corresponding to the probability heatmaps for each pixel to be part of the ID card and to not be part of it. A comparison of these two masks is done to obtain a final binary map of the ID card position. The UpBlocks and DownBlocks have two convolutional layers followed by a batch normalization and a ReLU activation function. The Upblocks are preceded by a 2×2 upsample layer. The output is summed with a 1×1 residual convolutional layer preceded by a 2×2 average pooling for the Donwblock. Lastly, the HeadBlock is the same as the UpBlocks without the residual layer and its final activation function is the channel wise softmax. In order to crop and staighten the image for the OCR reading, we mark the four corners of the cards as the four corners of the rectangle of smallest area that contains all the predicted ID cards pixels.

Edge-based UNet : We introduce a variation of the UNet called Edge-Based UNet designed to extract the 4 corners of ID cards. This model has the same architecture as our Unet model, except we replace the two output masks by four

masks representing the probability distribution of each pixel in the image to correspond to a corner according to the model's prediction. The four corners are then annotated following the brightest pixel's location on each mask.

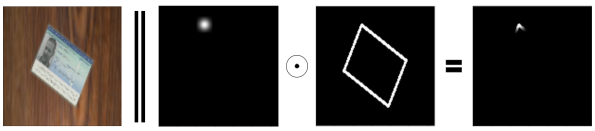


Fig. 4. Target mask example of the top left ID card corner. The four corner masks (right) are the Hadamard product of the keypoint Gaussian heatmap (left) by the edge contours masks (center).

To create the ground truth corner heatmaps used for training, we first compute the Gaussian heatmaps of each corner from its coordinates and draw the ID card outline mask for the image. We then compute the Hadamard product of these two heatmaps to obtain the target masks. In this way, we obtain a training sample that allows a more representative learning of the edges of the ID cards. An example is illustrated on Figure 4.

4 Experimental study

We conduct our experimental study in order to evaluate three important aspects of our approach: **(1)** The performance of our different models given the task of ID card segmentation in an image, especially the difference of performance between our original model Edge-based Unet in comparison with more conventional models such as a basic Unet or our Key Points Regressor model (basic CNN). **(2)** The validity of our simulation approach for training dataset creation in situations where very few training data is available compared to the well-established data augmentation pipeline. **(3)** The impact of guiding the dataset creation process with active learning compared to random sampling.

4.1 Baselines

Raw : Our first model to evaluate is the naive model without contour detection. We call this the Empty Model. This model gives us a lower baseline that allows us to evaluate the effect of applying an OCR system directly to the raw image. The difference between any model and the application of the raw model will show the interest of the approach. **Oracle :** Unlike the Raw model, the Oracle model knows the exact coordinates of the ID cards generated by our simulations. When used to evaluate an image, it returns the four exact corners of the card. Due to the perspective angle of the ID card shot, these points can take the form of a trapezium. In order to obtain a rectangle that is horizontally aligned with

the edges of the image and more easily interpreted by the OCR, we apply a geometric perspective transformation. This transformation is applied to all the models described, except for the raw model, where no transformation is applied at all. We consider Oracle's output to be the best possible result achievable by our models during evaluation. **Canny :** Canny Edge Detection [19] is a baseline approach useful for edge detection. This algorithm uses the intensity of gradients in the image to detect contours. First, we apply a bilateral filter, which allows us to remove noise while keeping edges sharp. The canny filter allows us to extract the contours in the image, to which we apply a dilation. We then extract the contours of the image and assume that the contour with the largest area is the ID card. Finally, we approximate this contour by a trapezoid whose sides correspond to the four predicted corners of the ID card in the image.

4.2 Training Datasets

The differents Neural network models proposed (KPR, Unet, EB-Unet) described Section 3.3 are trained to directly predict ID cards position. We also describe a pipeline to generate synthetic data. In consequence, we need to evaluate this models but also the impact of the generated datasets on training. All datasets are composed of 3000 ID cards for training the models. The differents dataset studied are the following: **Random Sampling (RS) :** ID cards are generated and projected randomly on the 3D simulation (all simulation parameters like position, rotation, angle, light,... are selected randomly). **Active Learning (AL) :** As described in Section 3.2, an amount of examples are actively and iteratively selected in order to emphasize the dataset creation on supposed interesting zones of the space of readable ID cards. 10% of the total amount of cards is selected randomly, the remainder is generated actively. **Data Augmentation (DA) :** This dataset is created using state of the art Data Augmentations techniques as described in [21].We started from 4 real pictures of 4 differents ID cards[3] to which we applied various transformations including rotation, translation, shifting, zooming, shearing, flipping, color mingling, cropping and noise injection. This is a common method used to create more examples when very few training data are available. **Hybrid (RS+DA) :** This dataset combines half simulated images generated through random sampling, and half images produced using data augmentation. **Hybrid (AL+DA) :** This dataset combines half active learned images and half images produced using data augmentation.

4.3 Metrics

Since we are interested in improving the output of an OCR by using image segmentation to improve its input, we evaluate our models on two different metrics that represent the steps of the text extraction process using our method.

Dice : To evaluate the quality of the ID card segmentation regarding the text box detection after the image is straightened and cropped, we use an open-source OCR project based on CTPN [20] which can accurately localize text lines

[3] We used our personal ID cards.

in an image. We compute the Dice coefficient [22] of the text zones locations predicted by the OCR when presented the cropped image.

Once the four corners of the card have been found by the model, the straightening is achieved by calculating the perspective matrix of the corners in the image and warping the image to this matrix, so that the trapezoid predicted for the card now becomes rectangular and takes up the entire size of the image.

Jaro : To evaluate the quality of the text extraction, we use the open-source OCR Tesseract[4] that supports French language. We calculate the Jaro distance between the predicted text and the real text of the ID card. The distance decreases as more characters of the predicted string match the real one.

4.4 Evaluation Datasets

We evaluated our models on three different datasets. The first two datasets make up the synthetic evaluation. They each consist of 5000 images of ID cards generated using our 3D simulation approach with random sampling and data augmentation. The purpose is to evaluate our models on a large representative set of data and to analyse the influence of different evaluation environments. The third dataset is used for qualitative evaluation. It consists of 100 images of real ID cards of 10 different people, taken with a mobile phone camera in different contexts. It allows us to evaluate our models on real data, which is essential to verify that the simulated data used for training is representative of reality and that the models' performance on the synthetic evaluation datasets translates well to real-life examples. In contrast to the first two datasets, we do not know the exact locations of the text boxes for the real-life dataset, as it would have been far too costly to manually annotate these coordinates. Therefore, we were not able to evaluate the dice on the qualitative dataset.

5 Experimental Results

Table 1 describes the results of our evaluation pipeline for an ID card example. It qualitatively summarises the full pipeline results, showing the ID card corner predictions and their order, as well as the pixel-wise segmentation confusion from the prediction. The ID card is then extracted from this segmentation and passed to the CTPN OCR to localise the text boxes within the ID card. As shown in Table 1 we can see with the Raw model that considering the ID card as the full image is not accurate. This proves the need to first detect and crop the ID card within the image to get more accurate results. The Canny model seems to get a better idea of the ID card's location, although it sometimes misses corners and cannot give a precise order for them, which can lead to unwanted vertical or horizontal flipping of the card. This shows the importance of getting the full positional information of the 4 corners. Looking at the results in Table 2, we can see that our EB-Unet model obtains the best results for all

[4] https://github.com/tesseract-ocr/tesseract

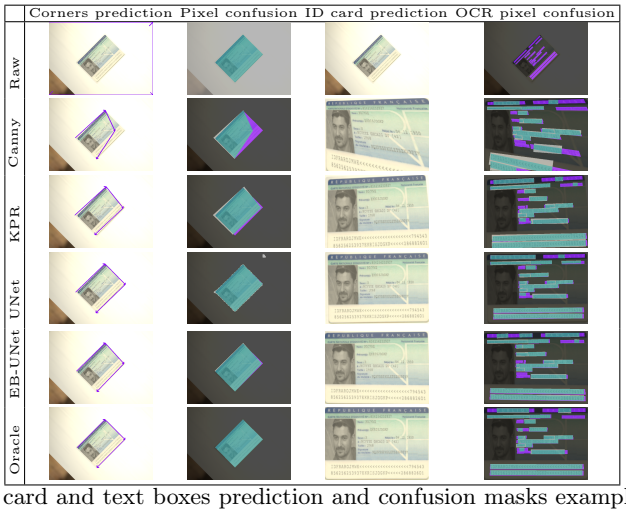


Table 1. ID card and text boxes prediction and confusion masks example for a given digital french ID card twin at each evaluation pipeline step. True Positive (blue), False positive (grey), and False Negative (purple).

datasets on the metrics of Dice and Jaro. On real data, our EB-Unet Hybrid AL+DA model shows much better results than all other models, which seems to confirm the relevance of our contribution. **Simulation vs Augmentation:**

Models	Dice Evaluation		Jaro Evaluation		
	Synthetic Dataset	DA Dataset	Synthetic Dataset	DA Dataset	Real Dataset
Raw	0.080015	0.075503	0.014071	0.065077	0.302591
Canny	0.261582	0.117614	0.253295	0.110559	0.192075
KPR RS	0.576401	0.320820	0.280018	0.134209	0.243554
KPR AL	0.588218	0.326579	0.324815	0.135301	0.253568
KPR DA	0.308425	0.463260	0.128293	0.227250	0.224553
KPR RS+DA	0.456271	0.448648	0.227862	0.225237	0.272988
KPR AL+DA	0.548479	0.394319	0.270078	0.193477	0.240959
U-Net RS	0.446206	0.285383	0.377550	0.279690	0.353886
U-Net AL	0.446914	0.233374	0.389795	0.206026	0.358050
U-Net DA	0.279221	0.331805	0.249823	0.321525	0.365210
U-Net RS+DA	0.449694	0.328581	0.380301	0.318117	0.419871
U-Net AL+DA	0.457636	0.322958	0.397315	0.315604	0.411983
EB-Unet RS	0.579265	0.310245	0.394510	0.199747	0.381991
EB-Unet AL	**0.620448**	0.268437	**0.416357**	0.178608	0.400699
EB-Unet DA	0.107115	0.516300	0.082561	0.329897	0.305460
EB-Unet RS+DA	0.523831	**0.563217**	0.343238	0.308663	0.438738
EB-Unet AL+DA	0.616479	0.540640	0.405847	**0.342679**	**0.464733**
Oracle	0.658440	0.654548	0.460686	0.404859	0.549665

Table 2. Image segmentation (Dice) and text extraction (Jaro) evaluation of the different models and datasets benchmarked

The performance of our models on real data seems to show a clear superiority of our dataset generation approach based on data simulation compared to a data augmentation method, proving that our simulation better represents reality. This analysis can also be supported by a cross-reading of the synthetic results: although the simulation models perform better on the simulated dataset and the

data augmentation models on augmented images, we notice that the performance of the simulated models is much less degraded when evaluated on the augmented dataset than the performance of the augmented data models evaluated on the simulated dataset, showing a better robustness and adaptability of the models trained on the simulated data. **Random Sampling vs Active Learning :** By comparing the results shown on Table 2 between the Random Sampling and Active Learning models and between our RS+DA and AL+DA hybrid models, we find a superiority of our models trained on the AL data over RS on both the simulated and real data sets. This shows the effectiveness of using our guided data generation method over random sampling to generate training data and confirms our hypothesis that harder to read simulated examples better represent reality for learning.

6 Conclusion

We presented a pipeline to generate a useful simulated dataset for the task of segmenting structured documents which we applied to the use case of French ID cards. We demonstrated that learning is possible even in the absence of real data. We then proposed an active learning approach to explore the simulation space by focusing on examples at the edge of readability, assuming that hard-to-read examples can better represent the difficult conditions under which images are taken in real life. We presented three supervised neural network models, KPR, Unet and EB-Unet, trained to identify ID cards in an image, and found that the EB-UNet model - trained to perform edge detection using a product of the Gaussian heat map of the corners and a mask of the edges of the card - resulted in significantly more efficient localisation of text in ID cards. We conducted synthetic and qualitative evaluations that demonstrated the efficiency of training our EB-Unet model on simulated data in conjunction with augmented data to extract text from ID cards in real images, as well as the relevance of our active learning method to guide the generation of synthetic data. In the future, we would like to generalise the approach to other types of structured documents, such as barcodes or sheet music, and enable the detection and extraction of multiple documents in an image.

References

1. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.
2. Capece, N., Banterle, F., Cignoni, P., Ganovelli, F., Scopigno, R., Erra, U. (2019). Deepflash: Turning a flash selfie into a studio portrait. Signal Processing: Image Communication, 77, 28-39.
3. Fan, Q., Yang, J., Hua, G., Chen, B., Wipf, D. (2017). A generic deep architecture for single image reflection removal and image smoothing. In Proceedings of the IEEE International Conference on Computer Vision (pp. 3238-3247).

4. Moffat, A., *Synthetic training data*,2020.
5. Simon, R., *ID card and tracking system*, US Patent Application Publication, 2003.
6. Nicoud, J., *ID card reader*, United States Patent, 1974.
7. Rusli, F. M., Adhiguna, K. A., Irawan, H., *Indonesian id card extractor using optical character recognition and natural language post-processing* ICoICT, 2021.
8. Qiu, D., Zeng, J., Ke, Z., Sun, W., Yang, C. (2020, November). Towards geometry guided neural relighting with flash photography. In 2020 International Conference on 3D Vision (3DV) (pp. 1137-1146). IEEE.
9. Chen, Q., Koltun, V. (2013). A simple model for intrinsic image decomposition with depth cues. In Proceedings of the IEEE International Conference on Computer Vision (pp. 241-248).
10. Zhu, W., Chen, Q., Wei, C., Li, Z. (2017, October). A segmentation algorithm based on image projection for complex text layout. In AIP Conference Proceedings (Vol. 1890, No. 1, p. 030011). AIP Publishing LLC.
11. K. Bulatov, D. Matalov, and V. Arlazarov, *MIDV-2019: challenges of the modern mobile-based document OCR*, ICMV, 2020.
12. Zhu, J., Ma, H., Feng, J., Dai, L. (2018, April). ID card number detection algorithm based on convolutional neural network. In AIP Conference Proceedings (Vol. 1955, No. 1, p. 040124). AIP Publishing LLC.
13. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P. (2017, September). Domain randomization for transferring deep neural networks from simulation to the real world. In 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 23-30). IEEE.
14. Cartucho, J., Tukra, S., Li, Y., S. Elson, D., Giannarou, S. (2021). VisionBlender: a tool to efficiently generate computer vision datasets for robotic surgery. Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization, 9(4), 331-338.
15. Karras, T., Laine, S., Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4401-4410).
16. Kumar, A., Alavi, A., Chellappa, R. (2017, May). Kepler: Keypoint and pose estimation of unconstrained faces by learning efficient h-cnn regressors. In 2017 12th ieee international conference on automatic face and gesture recognition (fg 2017) (pp. 258-265). IEEE.
17. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
18. Ronneberger, O., Fischer, P., Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.
19. Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6), 679-698.
20. Tian, Z., Huang, W., He, T., He, P., Qiao, Y. (2016, October). Detecting text in natural image with connectionist text proposal network. In European conference on computer vision (pp. 56-72). Springer, Cham.
21. Shorten, C. and Khoshgoftaar, T. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data (6)
22. Dice, L. R. (1945). Measures of the amount of ecologic association between species. Ecology, vol. 26, no. 3, pp. 297–302.

Set Function Representations in a Decision Process: Properties and Interpretation

Eiichiro Takahagi[1]

Senshu University, Kawasaki Kanagawa 2148580, JAPAN
takahagi@isc.senshu-u.ac.jp

Abstract. A set function representation is a method of representing an evaluation score vector using a set function while preserving the rank information of each element. An aggregated set function representation is created by aggregating the set function representations. In this study, three data layers are considered: alternatives, criteria, and experts. A set function representation for an expert and a criterion is created from the alternative evaluation vector. Aggregation with experts creates the set function representation of the criteria that preserves the rank information of the original evaluation vectors. Comparing the set function representations among the criteria allows for a richer analysis than with averages because rank information is included. For example, it is possible to interpret the agreement, disagreement, and polarization of opinions among experts. We define set function representations that can be transformed into each other, and demonstrate the meaning of their Shapley values.

Keywords: Set function representation · Rank information · Shapley value.

1 Introduction

When investigating a dataset consisting of multiple series of valuations, the basic methods are often the analysis of a specific series, such as the mean of a sample, or the relationships among series, such as correlations.

Table 1. Example

	Criteria A1			Criteria A2			Criteria A3		
Alternatives	B1	B2	B3	B1	B2	B3	B1	B2	B3
Expert C1	50	50	50	90	70	20	80	90	60
Expert C2	50	50	50	10	30	80	70	50	40
Total	100	100	100	100	100	100	150	140	100

Table 1 presents the evaluation of the two criteria (A1 and A2), three alternatives (B1,. . .,and B3), and two experts (C1 and C2). For A1 and A2, the total

for all the alternatives is 100, and there is no difference. However, the evaluation values for criteria A1 and A2 are clearly different. For criterion A2, experts C1 and C2 give each alternative in the opposite order. Expert C1 gives higher scores to alternatives B1 and B2, whereas expert C2 gives a higher score to alternative B3. High and low scores cancel each other out, and the total score is the same for all alternatives. As far as the total is concerned, the difference between the experts in A2 disappears.

In this study, we propose defining a set function representation and interpreting the results accordingly. The set function representation of an expert preserves the order of evaluation values of each alternative and the differences between them. The aggregated set function representation of a criterion for each expert also preserves the order and differences among experts.

[9] defines the set function representation; however, the focus is on the analysis of the Choquet integral with respect to a set function, which is defined as the sum of the products of the set function of the evaluation values and the fuzzy measure corresponding to the Choquet integral. This study analyzes the set function representations of the evaluation values. A method for analyzing the set function representation of the evaluation values is discussed.

In [7], a similar definition to this set function representation is given as a Möbius inversion of the interaction operator. The interaction operator using t-norms, and the interaction operator of t-norms using *min* operator is the set function representation in this study.

[4] proposes a method for integrating multi-attribute score data using the set function representation defined in [9]. When interpreting the set function representation, the concept of maximal chains, which is used in lattice theory, is important.

2 Set function representation

2.1 The assumed 3-layers data

In this study, we use 3-layers data such as shown in Table 1. For each criterion (A1,A2,...), set functions of experts are generated, and the set functions are compared; therefore, the criteria are called comparison items. The alternatives are the elements of each set of set functions and are called analysis items. The experts are called aggregation items because they are aggregated. Each score is denoted by $x_{i,j,k}$, where $i, i = 1, \ldots, M$ denotes the number of comparison items; $j, j = 1, \ldots, N$ denotes the number of analysis items; and $k, k = 1, \ldots, N$ denotes the number of aggregation items.

Values must be non-negative and satisfy strong commensurability or be normalized to satisfy it. Satisfying strong commensurability requires that the values are comparable between any two values and that the intervals have the same meaning on the unit. For all i, the difference in a given unit quantity must represent that in the same quantity of valuation values. This condition is the same as the relationship for the input series of Choquet integrals.

There are many such 3-layer data, such as the evaluation values of the Analytic Hierarchy Process (AHP)[8] and survey data on a 5-point scale.

2.2 Creation of set functions

Set function representations are set functions using the method of computing the input values of the Choquet integral [1]. Let $X = \{1, 2, \ldots, N\}$ be the entire set of analysis items. We create a set function representation for some $\boldsymbol{x}_{j,k} = (x_{j,k,1}, \ldots, x_{j,k,N})$.

For simplicity, we omit the superscript and subscript j, k and describe how to create a set function representation of a vector $\boldsymbol{x}$. We represent $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$, the row values in Table 1, as a set function representation. $\sigma(i)$ is the permutation on X; that is, $x_{\sigma(1)} \geq \ldots \geq x_{\sigma(N)}$, $x_{\sigma(N+1)} = 0$ and $X = \{\sigma(1), \ldots, \sigma(N)\}$. The permutation of $\boldsymbol{x}_{j,k}$ is $\sigma_{j,k}$.

Definition 1 (Mass form set function representation[9]). *For a $\boldsymbol{x}$ and $\forall A \subseteq X$, let us define $\eta^* : 2^X \to \boldsymbol{R}^+$.*

$$\eta^*(A) = \begin{cases} x_{\sigma(i)} - x_{\sigma(i+1)} & \text{if } A = \{\sigma(1), \ldots \sigma(i)\},\ i = 1, \ldots, N \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The set function representation assigned from $\boldsymbol{x}$ by the equation (1) is distinguished from other set function representations by adding *, such as η^*.

A mass form set function representation created from a certain $\boldsymbol{x}_{j,k}$ is denoted as $\eta^*_{j,k}$. In the formula (1), sets are denoted by numbers, but in examples, they are denoted by specific element names. For example, the set of table 1 is denoted by $X = \{B1, B2, B3\}$.

2.3 Maximal chain set sequence

Maximal chain set sequence is defined using the maximal chain([6],[2]) concept.

Definition 2 (Maximal chain set sequence). *We define the maximal chain set sequence $(R_1, \ldots, R_N)$ of $x_{\sigma(1)} \geq \cdots \geq x_{\sigma(N)}$ for $\boldsymbol{x}$.*

$$R_i = \{l \mid x_l < x_{\sigma(i+1)}\} \quad , i = 1, \ldots, N \tag{2}$$

Let $R_i^{j,k}$ be the maximal chain set sequence created from $\boldsymbol{x}_{j,k}$, and denote it as $(R_1^{j,k}, \ldots, R_N^{j,k})$. Additionally, $R_1 \subseteq R_2 \subseteq \ldots \subseteq R_N (= X)$.

In this sequence of sets, up to the ith rank $(R_1, \ldots, R_i)(i = 1, \ldots N)$ is called the maximal chain set sequence up to ith place. By interpreting these maximal chain set sequences, the rank information can be interpreted.

As shown in equation (1), for some η^*, all possible A values such that $\eta^*(A) > 0$ are contained in one maximal chain set sequence.

2.4 Aggregation of set function representations

In a 3-layer data, there are multiple experts ($k = 1, \ldots, K$), which we aggregate. Suppose we have a set function representation $\eta_{j,1}, \ldots, \eta_{j,K}$ for each k and for a j, and we aggregate the set function representations for k.

Definition 3 (Aggregated set function representation η_j).

$$\eta_j(A) = \sum_{k=1}^{K} \eta^*_{j,k}(A), \forall A \subseteq X \tag{3}$$

For each k, the rank of $x_{i,k}$ is generally different. Therefore, A for which $\eta^*_{j,k}(A) > 0$ is generally different, and the maximal chain set sequence for each k is also different.

In the aggregated set function representation, by interpreting A such that $\eta_j(A) > 0, \mid A \mid = 1$, we can interpret the rank information of the element that ranked first in more than one expert. In addition, $\eta_j(A), \mid A \mid = 1$ is the sum of the differences of the second place when A is ranked first by each expert, the value of $\eta_j(A)$ gives the degree of the first place of A.

By interpreting the value of the set A for which $\eta_j(A) > 0, \mid A \mid = 2$, we know the set of elements that are in first or second place, and we also know the sum of the differences from the third place and the degree of second place. Similarly, we can read $\mid A \mid = 3, \ldots, N$ for $\eta_j(A) > 0$.

2.5 Numerical example: set function representations

Figures 1 and 3 present the evaluation values $(\boldsymbol{x}_{3,1}, \boldsymbol{x}_{3,2})$ of experts C1 and C2 for criterion A3 in Table 1, and Figures 2 and 4 are the mass form set function representations $(\eta^*_{3,1}, \eta^*_{3,2})$ of their experts C1 and C2. Figure 5 is $\boldsymbol{x}_{3,1} + \boldsymbol{x}_{3,2}$ in Table 1, and Figure 6 is the aggregated set function representation (η_3), the sum of Figures 2 and 4. In the graphs of each set function representation, the bar of A where $\eta(A) = 0$ is omitted.

In Figures 1 and 3, each value is shown as a set of the equation (1), where the regions are divided by the differences in the input values and the maximal chain set sequence is presented. The regions of each set are extracted and displayed as set function representations in Figure 2 and 4. Figure 6 is the aggregated set function representation of figure 2 and 4. In Figure 6, the sets with one element appear as $\{B1\}$ and $\{B2\}$. This is because the maximal chain set sequence differs between C1 and C2.

2.6 Interpretation of a set function representation (η^*)

Sets with 1 element ($\mid A \mid = 1$) The value $\eta^*(A)$ with one element (e.g., $\{B2\}$) is the evaluation value of the part that exists with that element alone. A is the element of the first-ranked value and $\eta^*(A)$ represents the difference between the first- and second-ranked values. If this value is large, it indicates

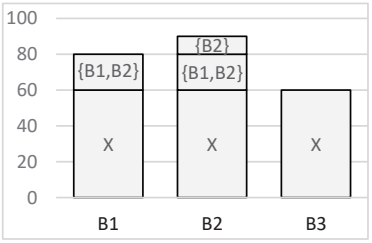


Fig. 1. Evaluation value of C1, $\boldsymbol{x}_{3,1}$

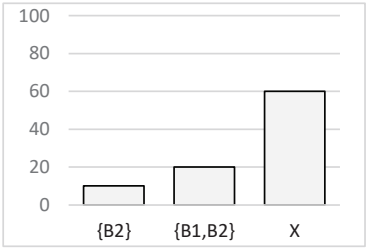


Fig. 2. Set function representation of expert C1, $\eta^*_{3,1}$

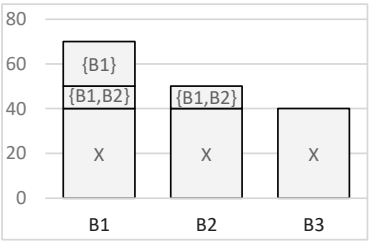


Fig. 3. Evaluation value of C2, $\boldsymbol{x}_{3,2}$

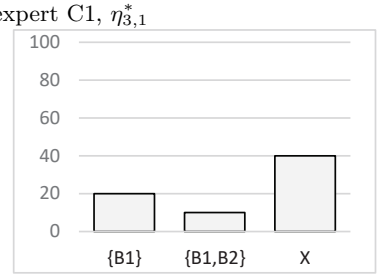


Fig. 4. Set function representation of expert C2, $\eta^*_{3,2}$

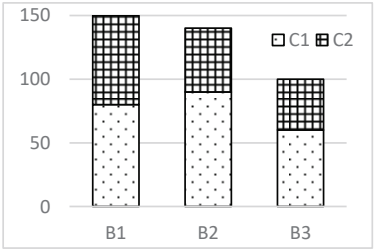


Fig. 5. Sum of two experts, $\boldsymbol{x}_{3,1} + \boldsymbol{x}_{3,2}$

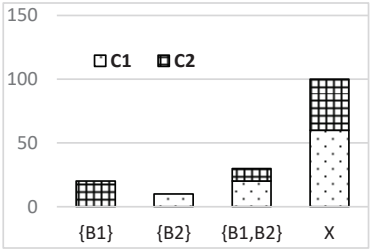


Fig. 6. Aggregated set function representation, η_3

that there is a large difference between the first and second places, and it shows the magnitude of the first place's evaluation value. Even if it is small, $\eta^*(A) > 0$ indicates that A exists in the first position. According to Figure 2, at $\eta^*(\{B1\}) > 0$ and B1 is in first place, whereas Figure 4 shows that $\eta^*(\{B2\}) > 0$ and B2 is in first place.

Sets with 2 elements ($\mid A \mid = 2$) The value $\eta(A)$ with two elements is the part whose elements are both present. This value does not include the portion of the value of the first-rank element alone. A large value indicates that the second place is far ahead of the third.

Sets with more than 3 elements ($\mid A \mid \geq 3$)
When $A = X$, $\eta_j(X) = \min_{i,k} x_{i,j,k}$ is the minimum value. If we consider the

minimum evaluation value as the base point of evaluation, the values of the set function with fewer than $N-1$ elements represent the good portion from the base point. Additionally, $X \setminus A$ in a set A with $N-1$ elements represents the minimum value element. For example, for a set of two elements in C1, $A = \{B1, B2\}$, and thus $X \setminus A = \{B3\}$ indicates that B3 is the minimam value element. A similar reading can be used for $N-2$ sets.

2.7 Interpretation of the aggregated set function representation(η_j)

An example of an aggregated set function representation is presented in Figure 6.

Set with 1 element($\mid A \mid = 1$) In Figure 6, there are two sets with one element: $\{B1\}$ and $\{B2\}$. This is because the first-ranked values are different, as can be seen from Figure 5. Since $\eta_j(\{B3\}) = 0$, it means that no expert ranks first. The value of $\eta(A)$(where $\mid A \mid$=1) is the sum of the degree of difference from the second place.

Set with 2 elements($\mid A \mid = 2$) In Figure 6, the only set with two elements that are greater than 0 is $\{B1, B2\}$. This indicates that B1 and B2 are in first or second place for all experts, indicating that they are in agreement for A3. According to Figure 5, the difference in the sum of the evaluation values of B1, B2, and B3 is small, and the difference in the total evaluation values of B1, B2, and B3 is not significant. However, alternative B3 is ranked third among all experts.

In addition to this example, when B, C are sets of two elements and $B \cap C = \emptyset$, and the values of $\eta(B)$ and $\eta(C)$ are larger than those of the other sets with two elements, we can interpret this as there being two polarized groups: those with a high value of B and those with a high value of C. The same is true for the analysis with three or more elements.

Set with $N-1$ elements ($\mid A \mid = N-1$) Because the number of with two elements is $N-1$, only $\eta(\{B1, B2\})$ is greater than zero, which means that the other elements $X \setminus \{B1, B2\} = \{B3\}$ are not in the second $(N-1)$ rank.

The set A with the small number of elements For some small number L, for example $L = 1$ or 2, there exist a set A where $\eta(A) > 0, \mid A \mid = L$ and $\eta(D) = 0, \forall D, |D| = L, D \neq A$, which indicates there is agreement that A is good alternative for the criterion. In addition, $\eta(\{B1\}) > 0, \eta(\{B1, B2\}) > 0, \eta(\{B1, B2, B3\}) > 0$ and $\eta(D) = 0, \forall D \notin \{\{B1\}, \{B1, B2\}, \{B1, B2, B3\}\}$ then it is consistent with being a good evaluation of B1 $\rightarrow$ B2 $\rightarrow$ B3.

Conversely, when there are many A, $\eta(A) > 0$, $\mid A \mid = L$, the evaluation is divided.

3 Properties of the set function representations

3.1 Weighted mass form set function representation

When comparing $\eta(A)$ of A with two elements and $\eta(A)$ with one element, it is better to consider the difference in the number of elements. The set with m elements appears in m places, as shown in Figures 1 and 3.

Definition 4 (Weighted mass form set function representation $\eta^\sharp$). *We define the weighted mass form set function representations $\eta^\sharp$ that adjusts for the number of elements.*

$$\eta^\sharp(A) =\mid A \mid \eta(A), \ \ \forall A \subseteq X \tag{4}$$

3.2 Canonical form set function representation

η and $\eta^\sharp$ do not include differences in the evaluation values of their proper subsets. From Figures 2 and 4, $\eta(\{B1, B2\})$ is the sum of the parts where both B1 and B2 exist. This is the "AND" part where B1 and B2 are both present. We define a set function representation ρ that contains either part of B1 or B2. For example $\rho(\{B1, B2\}) = \eta(\{B1\}) + \eta(\{B2\}) + \eta(\{B1, B2\})$ is the "OR" part where one of B1 and B2 exists in $\{B1, B2\}$.

Definition 5 (Canonical form set function representation). *We define the canonical form set function representations $\rho(A)$ and $\rho^\sharp(A)$.*

$$\rho(A) = \sum_{B \subseteq A} \eta(B), \forall A \subseteq X \tag{5}$$

$$\rho^\sharp(A) = \sum_{B \subseteq A} \eta^\sharp(B), \forall A \subseteq X \tag{6}$$

$\rho^*_{j,k}(A)$ where $A = \{\sigma_{j,k}(1), \ldots, \sigma_{j,k}(\mid A \mid)\}$ is the difference from the first place value to $\mid A \mid +1$.

$$\rho^*_{j,k}(A) = \sum_{i=1}^{|A|} [(x_{\sigma_{j,k}(i),j,k} - x_{\sigma_{j,k}(i+1),j,k})] = x_{\sigma_{j,k}(1),j,k} - x_{\sigma_{j,k}(|A|+1),j,k} \tag{7}$$

Additionally, ${\rho^*}^\sharp_{j,k}(A)$ where $A = \{\sigma_{j,k}(1), \ldots, \sigma_{j,k}(\mid A \mid)\}$ is the sum of the values of each element of A.

$${\rho^*}^\sharp_{j,k}(A) = \sum_{i=1}^{|A|} [i(x_{\sigma_{j,k}(i),j,k} - x_{\sigma_{j,k}(i+1),j,k})] = \sum_{i \in A} [x_{\sigma_{j,k}(1),j,k} - x_{\sigma_{j,k}(|A|+1),j,k}] \tag{8}$$

$${\rho_j}^\sharp(X) = \sum_{k=1}^{K} \sum_{i \in X} [x_{\sigma_{j,k}(1),j,k} - x_{\sigma_{j,k}(N+1),j,k}] = \sum_{k=1}^{K} \sum_{i \in X} x_{i,j,k} = \sum_{i=1}^{N} \sum_{k=1}^{K} x_{i,j,k} \tag{9}$$

Equations (5) and (6) show that η and are $\eta^\sharp$ Möbius transformation [3] of ρ and $\rho^\sharp$. η can be obtained from ρ as follows:

$$\eta(A) = \sum_{B \subseteq A} (-1)^{|A \setminus B|} \rho(A), \forall A \subseteq X \tag{10}$$

The set function representations of η and ρ are equivalent representations corresponding one-to-one to each other. The set function representations η and $\eta^\sharp$ are equivalent representations. Therefore, η, $\eta^\sharp$, ρ, and $\rho^\sharp$ are equivalent set function representations that correspond one-to-one.

3.3 Comparison of the set function representations (numerical example)

As a numerical example, for the aggregated set function representation of C1 and C2 in Figure 6 of Table 1, Figure 7 shows the mass form (η), weighted mass form ($\eta^\sharp$), canonical form(ρ), and weighted canonical form($\rho^\sharp$).

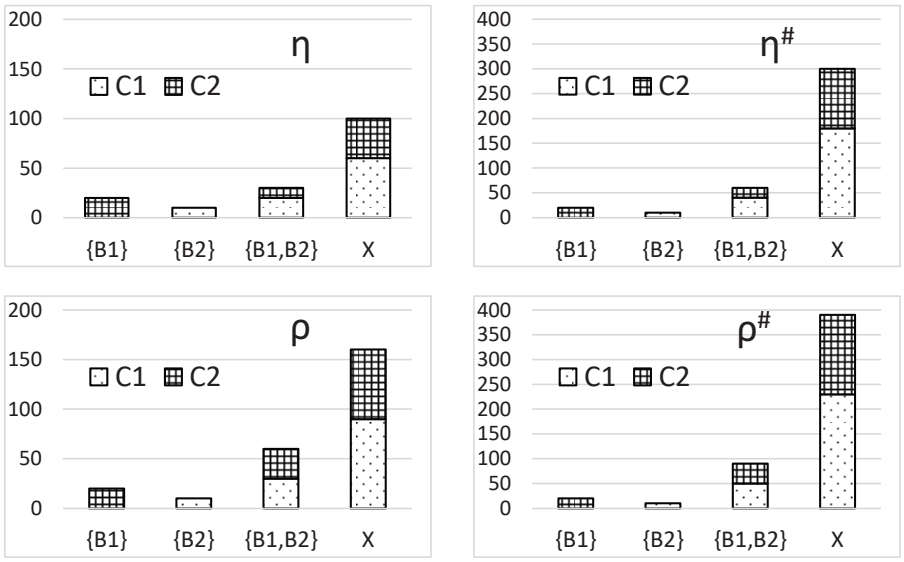


Fig. 7. Set function representations

Because $\eta^\sharp(A)$ is the $\eta(A)$ value multiplied by the number of elements in A, the difference between the 2nd and 3rd places for each expert of C1 and C2 is doubled, and the difference between the 3rd place and 0 is tripled. Therefore, $\eta^\sharp$ is the sum of all experts of each difference and is a comparison between sets with different numbers of elements. In Figure 7, the values of $\{B1\}$ and $\{B2\}$ are smaller than $\{B1, B2\}$ and $\eta^\sharp(\{B1\}) + \eta^\sharp(\{B2\}) < \eta^\sharp(\{B1, B2\})$. The difference between the first and second places is smaller than the difference between the second and third places for the whole experts.

$\rho(A)$ is the sum of the subsets. $\rho(\{B1, B2\})$ is the difference between the 1st and 3rd place values when B1 and B2 are the 1st and 2nd places, respectively. Therefore, the aggregated set function representation $\rho(\{B1, B2\})$ is an indicator of how well B1 or B2 are evaluated as a whole.

$\rho^\sharp(\{B1, B2\})$ also reveals how well B1 or B2 is evaluated. If the set of two elements other than $\rho^\sharp(\{B1, B2\})$ is 0, it indicates that only B1 and B2 have good evaluation values. In addition, $\rho^\sharp(X) = 390$ is equal to the sum of the individual evaluation values of C1 and C2 (Equation (9)).

3.4 Shapley value

By interpreting the properties of the Shapley values of ρ and $\rho^\sharp$, we can observe the properties of η and $\eta^\sharp$. The Shapley value of the i-th element of ρ is defined by the following equation. Using the Möbius transformation, we can obtain this from η.

$$sh_i(\rho) = \sum_{S \subseteq X} Q_N(S)[\rho(S) - \rho(S \setminus \{i\})] \tag{11}$$
$$Q_N(S) = \frac{(N - \mid S \mid)!(\mid S \mid -1)!}{N!}$$

$$sh_i(\rho) = \sum_{A \ni i} \frac{1}{\mid A \mid} \eta(A) \tag{12}$$

$$sh_i(\rho^\sharp) = \sum_{A \ni i} \frac{1}{\mid A \mid} \eta^\sharp(A) \tag{13}$$

3.5 Shapley value of $\rho^\sharp$

Theorem 1 (Shapley values of $\rho^{*\sharp}$).

$$sh_i(\rho^{*\sharp}) = x_i \tag{14}$$

(Proof)

$$sh_{\sigma(i)}(\rho^{*\sharp}) = \sum_{A \ni \sigma(i)} [\frac{1}{\mid A \mid} \eta^{*\sharp}(A)] = \sum_{j=i}^{n} \frac{1}{\mid \{\sigma(1), \ldots, \sigma(j)\} \mid} \eta^\sharp(\{\sigma(1), \ldots, \sigma(j)\})$$
$$= \sum_{j=i}^{n} \frac{1}{j} j \cdot [x_{\sigma(j)} - x_{\sigma(j+1)}] = \sum_{j=i}^{n} [x_{\sigma(i)} - x_{\sigma(j+1)}] = x_{\sigma(i)} \tag{15}$$

Theorem 2 (Shapley values of $\rho^\sharp$). *The Shapley value of $\rho^\sharp$ is equal to the sum of $x_{i,k}$*

$$sh_i(\rho^\sharp) = \sum_{k=1}^{K} x_{i,k} \tag{16}$$

(Proof)

$$sh_i(\rho^\sharp) = \sum_{A \ni i} [\frac{1}{\mid A \mid} \eta^\sharp(A)] = \sum_{A \ni i} [\frac{1}{\mid A \mid} \sum_{k=1}^{K} \eta^{*\sharp}(A)]$$
$$= \sum_{k=1}^{K} [\sum_{A \ni i} \frac{1}{\mid A \mid} \eta^{*\sharp}(A)] = \sum_{k=1}^{K} x_i \tag{17}$$

Therefore, $\sum_{i=1}^{N} sh_i(\rho_j^\sharp, X) = \sum_{i=1}^{N} \sum_{k=1}^{K} x_{k,i}$. This is consistent with the efficiency of the Shapley values. Thus, the Shapley value of $\rho^\sharp$ indicates the allocation of the sum of the evaluation values.

3.6 Shapley value of ρ

The i-th Shapley value of ρ^* is a weighted sum of the rank difference.

$$sh_{\sigma(i)}(\rho^*) = \sum_{A \ni \sigma(i)} [\frac{1}{\mid A \mid} \eta^*(A)] = \sum_{j=i}^{N} [\frac{1}{\mid \{\sigma(1), \ldots, \sigma(j)\} \mid} \eta^*(\{\sigma(1), \ldots, \sigma(j)\})$$
$$= \sum_{j=i}^{N} \frac{1}{j} \cdot [x_{\sigma(j)} - x_{\sigma(j+1)}] \tag{18}$$

This value emphasizes the difference with good rankings.

Theorem 3 (Sum of Shapley values of ρ^*). *The sum of ρ^* is the maximum value of the element $\boldsymbol{x}$.*
(Proof)

$$\sum_{i=1}^{N} sh_{\sigma(i)}(\rho^*) = \sum_{i=1}^{N} \sum_{j=i}^{N} \frac{1}{j} \cdot [x_{\sigma(j)} - x_{\sigma(j+1)}] = \sum_{i=1}^{N} (x_{\sigma(i)} - x_{\sigma(i+1)})$$
$$= x_{\sigma(1)} = \max(x_1, \ldots, x_n) \tag{19}$$

The Shapley values of ρ^* are interpreted as maximum based values because it allocates the maximum value. The Shapley value of the aggregated set function representation ρ is as follows:

$$sh_i(\rho) = \sum_{A \ni i} [\frac{1}{\mid A \mid} \eta(A)] = \sum_{A \ni i} [\frac{1}{\mid A \mid} \sum_{k} \eta_k^*(A)] = \sum_{k=1}^{K} \sum_{A \ni i} [\frac{1}{\mid A \mid} \eta_k^*(A)]$$
$$= \sum_{k=1}^{K} sh_i(\rho_k^*) \tag{20}$$

Because $sh_i(\rho)$ is the sum of the maximum based values for expert k, it is the maximum based aggregation value because the allocations are based on the maximum value of all experts. These values are also the aggregation values that emphasize higher-ranked values.

3.7 Numerical example: Shapley value of the set function representation

The Shapley value of the set function representation of Figure 7 is presented in Table 2. The Shapley value of ρ is the value that emphasizes the higher-ranked evaluation value.

Table 2. Shapley Value

	Evaluation Value			Shapley Value of ρ_3				Shapley Value of $\rho_3^\sharp$			
	B1	B2	B3	B1	B2	B3	Total	B1	B2	B3	Total
C1	80	90	60	30	40	20	90	80	90	60	230
C2	70	50	40	38.33	18.33	13.33	70	70	50	40	160
C1+C2	150	140	100	68.33	58.33	33.33	160	150	140	100	390
%	38%	36%	26%	43%	36%	21%	100%	38%	36%	26%	100%

4 Conclusion

The set function representations of the data are described in terms of its definitions, visualizations, interpretations, and several one-to-one correspondence representations and their characteristics. The set function representation can preserve the rank information, allowing rank-based considerations.

Although this paper deals only with small models, when the number of criteria, alternatives, and experts is large, a set function representation of the characteristics of the set function is needed.

References

1. Choquet, G.: Theory of capacities, Annales de l'Institut Fourier, **5**, 131–295, (1954)
2. Dukhovny, A.: General Entropy of general measures, Fuzziness, and Knowledge-Based Systems **10**(03), 213–225 (2002). https://doi.org/10.1142/S0218488502001442
3. Fujimoto, K. and Murofushi, T.: Some Characterizations of the Systems Represented by Choquet and Multi-Linear Functionals through the use of Möbius Inversion, International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems, **5**, 547-561, (1997)
4. Fujimoto, K.: A Study on Aggregation Methods of Multiattribute Data and its Interpretations, Journal of Japan Society for Fuzzy Theory and Intelligent Informatics, **32** 792–796, (2020). https://doi.org/10.3156/jsoft.32.4_792
5. Grabisch, M., Marichal, J. Roubens, M.: Equivalent Representation of Set Functions, Mathematics of Operations Research **25**(2), 157-178 (2000)
6. Honda, A., Grabisch, M: Entropy of capacities on lattices and set systems, Information Sciences, **76**(23), 3472–3489, (2006)

7. Honda, A and Okazaki, Y.: Inclusion-Exclusion Integral and t-norm Based Data Analysis Model Construction, IPMU 2016, Communications in Computer and Information Science, 610, pp. 65–77,Springer, 2016. https://doi.org/10.1007/978-3-319-40596-4_7
8. Saaty, T.L.: The Analytic Hierarchy Process, McGraw-Hill, (1980)
9. Takahagi, E.: On a Fuzzy Integral as the Product-Sum Calculation Between a Set Function and a Fuzzy Measure, IPMU 2016, Communications in Computer and Information Science, 610, pp.91–100, Springer, 2016. https://doi.org//10.1007/978-3-319-40596-4_9

Influence of Occlusion in Image Classification with Self-Supervised Capsule Networks

Ladyna Wittscher[0000−0003−2391−2347] and Christian Pigorsch

Chair of Economic and Social Statistics, Friedrich-Schiller-University,
Carl-Zeiß-Straße 3, Jena 07743, Germany
{Ladyna.Wittscher, Christian.Pigorsch}@uni-jena.de

Abstract. Occlusion significantly reduces the information content of images, corrupts object shapes and requires big data sets including a wide range of occlusion types for training. Capsule Networks are an architectural alternative to Convolutional Neural Networks for computer vision tasks that can achieve better results with more shallow networks requiring fewer parameters and generating better results for occluded images. At the same time, they suffer from issues with overfitting, slow convergence and limited robustness. We combine a Capsule Network architecture with self-supervised learning which creates synergies, mitigates shortcomings and improves the performance up to 5.6% for MNIST with high degrees of occlusion, up to 8.2% for the combination of occlusion and data scarcity, and up to 18.1% for noise and data scarcity. Even in scenarios where the accuracy is not being improved significantly, reconstruction quality is. Self-supervision furthermore significantly reduces overfitting and does not require a huge increase in computational complexity. Additionally, we analyze the learning behaviour in detail by comparing different occlusion scenarios, pretraining different layers and parameters as well as investigating the influence of the reconstruction and margin loss on the performance.

Keywords: Capsule Networks · Self-supervised Learning · Occlusion · Data Scarcity · Robust Machine Learning.

1 Introduction

Convolutional Neural Networks (CNN) have a limited robustness to occlusion even when being exposed to large amounts of samples during training [27]. Occlusion is a serious problem as features extracted from occluded images can be corrupted which leads to misclassifications, the information content is significantly decreased and object shapes are altered [19], which makes occlusion e.g. one of the main factors that reduces performance in video surveillance [5]. Cen et al. [4] show that classification accuracy for ResNet18 drops from 93.05% to 22.66% with 20% centered occlusion using Caltech101 [8] data set. The actual features which are affected by occlusion can also make a significant difference, e.g. facial expression recognition performance suffers more due to mouth than

eye occlusion [28]. The variability of occlusion types requires training with big data sets that include a wide range of examples [32]. At the same time, occlusion can be utilized as a regularization methods, improve shape estimation and help to robustify models [3, 9]. Performant models deal better with occlusion during training, targeted occlusion can be used for model interpretability and improve weakly supervised localization by forcing networks to not solely rely on the most discriminative parts of an object [3]. Occlusion can be applied to analyze what a network is focusing on as it forces the network to take the entire image into consideration instead of a subset [49]. Also, the "photographer bias" of benchmark data sets being perfectly centered and the network tending to focus on easily recognizable image parts can be reduced due to occlusion [9]. Consequently, occlusion is not only a problem in numerous use cases, but it is also an interesting scenario to evaluate and influence the model's learning behaviour.

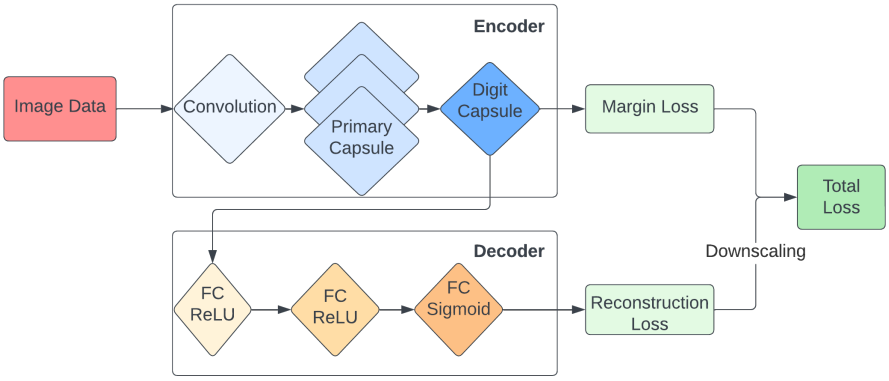


Fig. 1. Schematic functionality of Capsule Networks including encoder and decoder.

Capsule Networks (CapsNets) can detect objects even with a high degree of occlusion [44] due to their ability to gain more discriminative information from image data caused by a better preservation of the spatial relationship among various features of the unobstructed target part [43]. They have outperformed traditional CNN concerning vision tasks regarding rotational invariance, model interpretability and small training sets [48]. The main difference of CapsNets compared to CNN is the eponymous capsule, which is considered as a distinct entity representing several neurons and has a vector output [24]. The feature map is still extracted with convolutional layers in the first place, but then transformed into primary capsules [13]. While spatial hierarchies in CNN are lost due to max pooling, CapsNets do not only include new structural elements like the capsule, they also have an improved inter-layer communication due to dynamic routing, which ensures that each capsule output is forwarded to the next capsule receiving the most similar other inputs [45, 59]. Each capsule encodes an image feature by combining individual neurons that each represent a property of the feature, such

as texture, pose or deformation, which are represented in the output vector [16]. If a feature is detectable, the output vector of the capsule has a high magnitude and its direction is equivariant to the orientation of that particular feature [24]. Dynamic routing ensures that each capsule output is forwarded to the next capsule which gets the most similar other inputs, which adds more invariance, improves inter-layer communication, and makes connection strength a learnable parameter [45, 59]. Instead of an activation function, CapsNets use a squashing function [45]. While the encoder consists of a convolutional, a primary capsule (PrimaryCaps) and a digit capsule (DigitCaps) layer, the decoder is made up of fully connected layers [45], as can be seen in Fig. 1. After classification of the sample in the decoder, a class-conditional reconstruction sub-network generates the reconstruction loss to enable regularization [59]. CapsNets are characterized by fewer parameters than CNN, instead of links between individual neurons they only contain connections between capsules and more shallow CapsNets can have a comparable or better performance than CNN [45]. According to Kapadnis et al. [24], the main developmental steps of CapsNets are: transforming auto-encoders [17], vector capsules with dynamic routing [45], and matrix capsules with expectation-maximization routing [18]. This publication will focus mainly on the work of Sabour et al. [45] as functional framework. While CapsNets deal better with rotated data [59], class imbalance [21], deformation [38], small data sets [54] and occlusion [30] compared to CNN, they suffer from overfitting with too many iterations and layers, which limits the model complexity [51, 59]. They have a higher run time and a larger computational complexity [11]. Also, both dynamic routing and margin loss are counter-productive regarding transformation robustness [13] and CapsNets have difficulties reconstructing complex data sets like SVHN or CIFAR10 [36].

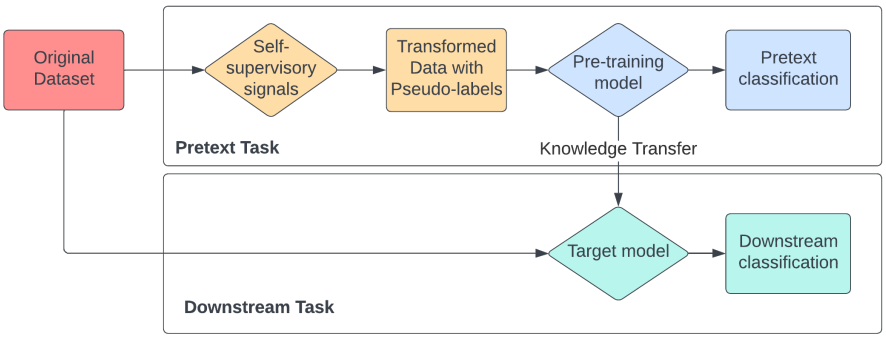


Fig. 2. Overview of self-supervised learning with pretext and downstream task.

Several shortcomings of CapsNets, such as overfitting [26], difficulties with weight initialization [53], and low convergence speed [11], can be mitigated by an alternative training approach: Self-supervised learning, a form of pretraining

that does not require man-made labels because supervisory signals are generated from the input data itself by solving auxiliary tasks [62]. The visual representations and intricate dependencies in the training data are learned during the so called pretext task and can then be used to improve diverse downstream computer vision tasks due to the pretrained feature extractors [20, 37]. Even when self-supervision cannot boost performance compared to training from scratch, it can significantly improve robustness [15]. An overview of the general functionality of self-supervised learning can be seen in Figure 2. Self-supervised learning is beneficial when training data is scarce, no pretrained models are available, overfitting has to be prevented or when manual annotation suffers from high inter- and intra-observer variability [22, 26, 37]. It boosts learning occlusion-invariant representations for all degrees of occlusion [39, 41] without needing man-made annotations as learning contextual information is incorporated in contrast to CapsNet [22, 42].

Having similar strengths, the combination of self-supervised CapsNets should create synergies, especially regarding robustness towards occlusion. At the same time, self-supervision improves model robustness. In our analysis, we focus on the three following key advantages of self-supervised CapsNets: (1) The model performance is better than non-pretrained CapsNets regarding high degrees of occlusion, noise and data scarcity. (2) Self-supervised learning reduces the overfitting tendency of self-supervised CapsNets significantly. We particularly show the influence of the reconstruction and the reconstruction loss on the overfitting problem and how self-supervision improves that. Thereby, self-supervision helps to mitigate a considerable disadvantage of CapsNet. (3) By variations of the occlusion scenarios, the influence of the pretext task on the learning behaviour can be studied in detail. Thereby, we generate a deeper understanding of both learning paradigms and the combination.

2 Related Work

2.1 Robustness of Capsule Networks

Robustness characterizes the ability of machine learning algorithms to deal with erroneous inputs and parameters as the success of models depends on the reliability of their performance [61]. So how robust are CapsNet, especially in comparison to CNN, which they have been intended to surpass? It is difficult to draw general conclusions concerning the robustness of CapsNets as their performance differs significantly for different data sets [23]. Using the original implementation by Sabour et al., CapsNets are moderately robust to small affine transformations [45]. Although being better at preserving spatial relationships, CapsNets still perform significantly better on untransformed inputs compared to transformed ones [12]. Li et al. compared a simple CapsNet and a CNN, in which the capsule layer was replaced with a fully connected layer [30]. They calculate a robustness index for the different scenarios, taking manipulation and the achieved accuracy into consideration [30]. The rotational robustness index is approximately 15% higher compared to CNN, for shifting it is improved by

13%, for scaling 11%, 10% for cropping, 3% for both brightness and blurring, 2% for noise as well as 9% gain for occlusion [30]. CapsNets achieve equivariance instead of translational invariance [31] and generalization over varieties of poses does not require massively replicated feature detectors across viewpoints, consequently, CapsNets have already been successfully used for training with small data sets (see e.g. [1, 2, 24, 43, 55, 58, 60, 64]). Ren et al. [43] tested different CapsNet architectures with different degrees of data scarcity, the average overall accuracy of their proposed CapsNet is 80% higher for 10% of the original MSTAR data set [43]. CapsNets can be better for training complex data sets with few training examples [23], as CNN loose more features due to the pooling operation, CapsNet in general need less training data [24, 64]. CapsNets have a high robustness regarding occlusion (see e.g. [6, 34, 44, 63]). They perform better regarding the average recognition rate of facial expressions given different degrees of occlusion; the improvement is up to 13.38 percentage points compared to CNN [30], due to their ability to preserve spatial information they are superior to CNN [43]. CapsNets are also beneficial regarding noise: The average recognition rate of six different facial expressions given either Gaussian noise or salt-and-pepper-noise results in higher accuracies using a CapsNet implementation with three layers compared to a CNN of the same depth [30]. According to Juralewicz & Markowska-Kaczmar, CapsNets are also more robust to randomly shuffled images than CNN, although capsule-specific elements do not "provide considerable improvement in preserving the spatial relationship between capsules" [23]. Especially with an elevated number of layers, CapsNets show a considerable overfitting tendency as reconstruction does not provide a strong enough regularization, even adding dropout is not sufficient [51, 59]. Gu et al. find dynamic routing actually being harmful to robustness as well as semantic representation and CNN being able to outperform CapsNets in terms of affine input transformations [13]. Consequently, they are more robust in some specific contexts, but their robustness should be improved further.

2.2 Improving model robustness with self-supervision

Self-supervised learning is a branch of unsupervised learning that can learn the underlying representations of unlabeled data by using the input data itself for supervision, e.g. by predicting some parts of the data from another [20, 33]. Self-supervision is a specific form of pre-training which solves an auxiliary task before the actual downstream task using the same data set, which is in general beneficial for small data sets, big domain gaps, and occluded data sets [41, 62]. Even when pre-training does not boost performance compared to training from scratch, it can significantly improve robustness concerning label corruption, adversarial accuracy [15] and increase verifiable robustness [47]. The approach can improve image classification robustness with CNN [57] and extracts more diverse features than supervised learning [50]. The best combination of robust and compact models can be achieved when applying pre-training and fine-tuning using similar objectives [47]. Kortylewsk et al. demonstrated that ResNext can develop an invariance to partial occlusion when being pre-trained with ImageNet [27].

Pre-training can also help to reduce overfitting in small data regimes and works significantly better for classification than for object detection [14].

2.3 Self-supervised Capsule Networks

Self-supervised learning and CapsNets have been previously combined, but self-supervision has mostly been used to approach unsupervised training. Sabour et al. introduce self-supervised training for visual part descriptors using a proxy motion task where the encoder pairs successive video frames to realize part discovery without annotations or segmentation masks [46]. A self-supervised model for primary capsule decomposition through permutation-equivariant attention in 3D point clouds can be trained with pairs of randomly rotated objects to outperform state-of-the-art method on both 3D point cloud reconstruction, canonicalization, and unsupervised classification [52]. Tran et al. apply a self-supervised CapsNet for volumetric medical image segmentation using an UNet-based architecture with a 3D Capsule encoder and 3D CNNs decoder, generating improvement using less data and needing no additional computation complexity at test time [53]. They use contrast transformation as pretext task as medical images often contain patterns of interest [53]. In a previous paper, we analyze self-supervised CapsNet for data scarcity and demonstrate that the combination can improve test accuracy by up to 11.7% for small data sets and by up to 11.5% for small and imbalanced data sets [58]. Mei & Yin propose a cascade residual CapsNet for hyperspectral images clustering with coding rate reduction as self-supervision to learn subspace structures of hyperspectral image cubes including brink loss [35]. Colorisation as a self-supervised learning task with UCapsNet architecture using convolutional operators for spatial details extraction in combination with capsules used for entity extraction has been developed by Pucci et al. [40]. While most approaches focus on the prospect of training without labels, Wiles et al. demonstrate that their co-attention CapsNet architecture having been trained using self-supervision via camera pose can outperform state-of-the-art models given challenging conditions, which is a sign of robustness [56]. To our knowledge, self-supervised CapsNet have not yet been studied with respect to occlusion, data scarcity and noise, nor has the influence of pretraining different layers and parameters.

3 Methods

3.1 Models

We adapt our model from the original vector CapsNet implementation [45] to analyze the effect of self-supervision on the fundamental functionality of CapsNets. The encoder contains a convolutional layer with 1 input channel, 256 output channels and kernel size 9, the primary capsules with 256 input channels, 32 output channels and kernel size 9 as well as a final digit capsule with 1152 routes, 8 input channels and 16 output channels. The subsequent decoder

includes three fully connected layers of output size 784. In contrast to the Sabour et al. implementation, no 2-pixel shift is applied. We use 3 routing iterations, ReLu activation function, Adam optimizer, a learning rate of 0.001 and batch size 10. An un-pretrained CapsNet is used for comparison and is referred to as the "reference model" in this paper. To ensure comparability, we use the same hyperparameters for both models besides the addition of the pretext task and a different weighting of the reconstruction loss (see Section 3.2). While CNN are highly optimized and well researched, CapsNets are a rather new development, so using a rather basic reference CNN puts more focus on comparing the general functionality instead of the level of optimization. The CNN reference model was also adapted from Sabour et al. [45]. As capsules are a more complex entity than neurons, comparisons solely based on the same number of layers are not constructive, instead the CapsNet version has less trainable parameters in general. The CNN model has two more layers than the CapsNet encoder and 4.3 times more trainable parameters than the whole CapsNet [45]. Three convolutional layers of 256, 256, 128 channels with 5x5 kernels and stride 1 are succeeded by two fully connected layers of size 128, 192. The last layer is connected with dropout to a 10 class softmax layer. Cross entropy loss is used instead of margin loss, otherwise all the training conditions remain the same. Although carefully choosing a suitable reference model, the comparison can not necessarily be generalized to other architectures. The number of training epochs has to be adjusted to the data set size but is in a range from 30-50 epochs. We always train both the reference and self-supervised model for the same duration and use a similar training duration for the self-supervised CNN. Results are determined in triplicate, the mean value is reported.

3.2 Reconstruction and margin loss

Reconstruction should encourage the network to learn more general representations of images [59]. But classification accuracy can plateau earlier than reconstruction accuracy, which indicates that reconstruction does not substantially support the classification [36]. In the original Sabour et al. implementation, reconstruction loss is down-scaled by 0.0005 so as not to overpower margin loss and avoid overfitting [29]. Xi et al. also experimented with different scaling factors for the reconstruction loss due to the complexity differences between MNIST and CIFAR10, but increasing reconstruction loss scaling decreases the accuracy for CIFAR10, which can be attributed to the non-ideal reconstruction quality of the CapsNet for this data set [59]. Low reconstruction qualities for complex data sets might be one general reason why CapsNets do not perform well on these, e.g. for CIFAR10 reconstructions are blurry and lack distinct features, which might be caused by the diversity of view points per class and could be solved by applying a deeper decoder [36, 59]. We analyze if self-supervision allows to up-scale the reconstruction loss and profit more from its regularizing capabilities. The effects of different down-scaling factors using MNIST are analyzed in Section 4.2.

3.3 Self-supervision

The selection of the right pretext task is crucial to successful self-supervised learning as the specific combination with the downstream task has a key influence on the model's behaviour [22]. We decided to use rotation as a pretext task to enhance the spatial knowledge of the CapsNet model further and avoid a too big surplus in computational complexity. Rotation does not generate easily detectable low-level artifacts but is only beneficial for pretraining with data sets that are not rotation invariant [20]. As CapsNets do not only contain different types of layers but also learn the connections between layers, consequently it is interesting to evaluate if pretraining the weights of the different layer types and the coupling coefficient changes performance (see ablation experiment in Section 4.1). We used a rather difficult scenario for this ablation analysis with only 1% of the original MNIST data set and an occlusion of 0-50% to find the best model parameters for the combination of occlusion and data scarcity. The number of pretraining epochs and consequently the pretraining accuracy is an important hyper-parameter for self-supervised learning. There is no linear relationship between pretext and downstream accuracy as phenomena like overfitting can also occur between the two different training steps [25], so hyper-parameter optimization is needed for all scenarios. In general, the smaller the data set and the higher the occlusion factor, the more pretext epochs are required. A good rule of thumb is to use 1 epoch for 100% of the data, 5 for 10% of the data, and 15 for 1%.

3.4 Data sets

The benchmark data set MNIST [7] is manipulated to simulate different degrees of randomized and centred occlusion, to ensure that the information loss is both severe and unpredictable. In the so called "Test" scenario, we manipulate both test and training set using the same degree of occlusion. For all other scenarios, we evaluate our results using the non-manipulated MNIST test set. As occlusion is generated randomly, it is important to use a test set big enough to ensure that diverse degrees of difficulty are present in the test set [10]. The test set size is consequently never reduced. Often real-world training sets do not contain enough occluded images plus the occluded samples might not be diverse enough or not relevant for the specific task [4], so tailored synthetic occlusion can be helpful. The scenarios are named using the minimum and the maximum percentage of the area which can be occluded. Several scenarios still include original images while others only contain occluded ones, so the model's shift of attention can be studied.

4 Results

4.1 Pretraining Ablation Analysis

We conduct an ablation analysis to identify the ideal combination of pretrained layers using 1% of the original MNIST data set and 0-50% occlusion. The results

can be seen in Table 1. An improvement of 4.60% compared to the non-pretrained version can be achieved if the weights of all layers are being pretrained, but the coupling coefficient is not. We will apply this pretraining strategy in all following experiments. Including a pretrained coupling coefficient decreases the improvement slightly to 4.46%, the pre-training of the connections between the capsules therefore does not appear to be advantageous. When only one of the three layers is pretrained, Digit Capsules is the best choice with an improvement of 2.24% compared to 1.35% with the Convolutional Layer and 0.16% with Primary Capsules. The combination of Convolutional and Primary Capsules already generates an improvement of 3.40%, the combination of Convolutional and Digit Capsules generates 3.49%, while Primary with Digit Capsules only results in an improvement of 1.86%.

Table 1. Accuracy improvements generated by pretraining different layers compared to the non-pretrained version using 1% of MNIST data set and 0-50% occlusion.

Pre-Trained Layers	Accuracy	Improvement
None	86.59%	-
Convolutional Layer	87.76%	1.35%
PrimaryCaps Layer	86.73%	0.16%
DigitCaps Layer	88.53%	2.24%
Convolutional Layer + PrimaryCaps Layer	89.53%	3.40%
Convolutional Layer + DigitCaps Layer	89.61%	3.49%
PrimaryCaps Layer + DigitCaps Layer	88.20%	1.86%
All	90.57%	4.60%
All + Coupling Coefficient	90.45%	4.46%

4.2 Reconstruction and margin loss

Literature indicates that reconstruction of CapsNets does not significantly improve classification (see Section 3.2), so we tried to up-scale reconstruction loss and evaluate if this has a positive effect. For the pretext task, it is counterproductive. If the weight of reconstruction loss is increased by factor 10, the accuracy is reduced by approximately 70%, which also results in a decrease in downstream performance. This can be attributed to the reconstruction quality not being sufficient, as the pretext training duration is shorter, while reconstruction converges later than the classification does [36]. Additionally, class-conditional reconstruction does only distinguish the different rotation modes but not the numbers, so the reconstructed images do not have a high semantic value. The effect using different down-scaling factors for downstream task and reference model can be seen in Table 2. The accuracy of the self-supervised approach increases to 90.75% if the weight is up-scaled by factor 10. When being augmented further, the accuracy drops again. If the factor is reduced to 0.0001, the accuracy also drops significantly by 1.15 percentage points. Self-supervision

has a positive influence on reconstruction quality and improves reduction loss up to 70.54%, which explains why increasing the influence of the reduction loss can have positive effects on the self-supervised model. This will be analyzed in more detail in Section 4.4. The self-supervised version only shows signs of overfitting for factor 0.05. In contrast, the reference model is affected by overfitting in all scenarios as the learning process in general only benefits from an up-scaling of the reconstruction loss if the ability of the CapsNet to reconstruct the data set is sufficient. In general, self-supervision improves the results of the encoder.

Table 2. Different down-scaling factors for reconstruction loss, the resulting train and test accuracies for the self-supervised and the reference model using 1% of MNIST data set and 0-50% occlusion including the percental improvement due to the pretext task.

	Self-supervised		Reference		Improvement
Down-scaling factor	Training [%]	Test [%]	Training [%]	Test [%]	Test[%]
0.0001	90.50	89.42	96.67	86.13	3.82
0.0005	91.83	90.57	95.33	86.33	4.91
0.001	90.71	90.58	97.28	86.25	4.93
0.005	90.83	90.89	97.17	86.28	5.34
0.01	90.78	90.40	96.61	86.34	4.70
0.05	94.56	89.20	97.50	86.38	3.26

4.3 Occlusion and data scarcity

Higher degrees of occlusion and smaller data set sizes both decrease the absolute accuracies of the CapsNet models, but the decrease is less severe for the self-supervised version (see Table 3). While pretraining only improves the accuracy slightly for 100% and 10% data set size if there are still un-occluded samples, it results in significant gains in accuracy for all sizes if only occluded data is available for training. The absolute area of occlusion seems to be less important than the existence of un-occluded examples in the training set, which becomes obvious when comparing the 0-80% scenario with the 13-30% one. For 0-80% occlusion, the accuracy is improved by 0.23%, 0.52% and 4.25% for 100%, 10% and 1% data set size due to self-supervision, while for 13-30% occlusion, the improvements are 5.55%, 6.05% and 7.79%. Surprisingly, the accuracies of the reference model for all data set sizes in the 13-50% scenario are better than the 13-30% scenario. This could be attributed to the fact that a bigger occluded area forces the model to shift its learning focus to the outer areas of the image which is available in every sample and more reliable for classification, although it contains less information. Consequently, the regularization effect of training with occluded data which helps to robustify models also depends on the specific make-up of the occlusion scenario. In general, the absolute accuracies are significantly lower if the test set is occluded as well, but also in this case pretraining makes a difference: For 0-80% occlusion in both training and test set and 1% data,

pretraining improves the results by 8.2%. With 10% and 100% data set size, the improvements due to self-supervision are only marginal. If we train with un-occluded data and test on occluded data, the accuracies drop significantly and the self-supervised model (47.29% with 1% of the data set and 0-80% occlusion) is only 0.5% better than the reference version (47.05%). It is more challenging to learn features that might not appear in the test data set than, conversely, to train with a data set affected by occlusion and then test on non-occluded data. Self-supervision improves the results only slightly if there is no occlusion, but the absolute accuracies are significantly higher in this case. For 100% and 10%, the improvement is neglectable, for 1% it is 0.62%, for 0.1% there is 4.08% improvement. Self-supervised CNN is inferior to self-supervised CapsNet in all scenarios with 1% and 10% data set size. In most cases with un-occluded samples, the non-pretrained CapsNet version is outperformed by self-supervised CNN. CNN also performs best in most 100% scenarios, the only exeption is the 13-30% scenario. Consequently, self-supervision in general does improve CapsNet performance but the combination cannot outperform self-supervised CNN if the full data set is available for training. Still, in many application scenarios self-supervision could be beneficial as it helps to avoid overfitting and improves the reconstruction quality.

Table 3. Test accuracies given different occlusion scenarios and data set sizes with MNIST using self-supervised CapsNet, non-pretrained CapsNet (Reference) and self-supervised CNN.

Occlusion scenario	Data set size [%]	Self-supervised [%]	Reference[%]	CNN[%]
0-30%	100	98.08	98.07	98.40
	10	96.36	95.60	96.31
	1	90.84	87.32	89.79
0-50%	100	97.17	97.04	98.42
	10	96.40	96.16	95.94
	1	90.57	86.33	86.43
0-80%	100	97.36	97.14	98.14
	10	96.11	95.61	95.03
	1	82.89	79.51	82.74
13-30%	100	88.20	83.56	87.78
	10	88.12	83.09	75.79
	1	72.09	66.88	64.88
13-50%	100	88.54	84.73	90.00
	10	86.90	83.92	82.26
	1	71.29	67.31	65.38
0-80%	100	75.40	75.35	78.06
(Test)	10	73.22	72.98	72.56
	1	62.81	58.05	61.05
None	100	98.87	98.84	99.16
	10	97.95	97.93	97.87
	1	94.02	93.44	93.03

4.4 Noise

High noise levels significantly decrease the accuracy of the self-supervised and the reference CapsNet model. In contrast to occlusion, there is no clear correlation between the level of difficulty and the improvement generated by self-supervision. Nevertheless, the self-supervised CapsNet can deal significantly better with noise than the non-pretrained counterpart. For Gaussian noise with standard deviation 10, the self-supervised version is 18.1% better, for standard deviation 1 there is no significant difference. For 0.1 and 0.01, the improvement is 1.6% and 1.3%. Self-supervision improves considerably reconstruction loss more than margin loss. For high noise levels, the improvement for margin loss is only slightly above 1%, while the improvement of reconstruction loss is minimum 38.5% and maximum 68.2% for 0.01 noise. Consequently, the general tendency that self-supervision leads to greater improvements if the task is more difficult cannot be observed here. In terms of margin loss, the trend is even almost reversed.

Table 4. Improvements in Reconstruction and Margin Loss due to self-supervision using different standard deviations for Gaussian noise with 1% of the MNIST data set. Additionally, the accuracies of the self-supervised and the reference model as well as the accuracy improvement due to self-supervision are shown.

	Accuracy[%]			Loss Reduction[%]	
Noise	Self-supervised	Reference	Improvement	Reconstruction Loss	Margin Loss
10	30.71	26.01	18.1	41.3	1.5
1	93.57	93.27	0.1	54.7	1.1
0.1	94.43	92.98	1.6	38.5	30.5
0.01	94.53	93.31	1.3	68.3	22.3

5 Conclusion

Self-supervised learning can mitigate some of the challenges that CapsNets are still facing as the combination creates promising synergies, but self-supervised CNN are still more performant for big data sets. Pretraining improves classification accuracy especially in difficult scenarios and generally decreases the overfitting tendency of CapsNets. The original implementation does not include sufficient regularization for increased model complexity, which can be improved by pretraining, so reconstruction loss can be given a higher weight for total loss, therefore the influence of the encoder can be increased. Self-supervision allows more information to be extracted from the same set of data and improves the spatial knowledge of the model further, increasing the accuracy especially given high occlusion factors and small data sets. Furthermore, self-supervised learning also renders CapsNets more stable regarding added noise and improves the capability to classify non-corrupted images while having been trained on occluded ones, so the approach boosts different dimensions of robustness.

References

1. Afshar, P., Heidarian, S., Naderkhani, F., Oikonomou, A., Plataniotis, K.N., Mohammadi, A.: Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images. Pattern Recognition Letters **138**, 638–643 (2020)
2. Aldahr, R.S., Alanazi, M., Ilyas, M.: Evolving deep learning models for epilepsy diagnosis in data scarcity context: A survey. In: 2022 45th International Conference on Telecommunications and Signal Processing (TSP). pp. 66–73 (2022). https://doi.org/10.1109/TSP55681.2022.9851282
3. Burgos-Artizzu, X.P., Perona, P., Dollár, P.: Robust face landmark estimation under occlusion. In: Proceedings of the IEEE international conference on computer vision. pp. 1513–1520 (2013)
4. Cen, F., Zhao, X., Li, W., Wang, G.: Deep feature augmentation for occluded image classification. Pattern Recognition **111**, 107737 (2021)
5. Chandel, H., Vatta, S.: Occlusion detection and handling: a review. International Journal of Computer Applications **120**(10) (2015)
6. De Sousa Ribeiro, F., Leontidis, G., Kollias, S.: Introducing routing uncertainty in capsule networks. Advances in Neural Information Processing Systems **33**, 6490–6502 (2020)
7. Deng, L.: The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE signal processing magazine **29**(6), 141–142 (2012)
8. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: 2004 conference on computer vision and pattern recognition workshop. pp. 178–178. IEEE (2004)
9. Fong, R., Vedaldi, A.: Occlusions for effective data augmentation in image classification. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). pp. 4158–4166. IEEE (2019)
10. Garcia-Garcia, A., Garcia-Rodriguez, J., Orts-Escolano, S., Oprea, S., Gomez-Donoso, F., Cazorla, M.: A study of the effect of noise and occlusion on the accuracy of convolutional neural networks applied to 3d object recognition. Computer Vision and Image Understanding **164**, 124–134 (2017)
11. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728 (2018)
12. Gu, J., Tresp, V.: Improving the robustness of capsule networks to image affine transformations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7285–7293 (2020)
13. Gu, J., Tresp, V., Hu, H.: Capsule network is not more robust than convolutional network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14309–14317 (2021)
14. He, K., Girshick, R., Dollár, P.: Rethinking imagenet pre-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4918–4927 (2019)
15. Hendrycks, D., Mazeika, M., Kadavath, S., Song, D.: Using self-supervised learning can improve model robustness and uncertainty. Advances in Neural Information Processing Systems (NeurIPS) (2019)
16. Hinton, G.: How to represent part-whole hierarchies in a neural network. arXiv preprint arXiv:2102.12627 (2021)
17. Hinton, G.E., Krizhevsky, A., Wang, S.D.: Transforming auto-encoders. In: International conference on artificial neural networks. pp. 44–51. Springer (2011)

18. Hinton, G.E., Sabour, S., Frosst, N.: Matrix capsules with em routing. In: International conference on learning representations (2018)
19. Huang, J.B., Yang, M.H.: Estimating human pose from occluded images. In: Asian Conference on Computer Vision. pp. 48–60. Springer (2009)
20. Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F.: A survey on contrastive self-supervised learning. Technologies **9**(1), 2 (2020)
21. Jiménez-Sánchez, A., Albarqouni, S., Mateus, D.: Capsule networks against medical imaging data challenges. In: Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis, pp. 150–160. Springer (2018)
22. Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: A survey. IEEE transactions on pattern analysis and machine intelligence (2020)
23. Juralewicz, E., Markowska-Kaczmar, U.: Capsule network versus convolutional neural network in image classification. In: International Conference on Computational Science. pp. 17–30. Springer (2021)
24. Kapadnis, S., Tiwari, N., Chawla, M.: Developments in capsule network architecture: A review. Intelligent Data Engineering and Analytics pp. 81–90 (2022)
25. Keshav, V., Delattre, F.: Self-supervised visual feature learning with curriculum. arXiv preprint arXiv:2001.05634 (2020)
26. Kolesnikov, A., Zhai, X., Beyer, L.: Revisiting self-supervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1920–1929 (2019)
27. Kortylewski, A., Liu, Q., Wang, A., Sun, Y., Yuille, A.: Compositional convolutional neural networks: A robust and interpretable model for object recognition under occlusion. International Journal of Computer Vision **129**(3), 736–760 (2021)
28. Kotsia, I., Buciu, I., Pitas, I.: An analysis of facial expression recognition under partial facial image occlusion. Image and Vision Computing **26**(7), 1052–1067 (2008)
29. Kumar, A.D.: Novel deep learning model for traffic sign detection using capsule networks. arXiv preprint arXiv:1805.04424 (2018)
30. Li, D., Zhao, X., Yuan, G., Liu, Y., Liu, G.: Robustness comparison between the capsule network and the convolutional network for facial expression recognition. Applied Intelligence **51**(4), 2269–2278 (2021)
31. Li, J., Zhao, Q., Li, N., Ma, L., Xia, X., Zhang, X., Ding, N., Li, N.: A survey on capsule networks: Evolution, application, and future development. In: 2021 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS). pp. 177–185. IEEE (2021)
32. Li, Y., Zeng, J., Shan, S., Chen, X.: Occlusion aware facial expression recognition using cnn with attention mechanism. IEEE Transactions on Image Processing **28**(5), 2439–2450 (2018)
33. Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., Tang, J.: Self-supervised learning: Generative or contrastive. IEEE Transactions on Knowledge and Data Engineering (2021)
34. Ma, J., Li, J., Du, B., Wu, J., Wan, J., Xiao, Y.: Robust face alignment by dual-attentional spatial-aware capsule networks. Pattern Recognition **122**, 108297 (2022)
35. Mei, Z., Yin, Z.: Self-supervised learning of discriminative spatial-spectral features for hyperspectral images clustering. IEEE Geoscience and Remote Sensing Letters (2022)

36. Nair, P., Doshi, R., Keselj, S.: Pushing the limits of capsule networks. arXiv preprint arXiv:2103.08074 (2021)
37. Ohri, K., Kumar, M.: Review on self-supervised image recognition using deep neural networks. Knowledge-Based Systems **224**, 107090 (2021)
38. Patrick, M.K., Adekoya, A.F., Mighty, A.A., Edward, B.Y.: Capsule networks–a survey. Journal of King Saud University-Computer and Information Sciences (2019)
39. Pautrat, R., Lin, J.T., Larsson, V., Oswald, M.R., Pollefeys, M.: Sold2: Self-supervised occlusion-aware line description and detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11368–11378 (2021)
40. Pucci, R., Micheloni, C., Foresti, G.L., Martinel, N.: Is it a plausible colour? ucapsnet for image colourisation. arXiv preprint arXiv:2012.02478 (2020)
41. Purushwalkam, S., Gupta, A.: Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. Advances in Neural Information Processing Systems **33**, 3407–3418 (2020)
42. Ramasinghe, S., Athuraliya, C., Khan, S.H.: A context-aware capsule network for multi-label classification. In: Computer Vision–ECCV 2018 Workshops: Munich, Germany, September 8-14, 2018, Proceedings, Part III 15. pp. 546–554. Springer (2019)
43. Ren, H., Yu, X., Zou, L., Zhou, Y., Wang, X., Bruzzone, L.: Extended convolutional capsule network with application on sar automatic target recognition. Signal Processing **183**, 108021 (2021)
44. Rodríguez-Sánchez, A., Dick, T.: Capsule networks for attention under occlusion. In: International Conference on Artificial Neural Networks. pp. 523–534. Springer (2019)
45. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. arXiv preprint arXiv:1710.09829 (2017)
46. Sabour, S., Tagliasacchi, A., Yazdani, S., Hinton, G., Fleet, D.J.: Unsupervised part representation by flow capsules. In: International Conference on Machine Learning. pp. 9213–9223. PMLR (2021)
47. Sehwag, V., Wang, S., Mittal, P., Jana, S.: Towards compact and robust deep neural networks. arXiv preprint arXiv:1906.06110 (2019)
48. Shi, R., Niu, L.: A brief survey on capsule network. In: 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT). pp. 682–686. IEEE (2020)
49. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of big data **6**(1), 1–48 (2019)
50. Srinivasan, V., Strodthoff, N., Ma, J., Binder, A., Müller, K.R., Samek, W.: On the robustness of pretraining and self-supervision for a deep learning-based analysis of diabetic retinopathy. arXiv preprint arXiv:2106.13497 (2021)
51. Sun, K., Yuan, L., Xu, H., Wen, X.: Deep tensor capsule network. IEEE Access **8**, 96920–96933 (2020)
52. Sun, W., Tagliasacchi, A., Deng, B., Sabour, S., Yazdani, S., Hinton, G.E., Yi, K.M.: Canonical capsules: Self-supervised capsules in canonical pose. Advances in Neural Information Processing Systems **34** (2021)
53. Tran, M., Ly, L., Hua, B.S., Le, N.: Ss-3dcapsnet: Self-supervised 3d capsule networks for medical segmentation on less labeled data. arXiv preprint arXiv:2201.05905 (2022)

54. Wang, Z., Zheng, L., Du, W., Cai, W., Zhou, J., Wang, J., Han, X., He, G.: A novel method for intelligent fault diagnosis of bearing based on capsule neural network. Complexity **2019** (2019)
55. Wang, Z.M., Tian, J.Y., Qin, J., Fang, H., Chen, L.M.: A few-shot learning-based siamese capsule network for intrusion detection with imbalanced training data. Computational intelligence and neuroscience **2021** (2021)
56. Wiles, O., Ehrhardt, S., Zisserman, A.: Co-attention for conditioned image matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15920–15929 (2021)
57. Wittscher, L., Diers, J., Pigorsch, C.: Improving image classification robustness using self-supervision. Stat **11**(1), e455 (2022)
58. Wittscher, L., Pigorsch, C.: Exploring self-supervised capsule networks for improved classification with data scarcity. In: International Conference on Image Processing and Capsule Networks. pp. 36–50. Springer (2022)
59. Xi, E., Bing, S., Jin, Y.: Capsule network performance on complex data. arXiv preprint arXiv:1712.03480 (2017)
60. Xu, F., Gao, J., Pan, X.: Cow face recognition for a small sample based on siamese db capsule network. IEEE Access **10**, 63189–63198 (2022). https://doi.org/10.1109/ACCESS.2022.3182806
61. Xu, H., Mannor, S.: Robustness and generalization. Machine learning **86**(3), 391–423 (2012)
62. Yang, X., He, X., Liang, Y., Yang, Y., Zhang, S., Xie, P.: Transfer learning or self-supervised learning? A tale of two pretraining paradigms. arXiv preprint arXiv:2007.04234 (2020)
63. Yu, Y., Gu, T., Guan, H., Li, D., Jin, S.: Vehicle detection from high-resolution remote sensing imagery using convolutional capsule networks. IEEE Geoscience and Remote Sensing Letters **16**(12), 1894–1898 (2019)
64. Zhang, X., Luo, P., Hu, X., Wang, J., Zhou, J.: Research on classification performance of small-scale dataset based on capsule network. In: Proceedings of the 2018 4th International Conference on Robotics and Artificial Intelligence. pp. 24–28 (2018)

Characterization of Brain Networks through the lens of Persistent Homology

Toni Lozano-Bagén[1], Eloy Martinez-Heras[4], Elisabeth Solana[4], Sandra Garrido-Romero[1], Sara Llufriu[4], Ferran Prados[1,2,3], and Jordi Casas-Roma[1]

[1] e-Health Center, Universitat Oberta de Catalunya, Barcelona, Spain
{alozanobag,sgarridoromero,fpradosc,jcasasr}@uoc.edu
[2] Centre for Medical Image Computing, Department of Medical Physics and Biomedical Engineering, University College London, London, United Kingdom
[3] Queen Square MS Centre, Department of Neuroinflammation, UCL Institute of Neurology, Faculty of Brain Sciences, University College London, London, United Kingdom
[4] Center of Neuroimmunology, Laboratory of Advanced Imaging in Neuroimmunological Diseases (ImaginEM), Hospital Clínic de Barcelona, Institut d'Investigacions Biomèdiques August Pi i Sunyer (IDIBAPS), Universitat de Barcelona, Barcelona, Spain

Abstract. Brain networks (or graphs) derived from magnetic resonance imaging (MRI) have demonstrated to be an optimal method to represent brain data organization. In this context, graph measures has been widely used as a methodology to represent and study brain data. These measures are typically derived from graph theory and can provide insights into the organization, efficiency, and communication patterns of the brain network. In this work, we propose to use persistent homology and Betti curves to extract features that are relevant to identify people with neurodegenerative diseases, such as multiple sclerosis, from structural brain connectivity, morphological gray matter and functional brain networks. We compare the features extracted from each single-layer and a multi-layer architecture, and prove that using a multi-layer architecture better preserves the brain alterations that drive cognitive processes and brain damage. Finally, we test our method in a cohort of people with MS, proving that features extracted using our proposed conceptual scheme are relevant to identify and classify both healthy volunteers and people with multiple sclerosis (MS).

Keywords: MRI · Brain networks · Graph theory · Persistent Homology · Multiple Sclerosis · Machine Learning

1 Introduction

Recent advances in magnetic resonance imaging (MRI) have facilitated the study of brain connectivity structures and functions, providing a comprehensive understanding of brain connectivity organization and behaviour [3, 11]. Over the last

decade, different complex preprocessing pipelines have been developed for obtaining patterns of structural brain connectivity [12], morphological gray matter [19] or functional brain networks [8].

Brain networks (or graphs) have demonstrated to be an optimal method to represent brain data organization. Specifically, graph theory has been used as straightforward, clear, robust and useful methodology to represent and study brain networks. Several works have been tackling this problem from the point of view of graph theory [17]. Graph theoretical analysis allows us to model complex network systems with comprehensive indices related to the integration, segregation and propagation of information inside the brain system [18]. However, different types of data may require different mathematical tools and techniques for analysis.

The main goal of this work is to propose a novelty methodology to extract relevant features (or embeddings) from brain networks by using tools from topological data analysis. Specifically, we use the persistent homology (PH) and Betti curves to extract important topological features from the structural brain connectivity, morphological gray matter and functional brain networks independently, and by the combination for multimodal brain networks. Additionally, the proposed methodology can be used to study the impact of each brain network data type in the context of neurodegenerative diseases, such as multiple sclerosis (MS) [6].

MS is a chronic, inflammatory, demyelinating, and neurodegenerative disease of the central nervous system characterized by widespread damage leading to disruption of large- and short-scale structural and functional connectivity, which leads to clinical alterations [9, 15]. Local and global implications of damage on networked systems, such as our brain, have been studied from many angles using complex networks. Thus, network theory approaches have been widely applied in the field of neuroscience to study both structural and functional connectivity and explore its relationship with cognitive function [12].

1.1 Contributions

This study presents the methodology that contributes to the state of the art from two points. Firstly, we explore and analyse the brain connectivity networks, including structural brain connectivity, morphological gray matter and functional data, through the lens of persistent homology (PH) and Betti curves. It is a novel approach to extract topological features from brain networks that are relevant in the context of neurodegenerative diseases. Secondly, we empirically compare all single-layers brain networks and the multi-layer architecture [5], and objectively demonstrate that using a multi-layer architecture better preserves the brain mechanisms that drive cognitive dysfunction and brain damage. Finally, our proposed scheme is used to study a cohort of people with MS as a proof of concept, and we claim that features extracted from persistent homology and Betti curves are relevant to identify and classify both healthy volunteers and people with MS.

1.2 Organization

The remainder of this paper is organized as follows: Section 2 explores the literature for related works. We briefly describe the extraction of data procedure as the two approaches designed in Section 3. Section 4 presents the proposed methodology and its related concepts, and we highlight the results in Section 5. Finally, the paper is concluded in Section 6, where we summarize our contributions, insights and future improvements.

2 State of the Art

Network theory approaches have been widely applied in the field of neuroscience to study both structural and functional connectivity and explore its relationship with cognitive function [12, 13, 16]. In this context, single network analysis is limited to only one feature, and hence, does not fully describe the complexity of brain mechanisms after damage. In order to overcome this limitation, a multi-layer architecture was proposed in [5]. However, to the best of our knowledge, there is no objective study about the relevance or importance of each single-layer brain network, i.e. structural, morphological and functional connectivity networks, and the multi-layer architecture to identify people with neurodegenerative diseases. Specifically, studying brain networks in the context of both health and disease plays a critical role in discovering the multimodal brain patterns that drive cognitive processes and brain damage [2].

Previous works investigated connectivity changes in people with MS related to their cognitive status, and they proposed an automatic classification method to classify subjects as patients and healthy volunteers (HV) using graph theory basic metrics, such as local efficiency and node strength, computed on weighted structural connectivity matrices [17].

Algebraic topology is a branch of mathematics that studies topological objects by means of assigning them algebraic structures. Topological Data Analysis (TDA) is a framework of data science that uses tools from algebraic topology to analyse datasets. One of the most used methods in TDA is persistent homology, which is an adaptation of homology when the objects to study are graphs or sets of points. For an overview of persistent homology, and to know more about its use in machine learning, see, for example, [14].

Topological Data Analysis, and in particular persistent homology, has been previously applied to study brain connectivity in relation to some diseases [4], but the specific techniques, dataset, and goal of our paper are quite different to the ones present in the literature.

3 Dataset

This study used data on patients with relapsing-remitting, primary or secondary progressive MS recruited at the MS Unit at Hospital Clínic de Barcelona. The Ethics Committee of the Hospital Clínic de Barcelona approved the study, and all participants signed an informed consent.

3.1 Participants

We analysed a cohort of $n = 147$ people with relapsing-remitting, primary or secondary progressive MS (104 women), mean age of 47.33 ± 10.14 years, mean disease duration of 15.96 ± 9.04 years, and median EDSS (Expanded Disability Status Scale) of 2.0 (range 0–7.5), and a group of $n = 18$ healthy volunteers (HVs), mean age of 36.62 ± 9.33 years. The clinical and demographic from the final cohort are summarized in Table 1.

Table 1: Clinical and demographic data. Continuous variables are given as the mean ± standard deviation. EDSS = Expanded Disability Status Scale; MS = multiple sclerosis. p values obtained from comparing the groups.

	Healthy volunteers (n = 18)	People with MS (n = 147)	p value
Age, years	36.62 ± 9.33	47.33 ± 10.14	<0.001
Female, n (%)	15 (83%)	104 (71%)	<0.001
Disease duration, years	–	15.96 ± 9.04	–
Median EDSS score (range)	–	2.0 (0-7.5)	–

3.2 Brain networks and adjacency matrices

The DTI, RS-fMRI and GM values are derived from 3 MRI modalities acquired within the same scan session in a 3T Siemens scan. Hence, the subject goes inside the scanner, and we do the experiments for acquiring diffusion weighted images, resting-state functional images and a structural scan using 3D T1 images. After a complex preprocessing of the images, we can derive DTI measures from the DWI MRI, functional measures from the RS-fMRI and get patterns of cortical thickness from the T1 MRI.

Therefore, each subject has three single-layer networks representing DTI structural connectivity, GM morphology and RS-fMRI functional activity. The nodes of the three brain networks constructed are the 76 brain regions defined in the common anatomical parcellation scheme. Therefore, the same parcellation is used within each network and nodes of all networks are equivalent and represent the same anatomical brain region. The scheme to create the three single-layer networks is depicted in Figure 1.

Structural brain connectivity network. The first step in constructing a structural connectivity matrix was to build a DWI preprocessing pipeline to fit the diffusion tensor imaging (DTI) model, an approach previously described and well established by [20]. The parcellation scheme (76 nodes) from the anatomical image was aligned to the FA map to determine which streamline connections needed to be selected between pairs of nodes to create the structural connectome. We defined the mean value of the FA metric along each connection to

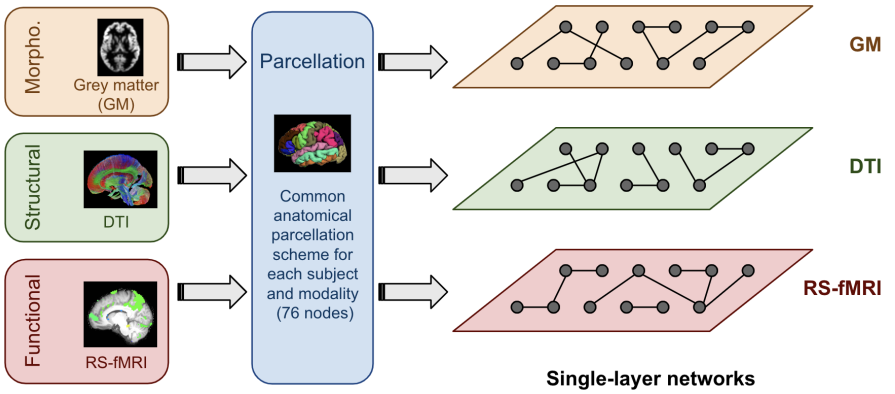


Fig. 1: Scheme to create structural brain connectivity network (DTI), morphological gray matter brain network (GM) and functional brain network (RS-fMRI)

generate the FA-weighted adjacency matrix of the network, denoted by $A^{(DTI)}$. The mean FA computed along the fiber pathway that connects each pair of brain regions enables the inclusion of the severity of the white matter damage at the macro- and microstructural levels [12]. Finally, the FA measures for the structural network were corrected for age and gender effect using a regression model [17]. The values of DTI connectivity matrices are in the range $[0, 1]$, where values close to 0 indicate null connectivity and values close to 1 point out the maximum connectivity.

Morphological gray matter brain network. The GM morphological network is based on the similarity of GM morphological patterns according to the defined anatomical parcellation scheme [19]. We construct the final GM morphological network and its adjacency matrix, denoted by $A^{(GM)}$, considering the defined parcellation scheme (76×76). The morphological networks obtained were corrected for the effects of age and gender using a regression model. The values of GM morphological matrices are in the range $[0, 1]$.

Functional brain network. Brain signal correlation/synchronization through resting-state functional connectivity (RS-fMRI) matrix was obtained following [7], and the defined parcellation was used to extract the average time series for each of the 76 brain regions, resulting in a functional connectivity network with adjacency matrix $A^{(RSfMRI)}$. Note that the values of RS-fMRI matrices are in the range $[-1, 1]$, indicating negative or positive correlation between nodes. However, we apply the absolute value in order to preserve only the strength of the relationship. As with the other networks, age and gender effects were also corrected for functional connectivity networks using a regression model. The final values of RS-fMRI matrices are in the range $[0, 1]$.

3.3 Multilayer brain network

We propose to include in our comparison the multi-layer scheme developed in [5], as a complex network composed of different layers, each representing a single type of relationship between nodes within one layer. Nodes represent the same exact object in each of the different layers, and encode different types of relationships throughout their edges. In this network, the authors differentiate between intralayer links, which encode the single type of relationship the layer represents, and interlayer links, which encode how the different node perspectives (types of relationships) are related within the system.

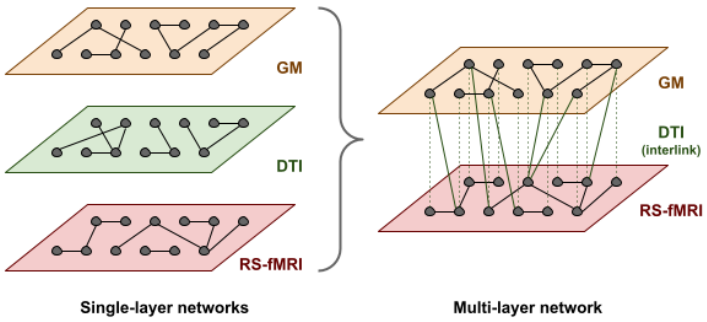


Fig. 2: Scheme to create the multi-layer network, as defined in [5]

In this multi-layer network, each subject has three single-layer networks representing GM morphology, DTI structural connectivity, and rs-fMRI functional activity, which are combined to create a multi-layer network composed of two layers, as can be seen in Figure 2.

3.4 Data and Code Availability

The proposed method were made publicly available by the authors[5], while data used in our work is publicly available[6] and described in [5].

4 Methodology

For each subject in the dataset, we have four possibilities to construct a graph:

- Single-layer graph with DTI structural connectivity network.
- Single-layer graph with GM morphological network.
- Single-layer graph with RS-fMRI functional network.

[5] Code repository: https://github.com/ADaS-Lab/PH-MRI/
[6] Data repository: https://github.com/ADaS-Lab/Multilayer-MRI

– Multi-layer graph with DTI, GM, and RS-fMRI network, following the architecture previously described.

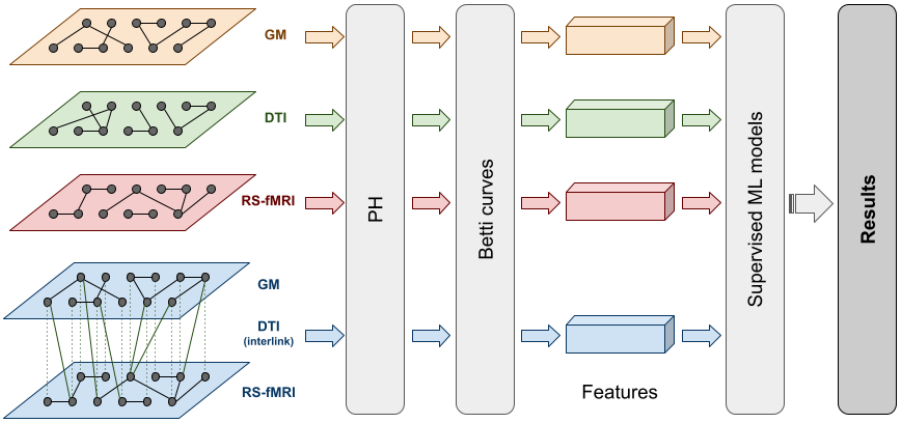


Fig. 3: Pipeline designed for our experiments

Our main goal is to measure the predictive information contained in the different graph constructions by training supervised machine learning (ML) models using each construction separately and computing performance metrics in each case. In particular, we are interested in the topological information present in each graph, so we will use persistent homology (PH) to create features (or embeddings) that we use to feed to several machine learning models. The complete pipeline for our methodology is detailed in Figure 3.

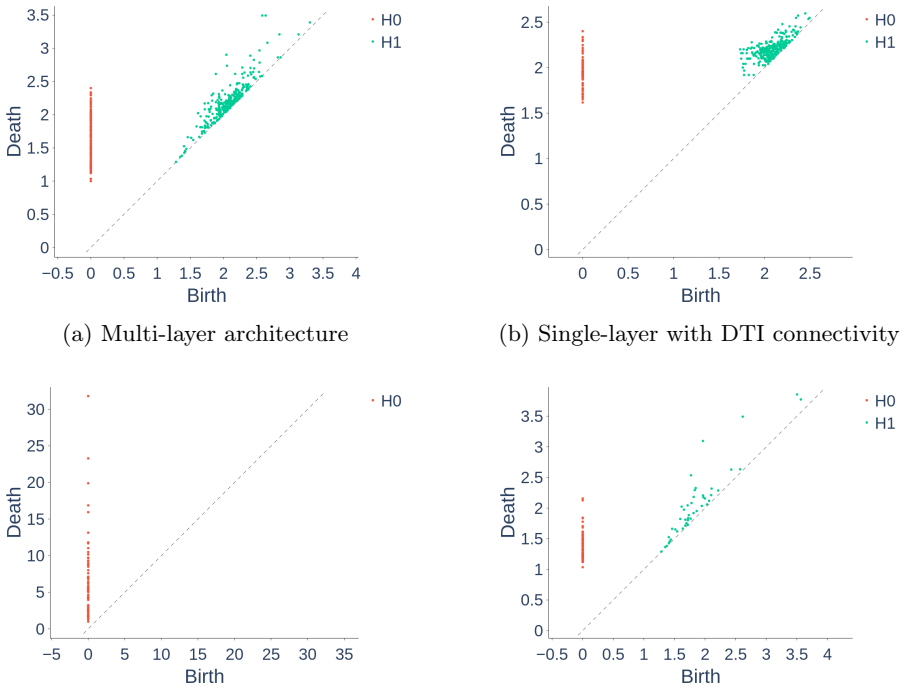
4.1 Persistent Homology

Usually, persistent homology is applied to a set of points by constructing a completely connected graph in which each node corresponds to a point in the dataset and a pair of nodes is connected by an edge with weight proportional to the distance between the two corresponding points. In our case, we have a dataset in which each data point is already a graph, so we apply persistent homology to each graph separately. The goal is to create persistent homology features that can be associated to each graph separately, to then train models to try to predict if a subject is a patient with MS or a healthy volunteer (HV).

Figure 4 shows the persistent diagrams resulting from computing persistent homology in dimensions 0 and 1 for a single subject, considering the four different possibilities to associate a graph to a subject discussed previously. Each dot in the diagram represents a topological feature, either in dimension 0 (H0, red dots) or in dimension 1 (H1, green dots). The position of the dot in the diagram indicates the value of the filtration parameter in which the topological feature

appears (Birth) compared to the value of the filtration parameter when the topological feature disappears (Death). The dots far away from the diagonal represent the more persistent topological features in the graph.

We can see in the persistent diagrams that the topological features for each type of graph have different structures, and these differences will be reflected later in the results of the machine learning models. Also, it is interesting to note that the graph constructed with GM connectivity data seems to have only 0 dimensional homology. This phenomenon is present in all subjects, and in a future work we are planning to investigate it further.



(a) Multi-layer architecture

(b) Single-layer with DTI connectivity

(c) Single-layer with GM connectivity

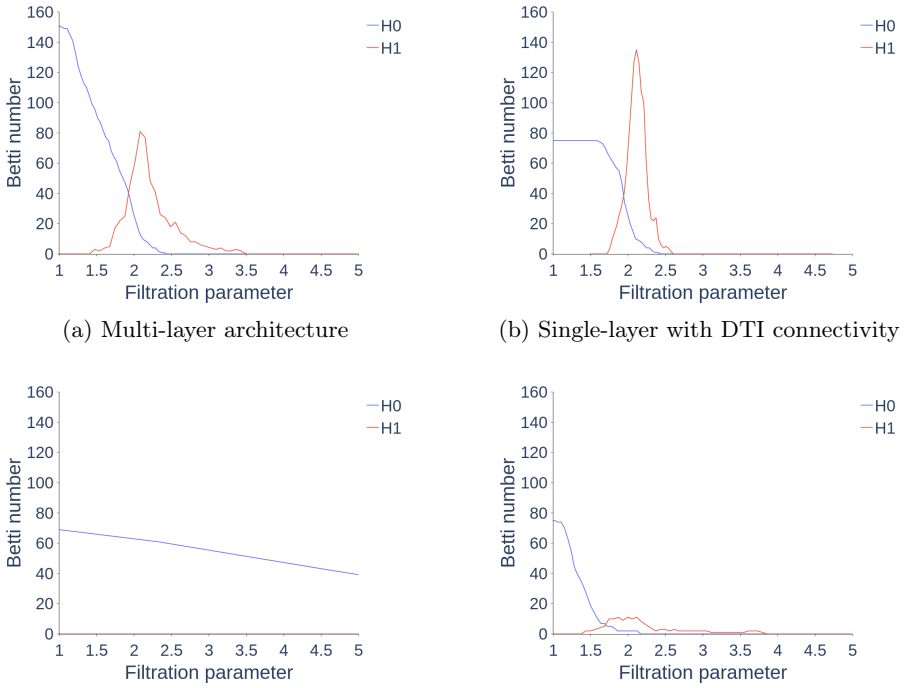
(d) Single-layer with RS connectivity

Fig. 4: Persistent Homology diagrams for a single subject.

The persistent homology diagrams have a variable number of features for each subject, depending on the actual topological structure for each graph. To be able to use this kind of information in a traditional machine learning model, we need to convert the variable number of topological features into a fixed number of numerical features.

There are different methods to convert a persistent diagram with a variable number of features to a vector with a fixed number of components. In this case, we will use Betti curves, presented in [21].

In Figure 5, we can see the Betti curves associated to the persistent diagrams shown in Figure 4. In these diagrams, each curve represents a dimension, and each point in the curve represents how many topological features are alive for each possible value of the filtration parameter. By construction, we can also see in these diagrams that the structure of topological features is different in each type of graph, so the machine learning models should be able to react different to the different graphs using the features extracted from the Betti curves.



(a) Multi-layer architecture

(b) Single-layer with DTI connectivity

(c) Single-layer with GM connectivity

(d) Single-layer with RS connectivity

Fig. 5: Betti curves for the same subject as Figure 4.

4.2 Machine Learning Models

To test if the features extracted from the graphs using persistent homology are useful, we will train different supervised machine learning models on each type of constructions to try to separate between patients with MS and healthy volunteers (HV). We will repeat the same experiment 10 times with different random initializations for the train-test split, and we will compute the mean and standard deviation of the AUC ROC metric for each classifier. The models we will consider are the following ones:

– A fully connected neural network (NN) with three hidden layers.
– A simple logistic regression (LR).
– A random forest (RF) classifier.

To try to compensate for the fact that the multilayer architecture is using all types of connectivity at the same time, we will also train the same models but using the concatenation of the features for the three single layer graphs. This allows us to see that the multilayer architecture is essential, and that it is not enough to consider information for all connectivity types in any kind of combination.

5 Results

Tables 2 and 3 present a summary of the results for the experiments outlined in the previous section, where Table 2 handles the case with homology dimensions 0, 1, and 2 and Table 3 handles the case with homology dimensions 0 and 1.

Table 2: AUC ROC (mean ± std) for different machine learning models applied to the different processing of the brain connectivity data. Using homology dimensions 0, 1, and 2

	Neural network	Logistic Regression	Random Forest
Multi-layer	$\mathbf{0.75 \pm 0.02}$	0.57 ± 0.02	$\mathbf{0.70 \pm 0.02}$
Single-Layer FA	0.57 ± 0.02	$\mathbf{0.58 \pm 0.02}$	0.58 ± 0.02
Single-Layer GM	0.56 ± 0.03	0.46 ± 0.03	0.46 ± 0.04
Single-Layer RS	0.59 ± 0.02	0.52 ± 0.03	0.54 ± 0.03
Concatenation of SL	0.68 ± 0.02	0.55 ± 0.02	0.67 ± 0.03

Table 3: AUC ROC (mean ± std) for different machine learning models applied to the different processing of the brain connectivity data. Using homology dimensions 0 and 1

	Neural network	Logistic Regression	Random Forest
Multi-layer	$\mathbf{0.75 \pm 0.01}$	$\mathbf{0.69 \pm 0.03}$	$\mathbf{0.73 \pm 0.02}$
Single-Layer FA	0.58 ± 0.02	0.58 ± 0.04	0.60 ± 0.02
Single-Layer GM	0.55 ± 0.02	0.44 ± 0.03	0.46 ± 0.03
Single-Layer RS	0.64 ± 0.02	0.57 ± 0.02	0.52 ± 0.03
Concatenation of SL	0.69 ± 0.01	0.57 ± 0.02	0.66 ± 0.03

These results strongly suggest that the multi-layer architecture is the right choice to work with brain networks in a graph format. In particular, we see that

the multi-layer architecture encodes much more useful topological information, specially in dimensions 0 and 1.

It can be seen in the results that the GM layer has less predictive power than the DTI and RS-fMRI measurements. This fact is compatible with the results previously obtained in different experiments. The results also show that the combination of the information from the three single-layers graphs is, in general, better than each single-layer graph in isolation, but the multi-layer architecture is even better in almost all cases.

6 Conclusion and Future Work

We have presented a methodology that explores and analyses the multimodal brain networks, integrating structural brain connectivity, morphological gray matter and functional connectivity patterns, through the lens of persistent homology and Betti curves to extract features that are relevant to identify people with neurodegenerative diseases, such as multiple sclerosis.

Additionally, we empirically demonstrated that using a multi-layer architecture better preserves the brain mechanisms that drive cognitive processes and brain damage. Our experiments in a cohort of people with MS proved that features extracted using our proposed conceptual scheme are relevant to identify and classify both healthy volunteers and people with MS.

Some interesting directions for future research have been uncovered by this work. For instance, we plan to optimize and tune the hyperparameters of the models and add volumetric measurements to try to improve the accuracy of the machine learning models used in the study. Additionally, we intend to test our conceptual scheme on other neurodegenerative diseases, such as Alzheimer or Parkinson.

References

1. Aktas, M.E., Akbas, E., Fatmaoui, A.E.: Persistence homology of networks: methods and applications. Appl Netw Sci **4**, 61 (2019).
2. Bassett, D. S., Sporns, O.: Network neuroscience. Nature Neuroscience, **20**(3), 353–364. (2017)
3. Bennett, I.J., Rypma, B.: Advances in functional neuroanatomy: a review of combined DTI and fMRI studies in healthy younger and older adults. Neurosci. Biobehav. Rev. **37**, 1201–1210 (2013)
4. Caputi, L., Pidnebesna, A., Hlinka, J.: Promises and pitfalls of topological data analysis for brain connectivity analysis. *NeuroImage* **238** 118245 (2021)
5. Casas-Roma, J., Martinez-Heras, E., Solé-Ribalta, A., Solana, E., Lopez-Soley, E., Vivó, F., Diaz-Hurtado, M., Alba-Arbalat, S., Sepulveda, M., Blanco, Y., Saiz, A., Borge-Holthoefer, J., Llufriu, S., Prados, F.: Applying multilayer analysis to morphological, structural, and functional brain networks to identify relevant dysfunction patterns. Network Neuroscience, **6**(3): 916–933 (2022)

6. Chard, D. T., Alahmadi, A. A. S., Audoin, B., Charalambous, T., Enzinger, C., Hulst, H. E., Rocca, M. A., Rovira, À., Sastre-Garriga, J., Schoonheim, M. M., Tijms, B., Tur, C., Gandini Wheeler-Kingshott, C. A. M., Wink, A. M., Ciccarelli, O., Barkhof, F., MAGNIMS Study Group.: Mind the gap: From neurons to networks to outcomes in multiple sclerosis. Nature Reviews Neurology, **17**(3), 173–184 (2021)
7. Chou, Y.-H., Panych, L. P., Dickey, C. C., Petrella, J. R., Chen, N.-K.: Investigation of long-term reproducibility of intrinsic connectivity network mapping: A resting-state fMRI study. American Journal of Neuroradiology, **33**(5), 833–838 (2012)
8. Esteban, O., Markiewicz, C.J., Blair, R.W. et al.: fMRIPrep: a robust preprocessing pipeline for functional MRI. Nat Methods **16**, 111-116 (2019)
9. Fleischer, V., Radetz, A., Ciolac, D., Muthuraman, M., Gonzalez-Escamilla, G., Zipp, F., Groppa, S.: Graph theoretical framework of brain networks in multiple sclerosis: A review of concepts. Neuroscience, 403, 35–53 (2019)
10. Tauzin et al, J.: giotto-tda: A Topological Data Analysis Toolkit for Machine Learning and Data Exploration. *Mach. Learn. Res.* **22.39** (2021)
11. Groppa, S., Gonzalez-Escamilla, G., Eshaghi, A., Meuth, S. G., Ciccarelli, O.: Linking immune-mediated damage to neurodegeneration in multiple sclerosis: Could network-based MRI help?. Brain Communications, **3**(4), fcab237 (2021)
12. Llufriu, S., Martinez-Heras, E., Solana, E., Sola-Valls, N., Sepulveda, M., Blanco, Y., Martinez-Lapiscina, E. H., Andorra, M., Villoslada, P., Prats-Galino, A., Saiz, A.: Structural networks involved in attention and executive functions in multiple sclerosis. NeuroImage: Clinical, **13**, 288–296 (2017)
13. Pagani, E., Rocca, M. A., De Meo, E., Horsfield, M. A., Colombo, B., Rodegher, M., Comi, G., Filippi, M.: Structural connectivity in multiple sclerosis and modeling of disconnection. Multiple Sclerosis, **26**(2), 220–232 (2020)
14. Pun, C. S., Lee, S. X., Xia, K.: Persistent-homology-based machine learning: a survey and a comparative study. *Artif Intell Rev* **55** 5169–5213 (2022)
15. Rocca, M. A., Amato, M. P., De Stefano, N., Enzinger, C., Geurts, J. J., Penner, I.-K., Rovira, A., Sumowski, J. F., Valsasina, P., Filippi, M., MAGNIMS Study Group.: Clinical and imaging assessment of cognitive dysfunction in multiple sclerosis. Lancet Neurology, **14**(3), 302–317 (2015)
16. Rocca, M. A., Valsasina, P., Meani, A., Falini, A., Comi, G., Filippi, M.: Impaired functional integration in multiple sclerosis: A graph theory study. Brain Structure and Function, **221**(1), 115–131 (2016)
17. Solana, E., Martinez-Heras, E., Casas-Roma, J., Calvet, L., Lopez-Soley, E., Sepulveda, M., Sola-Valls, N., Montejo, C., Blanco, Y., Pulido-Valdeolivas, I., Andorra, M., Saiz, A., Prados, F., Llufriu, S.: Modified connectivity of vulnerable brain nodes in multiple sclerosis, their impact on cognition and their discriminative value. Scientific Reports, **9**(1), 20172 (2019)
18. Sporns, O.: Network attributes for segregation and integration in the human brain. Current Opinion in Neurobiology, **23**(2), 162–171 (2013)
19. Tijms, B. M., Seriès, P., Willshaw, D. J., Lawrie, S. M.: Similarity-based extraction of individual networks from gray matter MRI scans. Cerebral Cortex, **22**(7), 1530–1541 (2012)
20. Tournier, J.-D., Smith, R., Raffelt, D., Tabbara, R., Dhollander, T., Pietsch, M., Christiaens, D., Jeurissen, B., Yeh, C.-H., Connelly, A.: MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. NeuroImage, 202, 116137 (2019).
21. Umeda, Y.: Time Series Classification via Topological Data Analysis, Information and Media Technologies. *Information and Media Technologies* **12** (2017)

Program design and implementation of inclusion-exclusion integral neural network

Aoi Honda and Yoshihiro Fukushima

Kyushu Institute of Technology, 680-2 Kawazu, Iizuka, Fukuoka 820-8502, JAPAN
aoi@ai.kyutech.ac.jp

Abstract. This paper describes the implementation of a white-box inclusion-exclusion integral neural network using PyTorch. Our program is available on GitHub as the ieinn library. We provide an illustration of how to use the program with real data and how to extract information from the network using Shapley values and other methods.

Keywords: nonlinear integral, IE-integral, Explainable neural network

1 Introduction

Neural networks have achieved various results in the field of artificial intelligence and data science. However, the so-called black box problem, in which the process and basis of inference cannot be explained because the system itself is a black box, remains an issue. There are two approaches to addressing this problem: one is a "model-agnostic approach" that examines the input-output relationships of models to identify such relationships, while the other is a "model-specific approach" that takes a more transparent approach to the models themselves. To solve the black-box problem of neural networks, we have constructed a neural network that incorporates an inclusion-exclusion integral, hereafter abbreviated IE integral, mathematical model into a neural network as a white-box neural network, and have proposed performance and information extraction methods[1, 2]. Our experiments demonstrate that the proposed network model achieves comparable performance to standard neural networks, random forests, and other conventional methods in data analysis. Furthermore, we show that the IE integral-based approach allows us to extract meaningful information from the network after training, which can be useful for interpreting the results and improving the system.

Neural network models can be easily implemented using existing frameworks instead of writing custom programs from scratch. However, it can be difficult to significantly deviate from the standard formats provided by the framework. Although the IE integral neural network includes skip connections between nodes and is therefore distinct from standard network architectures, it can be implemented with some ingenuity using an existing framework. This paper presents an implementation of the Inclusive-Exclusive-Integral Neural Network(IEINN) using PyTorch, an open-source machine learning library, as well as instructions

on how to use the program. We also describe how to extract information from the trained network using Shapley values. The program is available on our public GitHub repository [3].

This paper is structured as follows. Section 2 provides an introduction to the minimal mathematical background, the inclusive integral mathematical model, and the inclusive integral neural network. In Chapter 3, we provide a detailed explanation of how to implement the Inclusive-Exclusive-Integral Neural Network using PyTorch. Section 4 describes specific data analysis methods using the implemented program. Finally, Section 5 summarizes the findings and discusses future research directions.

2 Inclusion-Exclusion integral neural network

2.1 Inclusion-Exclusion integral mathematical model

Here, we provide the necessary definitions and background for the implementation and analysis of the proposed IE integral neural network. Among non-additive measure integrals, a number of partition integrals have been proposed, and the IE integral is one of them ([1, 4, 5]). Further details and many of the results can also be found in [6].

In our model, we consider a set of explanatory variables that correspond to a finite set of J points, denoted as $X = 1, \ldots, j, \ldots, J$, with $\mathcal{P}(X)$ representing the power set. The inputs are assumed to be in the unit interval, such that $f = (x_1, x_2, \ldots, x_J) \in [0, 1]^J$, where x_i represents either the arguments themselves or the values after appropriate data transformations. We will use the notation $|A|$ to denote the cardinality of a subset $A \subseteq X$.

The IE integral is a non-additive measure that allows for integration with a measure that is not necessarily assumed additivity and includes the Lebesgue integral and the Choquet integral[7, 8] as special cases. As the measure's additivity is not assumed, interactions such as synergies and canceling effects can be expressed. The IE integral can be used to construct mathematical models with high representativity that can express the interaction between items. The mathematical model of the IE integral uses the Möbius representation of the IE integral instead of its defining equation.

Definition 1 (Möbius trans representation of IE integral [1]).
Let μ be a monotone measure[9] on $(X, \mathcal{P}(X))$, m^μ be the Möbius inversion of μ, and $\otimes$ be an interaction operator on $[0, 1]^{|A|}$ for $|A| = 1, \ldots, J$. Then, the IE integral of $f = (x_1, \ldots, x_j, \ldots, x_J)$ with respect to μ and $\otimes$ is

$$\int^{IE} f \, d\mu := \sum_{A \in \mathcal{P}(X)} \left(\bigotimes_{i \in A} x_i \right) m^\mu(A), \quad m^\mu(A) := \sum_{B \subseteq A} (-1)^{|A \setminus B|} \mu(B).$$

The interaction operators $\otimes$ in the IE integral can be expressed using t-norms such as the algebraic product. When the logical product is multiplied, the IE integral becomes equivalent to the Choquet integral. However, more general

operators that do not satisfy symmetry can also be used as interaction operators (for example, see [10]).

When the IE integral is employed as a regression model, the objective or target variable can be denoted as y and the explanatory variables as $x_1, \ldots, x_J$. Then, we can express the model as follows:

$$\hat{y} = f(x_1, \ldots, x_J) = \sum_{A \in \mathcal{P}(X)} \beta_A \left(\bigotimes_{i \in A} x_i \right).$$

Here, the coefficients β_A correspond to $m^\mu(A)$. Using the IE-integral model has an important advantage of being a flexible and expressive extension that maintains the white-box nature of the linear regression model. In this model, the parameters are considered as integral measures, allowing the extraction of information statistics such as the Shapley index [11, 12] from the data. Therefore, after the training process, important insights can be gained from the parameters, too. When the number of explanatory variables is J, an IE integral mathematical model typically consists of 2^J terms. However, to avoid an excessive increase in the number of terms or when there is no need to assume interactions involving more than a few items, it is possible to reduce the number of terms by imposing a k-order additivity condition on the measure [13]. A non-additive measure μ is said to be k-order additive if it satisfies the property $m^\mu(A) = 0$ for any $A \in \mathcal{P}(X)$ with $|A| > k$.

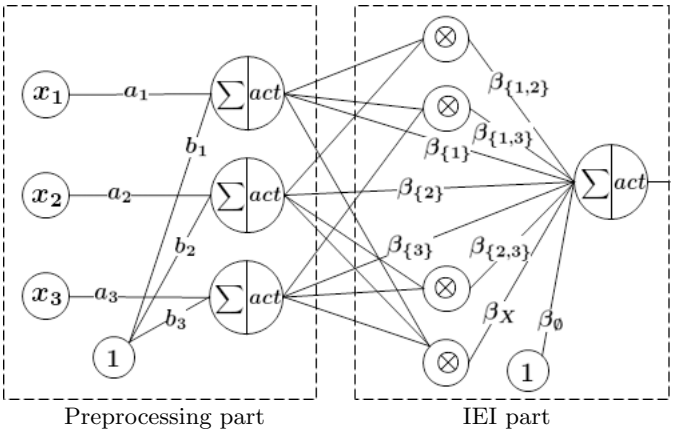


Fig. 1. IEINN model

2.2 Inclusion-exclusion integral neural network

The IE integral neural network is a network structure that is based directly on the Möbius transform representation of the IE integral. This enables not only optimized parameter estimation using the gradient method, but also automated preprocessing of inputs for the IE integral model. A network diagram for the case $J = 3$ is presented in Figure 1, with further details available in [6]. In contrast

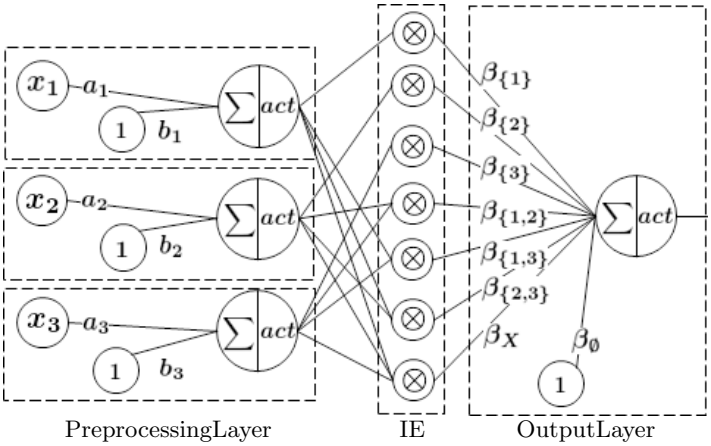


Fig. 2. implementation IE-integral neural network model

to general neural networks, the IE-integral network has unconnected units and unweighted edge and skip connections. It also includes units of aggregation operations that use multiplication-type operations, such as t-norms. The number of such units increases exponentially with the number of inputs, at most 2^J for J explanatory variables, i.e., exponentially with the number of inputs. As mentioned above, the use of a k-additive measure can alleviate the explosive growth of units.

3 Implementation of inclusion-exclusion integral neural network

We implemented the inclusion-exclusion integral neural network using the PyTorch library's torch.nn.Module, which is a module for building neural networks in Python. Although it is a special case, the inclusion-exclusion integral neural network can be considered a type of neural network. In the network diagram in Fig. 2, skip connections such as the edges with weights β_1, β_2, and β_3 can be treated as unary operations for convenience, allowing for relatively easy use of the neural network platform.

For the source code of the program, please refer to our "ieinn" library package, which is available on GitHub [3]. The "ieinn" module is included in the package. We imported torch, the main package of the PyTorch library, and used the nn package defined in it to create three classes: input layer, neural network (nn), and output layer, which inherit from the torch.nn.Module class. The class diagram of the "ieinn" module is shown in Fig. 3. In the following sections, we will describe these three layer classes.

3.1 PreprocessingLayer

The PreprocessingLayer class performs data preprocessing to prepare the input values for the IE integral layer. Specifically, one PreprocessingLayer instance

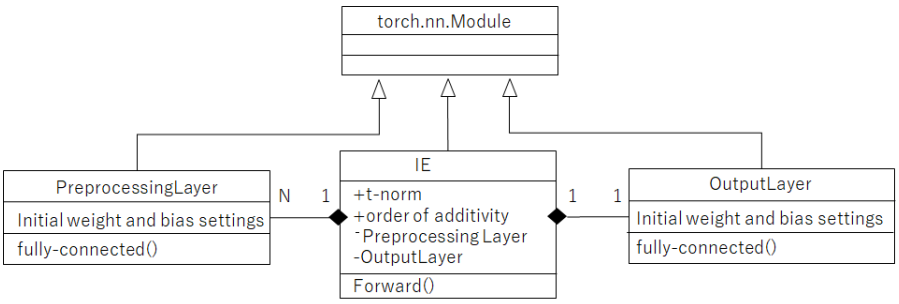


Fig. 3. Class diagram

is created for each feature, with all joins of 1 input and 1 output. Suppose there are J features and N instances of data, each feature being denoted by $X_j = x_{1,j}, x_{2,j}, \ldots, x_{N,j}$ and the objective variable being denoted by $Y = y_1, y_2, \ldots, y_N$. To ensure that the input values are normalized to the [0,1] interval and reasonably spread out, we apply the sigmoid function $s(x) = 1/(1+e^{-(ax+b)})$ as the activation function.

There are several options for initializing the weight and bias parameters, and we offer three methods, (i) to (iii), which can be selected based on the data distribution and the presence of outliers.

(i) PreprocessingLayerPercentile class To set the initial weights and biases, we use the 95th percentile ($X_j^{0.95}$) and the 5th percentile ($X_j^{0.05}$) of each feature. If the correlation coefficient between the objective variable and feature X_j is positive (negative), the weight and bias are initialized as follows: $weight_j = \pm 2 \times 2.94/(X_j^{0.95} - X_j^{0.5})$, $bias_j = \mp 2.94(X_j^{0.95} + X_j^{0.5})/(X_j^{0.95} - X_j^{0.5})$, respectively. Here, $s^{-1}(0.05) \approx -2.94$ and $s^{-1}(0.95) \approx 2.94$, so the 5th percentile value (95th percentile value) of feature X_j produces an output of 0.05 (0.95) when the correlation between X_j and the objective variable is positive.

(ii) PreprocessingLayerStandardDeviation class To determine initial values assuming that the data follows a normal distribution, the PreprocessingLayerNormal class is used. The mean and standard deviation of each feature are calculated from the training data, denoted as $E[X_j]$ and $SD[X_j]$, respectively. If the correlation coefficient between the objective variable and the feature is positive or negative, the weight and bias parameters are defined as $weight_j = \pm 2 \times 3.75/4SD[X_j]$, $bias_j = \mp 3.75 \times 2E[X_j]/4SD[X_j]$, respectively, $\Phi(2) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{2} e^{-\frac{1}{2}x^2} dx \approx 0.977, s^{-1}(0.977) \approx 3.75$. Here, $\Phi(2)$ denotes the cumulative distribution function of the standard normal distribution evaluated at 2, which is approximately 0.977. The inverse function of the sigmoid function evaluated at 0.977 is approximately 3.75, denoted as $s^{-1}(0.977) \approx 3.75$. Therefore, when the correlation between X_j and the objective variable is positive, the output is 0.977 when the input is $E[X_j] + 2SD[X_j]$, and when the correlation is negative, the output is 0.023 when the input is $E[X_j] - 2SD[X_j]$.

(iii) PreprocessingLayerMaxMin class In contrast to method (i) which employs the 95th and 5th percentile, the maximum and minimum values are utilized.

3.2 Inclusion-exclusion integral neural network class IEINN

This layer corresponds to the Interaction Operator layer in the IE integral network. The constructor, def_init_, defines the preprocessing layer and the output layer. The forward function implements the forward propagation process. The IE integral layer has no weight parameters that are updated during training. The set of n-term operators required to compute integrals, such as algebraic and logical products, is provided in the narray_op module, which is imported in the ieinn module. The original functions inherited from the torch.nn.Module class are available, along with several additional necessary functions.

3.3 OutputLayer

The IE layer, composed by the t-norm, outputs a single input that serves as an input to the instance, which corresponds to the output layer of the entire neural network. The number of inputs required depends on the additivity specification, with the default being $2^N - 1$ inputs without any additivity restrictions. This number is reduced if additivity restrictions are applied. The weights are stored in the order of $w_{\{1\}}, w_{\{2\}}, \ldots, w_{\{N\}}, w_{\{1,2\}}, \ldots, w_A, \ldots, w_{\{1,2,\ldots,N\}}$, where A corresponds to the Möbius transform of the fuzzy measure used in the integration. The weights and biases are adapted during training. The weight parameter w_A is initialized to $1/N$ for $A \in \mathcal{P}(X), |A| = 1$, and 0 otherwise. This initial value setting implies a fuzzy measure with maximum entropy [14].

4 Application to dataset for machine learning

In this section, we present a concrete analysis procedure using the program developed in the previous section and a real dataset. We demonstrate how to execute the program using Google Colaboratory, a Python execution environment provided by Google. This service offers a simple setup process for a Python environment and allows free access to GPUs. We have prepared a Python Notebook (.pynb) that performs the operations described here, and the "main.py" program for the data analysis is available in the library.

The dataset used in the analysis is the "Car Evaluation" dataset[15], which is designed for a regression problem. This well-known dataset consists of 1728 observation data with information on six features. The goal is to predict the overall evaluation value of each automobile. Table 1 provides an overview of the dataset. Since the raw feature values are ordinal categorical data in the form of word alternatives, they are converted to numerical values in order. Assume that the quantified data is stored in csv format in CarEvaluation.csv. This dataset is also available on GitHub.

Table 1. Car Evaluation Data

	attributes	alternatives	digitization	cor. w/. y
y	overall evaluation	(unaccept, acceptable, good, vry good)	(1,2,3,4)	1.00
x_1	buying price	(very high, high, medium, low)	(1,2,3,4)	0.28
x_2	maintenance cost	(very high, high, medium, low)	(1,2,3,4)	0.23
x_3	doors number	(2,3,4,more)	(1,2,3,4)	0.07
x_4	persons capacity	(2,4,more)a a	(1,2,3)	0.34
x_5	luggage size	(big, medium, small)	(1,2,3)	0.16
x_6	safety	(high, medium, low)	(1,2,3)	0.44

We will now proceed to explain the steps for executing the program. First, we mount Google Drive and download the ieinn library to it.

```
from google.colab import drive
drive.mount('/content/drive')
%cd /content/drive/MyDrive/
!git clone https://github.com/AoiHonda-lab/IEI-NeuralNetwork.git
import sys
sys.path.append("/content/drive/MyDrive/IEI-NeuralNetwork")
%cd IEI-NeuralNetwork/
```

Import the necessary libraries, including the IE integral network package, which is denoted by the filename "ieinn.py"[1].

```
import pandas as pd
import torch
import torch.nn as nn
import csv
from sklearn.model_selection import train_test_split
from ieinn import ieinn
```

If a GPU is available, set cuda to device, otherwise, set CPU.

```
# check GPU or CPU
device = 'cuda' if torch.cuda.is_available() else 'cpu'
print(device)
```

```
cuda
```

The data is processed by reading the CarEvaluation.csv file using pandas and creating the explanatory variable X and the objective variable y. After dividing the

[1] The subsequent actions can be obtained by executing "main.py" within the library.

data into training data and test data in appropriate proportions, convert them into the corresponding tensor type compatible with PyTorch. To split the dataset into training and validation data in arbitrary proportions, the train_test_split function from scikit-learn can be utilized. In this study, 80% of the data are randomly assigned to the training set, while the remaining 20% are assigned to the validation set. The data partitioning ratio can be adjusted according to the study requirements.

```
df=pd.read_csv
('CarEvaluation20221207.csv',encoding="shift jis")
df=df.drop(0,axis=0)
df=df.astype(float)
y=pd.DataFrame(df.iloc[:,0])
X=pd.DataFrame(df.iloc[:,1:])
# data Generating
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
↪  random_state=42)
X_train_df=X_train
y_train_df=y_train
# Extract as a numpy array with value and convert to tensor
X_train = torch.FloatTensor(X_train.values)
y_train = torch.FloatTensor(y_train.values)
X_test = torch.FloatTensor(X_test.values)
y_test = torch.FloatTensor(y_test.values)
```

The data content can be printed by the command **print(df)**.

```
print(df)
```

```
    evaluation price  maint   doors persons lug_boot  safety
1       1.0     1.0     1.0     1.0     1.0     1.0     1.0
2       1.0     1.0     1.0     1.0     1.0     1.0     2.0
3       1.0     1.0     1.0     1.0     1.0     1.0     3.0
 :
1727    3.0     4.0     4.0     4.0     3.0     3.0     2.0
1728    4.0     4.0     4.0     4.0     3.0     3.0     3.0
[1728 rows x 7 columns]
```

Next, the data for training is prepared by creating dataset for both training and testing, which are then passed as arguments to the **DataLoader**. During this process, the mini-batch size and data shuffling can be specified.

```
# Dataset creating
train_dataset = torch.utils.data.TensorD
ataset(X_train, y_train)
test_dataset = torch.utils.data.TensorDataset(X_test, y_test)
# DataLoade creating
train_loader = torch.utils.data.DataLoader(dataset=train_dataset,
↪  batch_size=128, shuffle=True, num_workers=2,
↪  generator=torch.Generator(device=device))
test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
↪  batch_size=128, shuffle=False, num_workers=2,
↪  generator=torch.Generator(device=device))
```

Create a model using the class IE imported by ieinn module and specify arguments such as a training data loader, the additivity order of the fuzzy measure, the polynomial operation used for IE-integral, and the preprocessing method. The training data loader is the "train_loader" created earlier, and the additivity order is an integer between 1 and the number of explanatory variables. Several t-norms such as logical product and algebraic product are available for polynomial operations. The preprocessing method can be selected from percentile, standard deviation, and maximum and minimum values. If the additivity order, polynomial operation, and preprocessing method are not specified, the defaults are complete non-additivity, algebraic product, and PreprocessingLayerPercentile, respectively.

```
model=IEINN.IE(train_loader, additivity_order=2,
↪  tnorm='Algebraic').to(device)
criterion = nn.MSELoss() #loss function
optimizer = torch.optim.Adam(model.parameters()) #Optimization method
print(model)
```

```
IE( (fc1): PreprocessingLayer()
     :
    (fc6): PreprocessingLayer()
    (fc2_0): Algebraic()
    (fc2_1): OutputLayer()
```

Check the initial parameters before training. The weights are $w_{\{1\}}, w_{\{2\}}, \ldots, w_{\{6\}}$, $w_{\{1,2\}}, w_{\{1,3\}}, \ldots, w_{\{5,6\}}$.

```
OrderedDict([('fc1.weight', tensor([[1.9600]])),
   ('fc1.bias', tensor([-4.9000])), ('fc2.weight', tensor([[1.9600]])),
```

```
   ('fc2.bias', tensor([-4.9000])), ('fc3.weight', tensor([[1.9600]])),
   ('fc3.bias', tensor([-4.9000])), ('fc4.weight', tensor([[2.9400]])),
   ('fc4.bias', tensor([-5.8800])), ('fc5.weight', tensor([[2.9400]])),
   ('fc5.bias', tensor([-5.8800])), ('fc6.weight', tensor([[2.9400]])),
   ('fc6.bias', tensor([-5.8800])),
   ('fc2_1.weight', tensor([[0.1667, 0.1667, 0.1667, 0.1667, 0.1667,
   ↪  0.1667, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
   ↪  0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
   ↪  0.0000]])),
    ('fc2_1.bias', tensor([0.]))])
```

Now that the network is ready, run the training. Specify train data loader, test data loader, criterion, optimizer, number of epochs, and device as arguments.

```
if (device == 'cuda'):
  torch.set_default_tensor_type('torch.cuda.FloatTensor')
history=model.fit_and_valid(train_loader, test_loader, criterion,
↪  optimizer, epochs=100, device=device)
```

```
Epoch [1/100], loss: 1.14647595 val_loss: 1.16615186
Epoch [2/100], loss: 0.95711808 val_loss: 0.97552333
 :
Epoch [100/100], loss: 0.27510643 val_loss: 0.29875589
time 28.790654182434082
```

Displays training results. Parameters after training can be displayed using the **model.state_dict** function as before training.

```
model.state_dict()
```

```
OrderedDict([('fc1.weight', tensor([[2.0274]])),
   ('fc1.bias', tensor([-4.8401])),('fc2.weight', tensor([[2.0580]])),
   ('fc2.bias', tensor([-4.7854])), ('fc3.weight', tensor([[2.0507]])),
   ('fc3.bias', tensor([-4.7873])), ('fc4.weight', tensor([[3.5346]])),
   ('fc4.bias', tensor([-5.3112])), ('fc5.weight', tensor([[3.0393]])),
   ('fc5.bias', tensor([-5.7852])), ('fc6.weight', tensor([[3.1182]])),
   ('fc6.bias', tensor([-5.9508])),
   ('fc2_1.weight', tensor([[ 0.2197, 0.2251, 0.1708, 0.3471, 0.2284,
   ↪  0.6126, 0.0568, 0.0568, 0.0568, 0.0568, 0.0568, 0.0488, 0.0488,
   ↪  0.0488, 0.0488, -0.0236, -0.0236, -0.0236, 0.1518, 0.1518,
   ↪  0.0583]])), ('fc2_1.bias', tensor([0.0434]))])
```

Functions for displaying the training progress using graphs, namely **plot**, **plot_train**, and **plot_test**, are provided.

```
model.plot()
```

There is also a function, **model.r2_score** score, to obtain the coefficient deter-

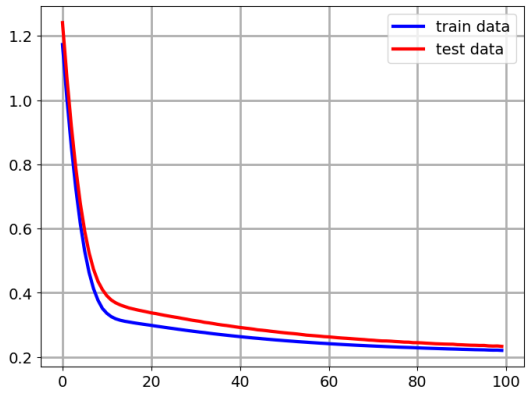


Fig. 4. Learning curves for train data and test data

mination.

```
model.r2_score(test_Loader)
```

```
0.6534947140654361
```

Information can be extracted from the parameters after training. In the preprocessing layer parameters, for example, fc1.weight $= 2.0274$, fc1.bias $= -4.8401$. Then we see, for example, that the explanatory variable x_1 is converted from $(1, 2, 3, 4)$ to $(0.0452, 0.2620, 0.7267, 0.9522)$ by passing through the preprocessing layer. Transform so that $f(x) = 1/[1 + exp(2.0274x - 4.8401)]$. fc2_1.weight is the weight of the IE integral layer, which corresponds to the Möbius transform of the non-additive measure. They shows that there is a synergistic effect between the importance of persons capacity and safety, with both being particularly highly valued. Conversely, the doors number and luggage size are complementary, meaning that either one of them should be satisfied. From the Möbius transform, the Shaplay values can be computed directly using

$$\phi_i(v) = \sum_{A \ni \{i\}} m^v(A)/|A|.$$

The Shapley value obtained from this result is $(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6) = (0.2293, 0.5040, 0.3746, 0.7346, 0.4886, 0.8200)$. From the Shapley values, we can see that the most important explanatory variable is safety, followed by a persons capacity.

5 Conclusions

In this paper, we presented a powerful approach to implement IE-integral neural networks for data analysis. The program has been made publicly available, which we hope will benefit the scientific community in their research. Our future work will focus on improving the program by adding more necessary functions to this library. We acknowledge that there are still many challenges that need to be overcome, such as developing more advanced preprocessing techniques for handling complex data and designing larger-scale neural networks using this framework. Nevertheless, we believe that this method has significant potential to advance the field of data analysis and information extraction.

References

1. A. Honda and Y. Okazaki. Theory of inclusion-exclusion integral. *Information Sciences*, Vol. 376, pp. 136–147, 2017.
2. A. Honda, Y. Kamata, and S. James. Representation and interpretability of IE integral neural networks. In *International Conference on Modeling Decisions for Artificial Intelligence*, pp. 168–180. Springer, 2022.
3. HondaAoi-Lab. https://github.com/AoiHonda-lab/IEI-NeuralNetwork, Inclusion-Exclusion Integral Neural Network Library. 2023.
4. E. Lehrer. A new integral for capacities. *Economic Theory*, Vol. 39, pp. 157–176, 2009.
5. R. Mesiar, J. Li, and E. Pap. Superdecomposition integrals. *Fuzzy Sets and Systems*, Vol. 259, pp. 3–11, 2015. Special issue Linz 2013: Non-Classical Measures and Integrals.
6. A. Honda, M. Itabashi, and S. James. Representation and interpretability of IE integral neural networks. *preprint*, 2021.
7. G. Choquet. Theory of capacities. *Contributions to the Theory of Games (AM-28)*, Vol. II, pp. 307–318, 1953.
8. R. Mesiar. Choquet-like integrals. *Journal of Mathematical Analysis and Applications*, Vol. 194, No. 2, pp. 477–488, 1995.
9. E. Pap. Null-additive set functions. 1995.
10. A. Honda and S. James. Averaging aggregation functions based on inclusion-exclusion integrals. In *IFSA-SCIS2017*.
11. L.S. Shapley. A value for n-person games. *Ann. Inst. Fourier*, Vol. 5, pp. 131–295, 1953.
12. M. Grabisch. *Set Functions, Games and Capacities in Decision Making*. Springer, Berlin, New York, 2016.
13. M. Grabisch. k-Order additive discrete fuzzy measures and their representation. *Fuzzy Sets and Systems*, Vol. 92, pp. 167–189, 1997.
14. A. Honda and M. Grabisch. Entropy of capacities on lattices and set systems. *Information Sciences*, Vol. 176, No. 23, pp. 3472–3489, 2006.
15. M. Bohanec and B. Zupan. Car evaluation. *UCI machine learning repository*, 1997. http://archive.ics.uci.edu/ml/datasets/Car+Evaluation.

Textual Explanations of Tabular Data*

Amber Zelvelder[1 [0000−0003−0357−9487]], Marcus Westberg[2 [0000−0001−5261−8898]], Tommy Löfstedt[1 [0000−0001−7119−7646]], and Kary Främling[1 [0000−0002−8078−5172]]

[1] Department of Computing Science, Umeå University, 901 87 Umeå, Sweden
amberez@cs.umu.se
[2] Delft University of Technology, 2628 CD Delft, Netherlands

Abstract. With the increased interest in and demand for explanations of decisions made by Machine Learning systems, and due to ethical issues and concerns raised by researchers and governmental bodies worldwide, as well as the proposed regulations connected to these, several techniques to create explanations have been developed. Several methods have been developed in recent years that seek to create textual explanations of data. In this paper we compare three of these methods: Anchor, LORE and CIU. We show the capabilities of these methods on tabular data and the type of explanations they generate. We also analyse their shortcomings and what would be needed in development to put these methods at the forefront of explainable AI.

Keywords: Explainable AI · Textual explanations · Contextual Importance and Utility · Anchor · LORE.

1 Introduction

The ultimate challenge of Explainable AI (XAI) is reaching out to end users. With so called Good Old Fashioned AI (GOFAI), this could be done by the developers of applications, as they knew the logic behind their system (see e.g. [5]). But with the opaqueness of Machine Learning (ML) methods, these same programmers don't know why the applications would have a specific outcome, leading to the so called black-box AI systems. This is a growing problem since the usage of black-box methods is expanding to cover more domains and to solve more complex problems. The less clear these processes are, the more they will increase the risk of leaving the end-user confused or upset about the outcome, in particular if it is undesired or unexpected.

A great amount of research is being done to provide some view into the way these methods work, or to approximate their working, to clarify their reasoning. In order to ensure user trust in AI as it becomes more widespread, solutions that provide explanations to end-users should be explored. This means letting a user know the underlying reason for an automated decision or judgement that is made. For example, if a user has had their loan application rejected, or if their insurance

* This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

fee has increased, the user should receive an explanation of the reasoning behind this even when the decision has been made by an AI system. Based on this the user could have a better understanding of the reasons behind the decision and could make a more informed decision on when to reapply or how to object to the decision. It is also important to see what type of explanations are the most valuable and clear to different end-users. As it is now, textual explanations seem to improve the possibilities to achieve this potential increased trust. One of the reasons for this is that there is a reduced risk of misunderstandings in comparison to interpreting visual explanations. In addition, text can also be made available to the visually impaired, by means of text-to-speech methods. In this paper we present an evaluation of three XAI methods that were designed to produce textual explanations.

2 Background

The roots of XAI can be found in expert systems explaining their results via applied rules. These early types of explanation methods date back to the 1970s and early 1980s [19, 18, 21]. The biggest difference in problems faced then and now is that current AI applications do not always operate on rule-based decision-trees but rather on opaque neural networks or other ML models. This poses an additional challenge to current XAI explanation methods that are required to not only provide a satisfactory explanation of a decision, but to do so through new mechanisms that can derive explanations from the output of systems whose decision-making methods cannot be interpreted or known directly. This is in great contrast to rule-based systems whose inference structures *can* be known directly by observing or being knowledgeable of the rules.

The leading categories of current XAI method development are post-hoc explanations [3, 13] and transparency by design [14], also called the glass box approach. In this work we deal with explanation methods of the former category. In this category of explanations, most methods of explanation concern ranking the main contributing features by their role or influence in the result, but there is also increased interest in counterfactual explanations for classification problems [20, 13], which indicate what changes could be done to change the outcome for a classification method (even though "counterfactual" is not limited to classification tasks per se). We will be looking at and comparing the performances of three different explanation methods for generating textual explanations of machine learning outputs.

The first explanation method is Contextual Importance and Utility (CIU) [7]. The original development of this method was inspired by Multiple-Criteria Decision Making (MCDM) methods such as Analytic Hierarchy Process (AHP) [17]. CIU computes the "Contextual Importance" (CI) and "Contextual Utility" (CU) of features for a given decision. These terms are loosely based on the existing definition of importance and utility in MCDM methods. The CI indicates the importance of a feature for the model and for a specific instance, or "context". The CU on the other hand indicates to what extent the studied instance's feature

values contribute towards a higher probability for a specific class in the case of classification, or whether it slides the result up or down in a regression problem. This also means that CIU, in a classification problem, can be used to explain why an instance does *not* belong to a specific other class. Implementations of CIU exist in R [8] and Python [2] programming languages.

The other two methods that we will be looking at — LORE and Anchor — both function by creating a decision-tree model that approximates the functioning of the AI model. The difference between these methods is how the decision-tree is converted into an explanation. LORE uses a rule-extraction method to provide explanations, where Anchor instead creates local regions known as "anchors" based on the decision-tree.

Anchor was first proposed in 2018 [16], by the same authors that developed LIME [15]. LIME is an earlier explanation method, that has a focus on visual explanations and was published in 2016. Anchor was initially developed as an extension of LIME, to provide explanations in a new format, including textual explanations. The internal difference to LIME is that Anchor uses a local region, known as an "Anchor" to create explanations. The Anchor method is model-agnostic, so it should work with any type of ML model, but is set up to be only available for classification problems. Since part of the methods is to create a decision tree, based on the model, the results from these explanations come as a set of rules that represent the decision nodes that are touched upon to reach the final classification. Anchor is available as an independent Python package [16] and is also included in the "alibi" library [12], which is a Python library that contains a collection of different explanation methods.

LORE (LOcal Rule-based Explanations) is still in an early development stage [9]. The method has been developed as an improvement on the faithfulness of textual explanations. It is designed to explain model outcomes in binary classification problems using a sequence of neighborhood generation, rule extraction, and optionally counterfactual extraction. Unlike Anchor and CIU, the design of LORE is not centered around a definition, but around the combined use of a set of methods, such as genetic algorithms [11]. The first step of LORE is to find neighborhoods for each feature in which the possible outcomes reside. After neighborhood generation, the neighborhoods are merged and a decision tree is built from these merged neighborhoods. Rules are extracted from this decision and given as the explanation for the studied instance, as well as all instances whose feature values satisfy the rule conditions. It ls also possible to identify so called counterfactual instances by identifying instances that do not satisfy the rules [9]

3 Methodology

To strive towards objectivity in the tests, we approximated the same setup for the different methods as much as possible. To begin, for each dataset we used the same file with data, in this case a data format file, and separated the training and testing dataset the same way. As the different methods expect different inputs,

and a significant amount of pre-processing, a perfect simulacrum of the models was not easily achieved. Our hope was that this could be countered by using the same seeds and bootstrapping a larger number of tests and outputs with the same seeds.

LORE needs access to the possible outcomes, the entire dataset kept in memory and the path to the source data file (access to the entire dataset is not enough), to run their explanation function. Since LORE is designed for binary classification, it neither functions with classification of multiple outputs, regression problems, nor non-tabular data. This will be reiterated in the results and addressed in the discussion section.

To run Anchor, a list with all features and their possible values, the possible outcomes and the training data were required as these are not automatically extracted from the data. Anchor is designed around classification, so regression problems are not within their designed functionality.

In the R implementation of CIU, the only needed inputs were the model and the possible outcomes. We used the R implementation of CIU in our tests [8].

The first dataset we used for our test is the "adult" dataset, also known as the "census income" dataset, from the University of California Irvine Machine Learning Repository [4]. It is a classification dataset with two possible outcomes. The outcomes are to predict whether or not the instance's yearly income exceeds $ 50k. As all three methods are capable to work with binary classification data, the "adult" dataset can be considered as a baseline dataset, for which the quality and accuracy of the explanations can be shown.

The next dataset is a classification dataset with more than two possible result classes, which excludes LORE as a method. We used the Iris dataset [1, 6], for its simplicity and since both CIU and Anchor have shown to work for it previously [7, 16].

Finally we tested the explanations with a regression dataset. For this we decided on the Boston Housing dataset [10], as it is widely used for benchmarking regression problems.

During our tests, we made general observations of the different algorithms, to see how they themselves evaluate their accuracy, and in what form they present their explanations. We started by observing a typical explanation of a single instance. We then proceeded to run the explanation method ten times on the same model and instance. In this time we made note of the following factors.

- Self-reported accuracy of the result of the explanation, when the method has this as a functionality.
- How rapidly the model generates an explanation, measured by the `system.time()` function's "Elapsed" value.
- Consistency of the results, i.e. whether the explanations remain the same over multiple runs of a method for the same instance. We used 10 runs in all experiments.
- Overall agreement in generated results, and to what extent it can vary per case. (*e.g.*, what features are found by all methods to be 'important').

4 Results

We trained a random forest model for each of the datasets, which gave a good prediction accuracy for all of them. For each dataset, we have included a figure illustrating an instance of the explanations for each of the different methods. This is to give an indication on the form the explanation takes in each method. We then present the results of running each method ten times on the same instance, to show their consistency and speed.

4.1 Adult dataset

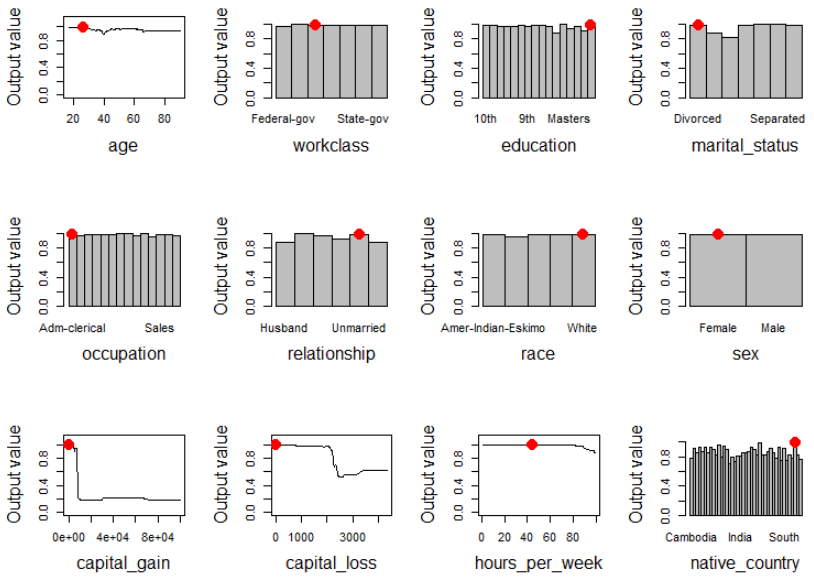


Fig. 1. Effect of adult dataset features on the probability of the studied instance to belong to the "`<=50k`" class, with the values of the studied instance highlighted in red.

The Adult dataset is a binary classification dataset. Fig. 1 illustrates the effect of varying the feature values on the probability of the studied instance to belong to the class "`<=50k`", which signifies having a yearly income of less than $50 000. The red dot in the figure shows the feature values of the instance we used, which are the following: **age**: *27*, **workclass**: *'Private'*, **education**: *'Some-college'*, **marital-status**: *'Divorced'*, **occupation**: *'Adm-clerical'*, **relationship**: *'Unmarried'*, **race**: *'White'*, **sex**: *'Female'*, **capital-gain**: *0*, **capital-loss**: *0*, **hours-per-week**: *44*, **native-country**: *'United-States'*.

```
r = {'capital-gain': '<=5178', 'age': '<=36'} --> {'class': '<=50K'}
Delta:
delta {'class': '<=50K'}
delta {'capital-gain': '<=5178', 'age': '<=36'}
delta [541.0, 1.3]
Coverage:
3078
```

Fig. 2. LORE explanation and delta for the adult dataset.

```
Prediction:  <=50K
Anchor: Age <= 29.00 AND Relationship > 1.00
Precision: 0.99
Coverage: 0.17
```

Fig. 3. Anchor explanation for the Adult dataset.

```
The value of output '<=50K' for instance '22279' is 0.988, which is very good (CU=0.988).
Feature 'capital_gain' is extremely important (CI=0.814) and value '0' is very good (CU=0.995).
Feature 'capital_loss' is important (CI=0.472) and value '0' is very good (CU=1).
Feature 'native_country' is slightly important (CI=0.29) and value 'United-States' is very good (CU=1).
Feature 'marital_status' is not important (CI=0.172) and value 'Divorced' is very good (CU=0.965).
Feature 'relationship' is not important (CI=0.12) and value 'Unmarried' is very good (CU=0.95).
Feature 'hours_per_week' is not important (CI=0.116) and value '44' is very good (CU=0.897).
Feature 'education' is not important (CI=0.11) and value 'Some-college' is very good (CU=0.945).
Feature 'age' is not important (CI=0.06) and value '27' is very good (CU=1).
Feature 'occupation' is not important (CI=0.04) and value 'Adm-clerical' is very good (CU=0.95).
Feature 'race' is not important (CI=0.04) and value 'White' is very good (CU=1).
Feature 'workclass' is not important (CI=0.022) and value 'Private' is average (CU=0.545).
Feature 'sex' is not important (CI=0.006) and value 'Female' is very good (CU=1).
```

Fig. 4. Example of CIU output for the adult dataset.

Feature	LORE	Anchor	CIU
capital gain	×	+	×
capital loss	−	−	×
native country	−	−	×
marital status	−	×	+
relationship	+	+	+
hours per week	+	−	+
education	−	+	+
age	×	×	−
occupation	−	+	−

Table 1. The features of the adult dataset. Features consistently included or considered very important are indicated with an ×, features only sporadically included, or with a lower but still notable importance is indicated with a +. Features that are not included in LORE and Anchor explanations or that are considered not important by CIU are indicated with a −.

LORE. The explanation provided by LORE, illustrated in Fig. 2, shows the decision-tree steps required to confidently explain the right outcome. Out of the 10 explanations, all included a "capital gain" below a threshold value that varied between 3589 and 7688. For 8 out of 10 runs, "age" was included with values below thresholds that varied from 27 to 59. One run produced a rule where

"age" should be between 23 and 30. The value of the feature "relationship" being "unmarried" was included in the decision-tree one time and "hours per week" being between 42 and 45 was also included once. No other features were included in the decision-tree to explain the outcome of an income below 50k. Each time the explanation was generated, it took between 17.3 and 17.9 seconds.

Anchor. Anchor performed within its own definition of acceptable accuracy, above 95 percent. Out of the 10 runs on the same instance, all of the anchors included "Age $\leq$ 29.00". In one run, "Age $\leq$ 29.00" was the only rule included in the anchor explanation. This rule was most often paired with the rule "Marital Status $>$ 1.00", which occurred in 5 out of 10 Anchors as the Anchor explanation. After that "Occupation $\leq$ 2.00" was paired with "Age" twice. The "Relationship" feature was also paired with "Age" twice, but with a variation, one was "Relationship $>$ 0.00" and the other was "Relationship $>$ 1.00". In two results, a third feature appeared as part of the Anchor, paired with "Age" and "Marital Status". These were "Education $>$ 1.00" and "Capital gain $\leq$ 0.00".

The one run that was different cited the Anchor as "Age $\leq$ 29.00 AND Hours per week $\leq$ 45.00 AND Education $>$ 1.00". The Precision function that Anchor uses to estimate how true to the model the explanation had values in the range 0.95 to 0.99. The time for each explanation to run was in the range 0.068 to 0.159 seconds.

CIU. For CIU, the 10 runs all gave exactly the same results. Capital gain and loss both have high importance and utility, with native country having some importance and high utility. For the studied instance, all values except "workclass" are of high utility, which in this case signifies that they are typical for instances that belong to the class "`<=50k`". Looking at the most important features, that didn't make the "slightly important" threshold, it shows marital status, education, relationship, and hours per week.

4.2 Iris dataset

The Iris dataset is a classification dataset with multiple outputs. Fig. 5 illustrates the effect of varying the feature values on the probability of the studied instance to belong to the class "`virginica`". The red dot in the figure shows the feature values of the studied instance, i.e.: **Petal.Length**: *5.1*, **Petal.width**: *1.9*, **Sepal.Length**: *5.8*, **Sepal.Width**: *2.7*.

LORE. LORE was not designed to work with non-binary classification, and no results are therefore available here.

Anchor. All 10 runs with the studied instance produced the Anchor explanation "petal width (cm) $>$ 1.80 AND sepal width (cm) $\leq$ 2.80". The Precision function ranged between 0.97 and 0.99. The time for each explanation to run ranged from 0.111 to 0.138 seconds.

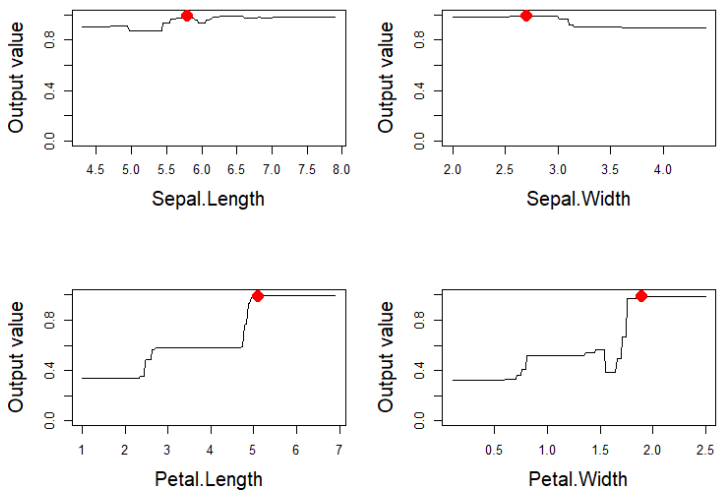


Fig. 5. This figure displays the effect that the different features have on the estimated probability of the studied instance being an Iris Virginica.

```
Prediction:  virginica
Anchor: petal width (cm) > 1.80 AND sepal width (cm) <= 2.80
Precision: 0.98
Coverage: 0.08
```

Fig. 6. Anchor output for the Iris dataset.

```
The value of output 'setosa' for instance '143' is 0, which is very bad (CU=0).
Feature 'Petal.Length' is important (CI=0.448) and value '5.1' is very bad (CU=0).
Feature 'Petal.Width' is important (CI=0.402) and value '1.9' is very bad (CU=0).
Feature 'Sepal.Length' is not important (CI=0.012) and value '5.8' is very bad (CU=0).
Feature 'Sepal.Width' is not important (CI=0.008) and value '2.7' is very bad (CU=0).
The value of output 'versicolor' for instance '143' is 0.012, which is very bad (CU=0.012).
Feature 'Petal.Width' is very important (CI=0.606) and value '1.9' is very bad (CU=0.003).
Feature 'Petal.Length' is important (CI=0.412) and value '5.1' is very bad (CU=0.01).
Feature 'Sepal.Length' is not important (CI=0.114) and value '5.8' is very bad (CU=0.018).
Feature 'Sepal.Width' is not important (CI=0.09) and value '2.7' is very bad (CU=0.067).
The value of output 'virginica' for instance '143' is 0.988, which is very good (CU=0.988).
Feature 'Petal.Width' is very important (CI=0.666) and value '1.9' is very good (CU=0.997).
Feature 'Petal.Length' is very important (CI=0.654) and value '5.1' is very good (CU=0.994).
Feature 'Sepal.Length' is not important (CI=0.118) and value '5.8' is very good (CU=0.983).
Feature 'Sepal.Width' is not important (CI=0.096) and value '2.7' is very good (CU=0.937).
```

Fig. 7. CIU output for the Iris dataset, including "why" explanation for the actual class ("virginica") and "why not" explanations for the other two classes.

CIU. For CIU, the 10 runs all gave identical results. All of the values indicated that the output was a Virginica, with Petal Length and Petal Width being the most important features.

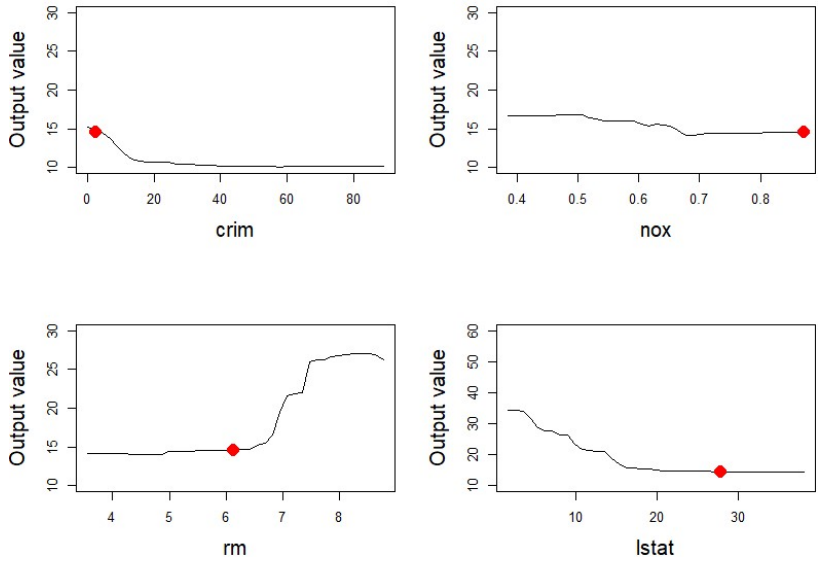


Fig. 8. Effect of the less linear features of the Boston Housing data on the estimated value.

4.3 Boston housing dataset

The Boston Housing dataset is a regression dataset, with a numerical output. Fig. 8 presents some of the features that have a highly non-linear effect on the predicted price for the used instance, "14.585". The red dot in the figures indicates the values of the studied instance.

```
The value of output 'medv' for instance '146' is 14.585, which is bad (CU=0.213).
Feature 'lstat' is important (CI=0.451) and value '27.8' is very bad (CU=0.005).
Feature 'rm' is slightly important (CI=0.29) and value '6.13' is very bad (CU=0.027).
Feature 'crim' is not important (CI=0.12) and value '2.37934' is good (CU=0.797).
Feature 'nox' is not important (CI=0.062) and value '0.871' is very bad (CU=0.176).
Feature 'indus' is not important (CI=0.04) and value '19.58' is bad (CU=0.288).
Feature 'age' is not important (CI=0.036) and value '100' is very bad (CU=0).
Feature 'tax' is not important (CI=0.034) and value '403' is good (CU=0.789).
Feature 'dis' is not important (CI=0.021) and value '1.4191' is very bad (CU=0).
Feature 'ptratio' is not important (CI=0.012) and value '14.7' is very good (CU=0.95).
Feature 'black' is not important (CI=0.011) and value '172.91' is bad (CU=0.302).
Feature 'rad' is not important (CI=0.003) and value '5' is bad (CU=0.227).
Feature 'chas' is not important (CI=0.002) and value '0' is very bad (CU=0).
Feature 'zn' is not important (CI=0.001) and value '0' is very bad (CU=0).
```

Fig. 9. Example of CIU output of the Boston housing dataset.

LORE. LORE is not designed for regression problems, and therefore failed to give an accurate explanation. As a regression model is not part of the LORE design, it is removed from this comparison.

Anchor. Anchor attempted to explain the Boston housing dataset, but failed to meet it's own internal criteria of precision. It treated all different outcomes as classes instead of identifying it as a regression problem. This resulted in an error of Anchor not meeting the self-defined precision criterion, even when lowering the precision threshold to 0.7 (from the default 0.95). It can still be run on the data, but will create an Anchor that includes all features and has low precision, which is not what the method is intended to do.

CIU. CIU displayed identical results for every run, shown in Fig. 9. The estimated output value (`medv`="Median value") is shown with the descriptor 'bad' because it is a low value compared to the other instances in the data set. This result is largely explained by the most important two features, 'lstat' and 'rm' that are both 'very bad', but some of the 'not important' features are 'good' or 'very good', which in this case puts the overall value up to just 'bad', rather then 'very bad'.

5 Discussion

Although our initial goal was to apply all three methods to all the studied data sets, we rapidly realized that not all methods or implementations were designed for all types of tabular data. In the results, each method provided some form of text explanation for binary classification and showed some amount of consistency for the same instance. For the instance used, LORE was significantly slower then the others. For the classification on the IRIS dataset, the results for both Anchor and CIU were identical and of similar speed. Since only CIU worked as expected for the regression dataset, no comparison is possible.

Apart from LORE and Anchor not working for regression at all, all three methods had strengths and weaknesses. In the case of LORE the greatest weakness was that it only worked for binary output, and current implementations take long to generate the explanations. In contrast, the Anchor and CIU implementations are well suited for such tasks and both run relatively rapidly. The CIU method was the most consistent in generating explanations, since for all three tests the explanations were identical on every run. The clarity of the explanation is subjective, but Anchor and LORE have the benefit of keeping the explanation concise, whereas for CIU, the explanation includes all features unless specifically bounded by a stated threshold. This also ascribes a benefit to CIU, in that more of the instance is included in the explanation, so lesser contributing factors are also included.

6 Conclusion

In this paper we evaluated three methods of generating textual explanations, LORE, Anchor and CIU. We found that while all methods can explain a binary classification outcome, Anchor and CIU can explain multiple output classification and only CIU can generate an explanation for a regression problem. When comparing the methods, we found that CIU is the most consistent in the explanation it produced. To truly know which method is better for generating textual explanations, further study must be done using these methods. Instead of a focus on technical limitations of the methods, a user study on how far these explanations take their understanding and if they are satisfied with such output as an explanation would be a valuable next step.

Acknowledgements This research was was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

1. Andrews, D.F., Herzberg, A.M.: Iris Data, pp. 5–8. Springer New York, New York, NY (1985)
2. Anjomshoae, S., Kampik, T., Främling, K.: Py-CIU: A Python Library for Explaining Machine Learning Predictions Using Contextual Importance and Utility. In: IJCAI-PRICAI 2020 Workshop on Explainable Artificial Intelligence (XAI) (2020)
3. Anjomshoae, S., Najjar, A., Calvaresi, D., Främling, K.: Explainable agents and robots: Results from a systematic literature review. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. p. 1078–1088. AAMAS '19, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2019)
4. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
5. Fagan, L., Shortliffe, E., Buchanan, B.: Computer-based medical decision making: from MYCIN to VM. Automedica **3**, 97–106 (1980)
6. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of eugenics **7**(2), 179–188 (1936)
7. Främling, K.: Decision theory meets explainable AI. In: International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems. pp. 57–74. Springer (2020)
8. Främling, K.: Contextual Importance and Utility in R: the 'ciu' Package. In: Proceedings of 1^{st} Workshop on Explainable Agency in Artificial Intelligence, at 35^{th} AAAI Conference on Artificial Intelligence. pp. 110–114 (2021)
9. Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., Giannotti, F.: Local rule-based explanations of black box decision systems. arXiv:1805.10820 (2018)
10. Harrison, D., Rubinfeld, D.: Hedonic housing prices and the demand for clean air. Journal of Environmental Economics and Management **5**, 81–102 (1978)

11. Holland, J.H.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press (1992)
12. Klaise, J., Looveren, A.V., Vacanti, G., Coca, A.: Alibi explain: Algorithms for explaining machine learning models. Journal of Machine Learning Research **22**(181), 1–7 (2021)
13. Molnar, C.: Interpretable Machine Learning: A Guide For Making Black Box Models Explainable (2020)
14. Murmann, P., Fischer-Hübner, S.: Tools for achieving usable ex post transparency: a survey. IEEE Access **5**, 22965–22991 (2017)
15. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should I trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA. pp. 1135–1144 (2016)
16. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-precision model-agnostic explanations. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
17. Saaty, T.L.: The analytic hierarchy process : planning, priority setting, resource allocation. McGraw-Hill International Book Co., New York; London (1980)
18. Scott, A.C., Clancey, W.J., Davis, R., Shortliffe, E.H.: Explanation capabilities of production-based consultation systems. Tech. rep., Stanford University, Department of Computer Science (1977)
19. Shortliffe, E.H., Davis, R., Axline, S.G., Buchanan, B.G., Green, C., Cohen, S.N.: Computer-based consultations in clinical therapeutics: Explanation and rule acquisition capabilities of the mycin system. Computers and Biomedical Research **8**(4), 303–320 (1975)
20. Sokol, K., Flach, P.: Counterfactual explanations of machine learning predictions: Opportunities and challenges for ai safety. In: Proceedings of the AAAI Workshop on Artificial Intelligence Safety (SafeAI 2019). CEUR Workshop Proceedings, vol. 2301. CEUR Workshop Proceedings (2019)
21. Swartout, W.R.: Explaining and justifying expert consulting programs. In: Computer-assisted medical decision making, pp. 254–271. Springer (1985)

Some examples of probabilistic metric spaces by means of fuzzy measures

Yasuo Narukawa[1],Vicenç Torra[2]

[1] *Department of Management Science,*
Tamagawa University,
6-1-1 Tamagawagakuen, Machida,
Tokyo, 194-8610, Japan
[2] *Department of Computing Sciences, Umeå University,*
Umeå, Sweden
Email: nrkwy@eng.tamagawa.ac.jp,vtorra@ieee.org

Abstract. Probabilistic metric spaces generalize metric spaces. While in a metric space the metric is a function that returns a number representing the distance, in a probabilistic metric space the corresponding function returns a distribution. In a recent work we have introduced probabilistic metric spaces based on fuzzy measures. They are the f-spaces. In this paper we provide some new examples and results for this family of probabilistic metric spaces.

1 Introduction

Probabilistic metric spaces [9,10] were introduced as a generalization of metric spaces. In probabilistic metric spaces we deal with distribution functions. They play the role of the distance. The triangle inequality is then replaced by an inequality on distribution functions. More precisely, the inequality is based on triangle functions. Triangle functions are connected with t-norms [1].

Several families of probabilistic metric spaces have been introduced in the literature. Menger spaces, E-spaces [11,12], and F-spaces are of relevance in this work. We introduced F-spaces recently [7] as a generalization or extension of E-spaces. While in E-spaces the probabilistic metric space is based on a probability distribution, in F-spaces we replace the probability with a fuzzy measure. Then, under some conditions, the construction also produces a probabilistic metric space with some interesting properties.

In this paper we further study these probabilistic metric spaces, provide some results based on the Choquet integral, and include some examples. Recall that the Choquet integral is a generalization of the Lebesgue integral when the measure is not additive, and, thus, permits to integrate a function with respect to a non-additive (fuzzy) measure. See e.g. [3,13,14,5] for details on the Choquet integral.

The structure of this paper is as follows. We begin in Section 2 with some preliminaries. Then, in Section 3 we review the probabilistic metric spaces we

need in our work. Section 4 includes our main results on the F-space and the Choquet integral. The paper finishes with a summary.

2 Preliminaries

In this section we review t-norms, and some results related to fuzzy measures and integrals.

2.1 t-Norms

We will use in this paper the concept of t-norm. They are functions on $[0,1]\times[0,1]$ that generalize Boolean conjunction. The definition follows.

Definition 1. *A function $\top : [0,1] \times [0,1] \to [0,1]$ is a t-norm if and only if it satisfies the following properties:*

(i) $\top(x,y) = \top(y,x)$ *(symmetry or commutativity)*
(ii) $\top(\top(x,y),z) = \top(x,\top(y,z))$ *(associativity)*
(iii) $\top(x,y) \leq \top(x',y')$ *if $x \leq x'$ and $y \leq y'$ (monotonicity)*
(iv) $\top(x,1) = x$ *for all x (neutral element 1)*

They are usually required to satisfy also continuity and subidempotency (i.e., $\top(x,x) < x$ for $x \neq 0$). Such t-norms are called Archimedean t-norms.

Some examples of t-norms follow.

Example 1. The following functions are t-norms.

- **Minimum:** $\top(x,y) = \min(x,y)$. The minimum is often denoted by $\wedge$. That is, $x \wedge y = \min(x,y)$.
- **Algebraic product:** $\top(x,y) = xy$. This t-norm is denoted by $\Pi(x,y)$ following [10].
- **Bounded difference or Lukasiewicz t-norm:** $\top(x,y) = \max(0, x+y-1)$. This t-norm is denoted by $W(x,y)$ following [10].
- **Yager family:** $\top_w(x,y) = 1 - \min\left(1, ((1-x)^w + (1-y)^w)^{1/w}\right)$ for $w \geq 0$.
- **Drastic:** $\top_d(x,y) = y$ if $x = 1$, $\top_d(x,y) = x$ if $x = 1$, and $\top_d(x,y) = 0$ otherwise.

It easy to see that t-norms are proper generalizations for conjunctions, as, for all of them, $\top(0,0) = \top(0,1) = \top(1,0) = 0$ and $\top(1,1) = 1$.

2.2 Fuzzy measure and the Choquet integrals

This section introduces some concepts related to fuzzy measure and the Choquet integrals that are needed later on in this paper.

We use X to denote the reference set. Then, let $\mathcal{A}$ denote a subset of the power set of X (i.e., 2^X) such that $\emptyset \in \mathcal{A}$. An element of $\mathcal{A}$ is said to be a *fuzzy measurable set* and, then, $(X,\mathcal{A})$ is a *fuzzy measurable space*. In addition, we say that a function $f : X \to R$ is *measurable* when $\{x|f(x) > r\} \in \mathcal{A}$ for all $r \in R$. We denote the class of measurable functions by $\mathcal{M}$. In addition, we denote by $\mathcal{M}^+$ the class of non-negative measurable functions.

Definition 2. *A fuzzy measure (also known as non-additive measure and capacity) μ is a real valued set function $\mu : \mathcal{A} \longrightarrow [0,1]$ that satisfies the following properties:*

(i) $\mu(\emptyset) = 0$ *(boundary condition)*
(ii) $\mu(X) = 1$ *(boundary condition)*
(iii) $A \subset B$ *for* $A, B \in \mathcal{A}$ *implies* $\mu(A) \leq \mu(B)$ *(monotonicity condition)*

If $\mu(X) < \infty$, the conjugate of a measure μ, denoted by μ^c, is defined by $\mu^c(A) = \mu(X) - \mu(A^C)$ for $A \in \mathcal{A}$.

Among the different families of fuzzy measures, some are of interest in this work. We define them below.

Definition 3. *Let X be a set, then we consider the following fuzzy measures on $(X, \mathcal{A})$.*

(i) *Probability measures. A measure P is a probability measure if it satisfies the additivity axiom. That is, for all $A \cap B = \emptyset$ we have that*

$$P(A \cup B) = P(A) + P(B),$$

and, in addition,

$$P(X) = 1.$$

(ii) *Possibility measures. A measure Pos is a possibility measure if it satisfies the following axiom*

$$Pos(A \cup B) = \max(Pos(A), Pos(B))$$

for all A, B. These measures were introduced by Zadeh [17] in the context of fuzzy sets.
(iii) *Necessity measure. A measure Nec is a necessity measure if it satisfies*

$$Nec(A \cap B) = \min(Nec(A), Nec(B))$$

for all A, B. These measures were also introduced by Zadeh [17] and they can be defined as the conjugate of possibility measures.
(iv) *The 0-1 possibility measure Pos_A focused on a set $A \subseteq X$. Given a set A we define the measure as follows.*

$$\mathrm{Pos}_A(B) = \begin{cases} 1 & \text{if } A \cap B \neq \emptyset \\ 0 & \text{if } A \cap B = \emptyset \end{cases}$$

(v) *The 0-1 necessity measure Nec_A focused on a set $A \subseteq X$. This measure is defined as follows, and corresponds to the unanimity game [5].*

$$\mathrm{Nec}_A(B) = \begin{cases} 1 & \text{if } A \subseteq B \\ 0 & \text{if } A \not\subset B \end{cases}$$

Definition 4. *Let μ be a fuzzy measure on the measurable space $(X, \mathcal{A})$. Then,*

– μ *is submodular if* $\mu(A) + \mu(B) \geq \mu(A \cup B) + \mu(A \cap B)$*;*
– μ *is supermodular if* $\mu(A) + \mu(B) \leq \mu(A \cup B) + \mu(A \cap B)$*.*

Definition 5. *Let* φ *be a real valued function on closed interval [c,d]. Then,*

– φ *is said to be convex if*

$$\varphi(\lambda x + (1-\lambda)y) \leq \lambda\varphi(x) + (1-\lambda)\varphi(y)$$

for $x, y \in [c,d]$*,* $0 < \lambda < 1$*, and*
– φ *is said to be concave if*

$$\varphi(\lambda x + (1-\lambda)y) \geq \lambda\varphi(x) + (1-\lambda)\varphi(y)$$

for $x, y \in [c,d]$*,* $0 < \lambda < 1$*.*

The following results are of relevance.

Proposition 1. [8] Let μ be a non-additive measure on $(X, \mathcal{A})$, and $\varphi : [0,1] \to [0,1]$ be a non-decreasing function with $\varphi(0) = 0$ and $\varphi(1) = 1$.

(i) If φ is convex, then $\varphi \circ \lambda$ is supermodular.
(ii) If φ is concave, then $\varphi \circ \lambda$ is submodular.

We begin introducing the Choquet integral which was introduced by Choquet [2]. This integral generalizes the Lebesgue integral. More particularly, when the measure is additive (as we require $\mu(X) = 1$ this measure will be a probability), the Choquet integral of f is the Lebesgue integral of f. Thus, the Choquet integral of f with respect to μ corresponds to the expectation of f. For details on fuzzy measures and integrals see e.g. [13,14,5].

Definition 6. *[2] Let* X *be a set, let* f *be a function on* X *as above, and let* μ *be a fuzzy measure on* $(X, \mathcal{B})$*. Then, the Choquet integral of* $f \in \mathcal{M}^+$ *with respect to* μ *is defined by*

$$(C)\int f d\mu = \int_0^\infty \mu_f(r)dr,$$

where $\mu_f(r) = \mu(\{x | f(x) \geq r\})$*.*

We also need to consider the restriction of the integral on a set. Let $A \in \mathcal{B}$ *be such set. Then, the Choquet integral of* f *with restricted domain* A *is defined as follows:*

$$(C)\int_A f d\mu = \int_0^\infty \mu(A \cap \{x | f(x) \geq r\})dr.$$

From the above definitions, it is obvious the next theorem.

Theorem 1. *Let* $(X, \mathcal{B})$ *be a measurable space, let* f *be a nonnegative measurable function on* X *and* $A \in \mathcal{B}$*. Then,*

(i) $(C)\int f d\mathrm{Pos}_A = \sup_{x\in A} f(x)$ *where* Pos_A *is the 0-1 possibility measure focused on* A.

(ii) $(C)\int f d\mathrm{Nec}_A = \inf_{x\in A} f(x)$ *where* Nec_A *is the 0-1 necessity measure focused on* A.

The next theorem is known as the subadditivity theorem.

Theorem 2. *[2,3] Let* $(X, \mathcal{B})$ *be a measurable space, let* f *and* g *be a nonnegative measurable function on* X *and* $A \in \mathcal{B}$. *If a fuzzy measure* μ *on* $\mathcal{B}$ *is submodular, then*

$$(C)\int (f+g)d\mu \leq (C)\int f d\mu + (C)\int g d\mu.$$

We need some additional notation. Let us denote by $||f||$ the following: $||f|| := (C)\int |f| d\mu$. Then, because of the subadditivity theorem, we know that if μ is submodular, we have $||f+g|| \leq ||f|| + ||g||$. Then $||\cdot||$ is a seminorm.

We will also use the following.

$$L_\mu^+(X) := \{f | f \text{is a nonnegative measurable function}, ||f|| < \infty\}.$$

3 Probabilistic metric spaces

In this section we review the concept of probabilistic metric spaces. To do so we begin with the concept of metric spaces and then we review the definitions of distance distribution functions and triangle functions.

3.1 Menger space

A metric space is defined in terms of a set S and a function $d : S \times S \to \mathbb{R}^+$ that plays the role of distance on the set S. Here, we understand $\mathbb{R}^+ = [0, \infty)$ and $\overline{\mathbb{R}^+} = [0, \infty]$.

Definition 7. *Let* $d : S \times S \to \mathbb{R}^+$, *then* d *is called a pseudometric on* S *if the following properties hold for* $a, b, c \in S$*:*

- $d(a, b) \geq 0$ *with equality if* $a = b$ *(positive property),*
- $d(a, b) = d(b, a)$ *(symmetry property), and*
- $d(a, b) \leq d(a, c) + d(c, b)$ *(triangle inequality property).*

Definition 8. *The pair* (S, d) *where* d *is a metric on* S *is called a pseudo metric space and* $d(a, b)$ *is the distance between* a *and* b.

A pseudo metric space (S,d) is a metric space if $d(a, b) = 0$ *implies* $a = b$.

The pair (S, d) *where* d *is a function* $S \times S \to \mathbb{R}^+$ *that satisfies the positive property and triangle inequality (but not the symmetry property) is a quasimetric space. The pair* (S, d) *where* d *is a function* $S \times S \to \mathbb{R}^+$ *that satisfies positive property and the symmetry property (but not the triangle inequality) is a semimetric space.*

Probabilistic metric spaces were introduced as a generalization of the concept of a metric. They replace the distance function in a metric space by a distance distribution function. So, the *distance* between a pair of elements in S is not a number but a distribution on these numbers. We introduce this concept below.

Definition 9. *[10] A nondecreasing function F defined on $\overline{\mathbb{R}^+}$ that satisfies (i) $F(0) = 0$; (ii) $F(\infty) = 1$, and (iii) that is left continuous on $(0, \infty)$ is a distance distribution function.*

Δ^+ denotes the set of all distance distribution functions.

In this definition we can understand $F(x)$ as the probability that the distance is less than or equal to x. In this way we can write any classical distance a in terms of a distance distribution function. More particularly, we will use in this case ϵ_a defined as follows. Naturally, ϵ_a is a step function at a.

Definition 10. *[10] (Def. 4.1.4) For any a in $\mathbb{R}^+$, we define $\epsilon_a \in \Delta^+$ by*

$$\epsilon_a(x) = \begin{cases} 0, & 0 \leq x \leq a \\ 1, & a < x \leq \infty \end{cases}$$

The next step towards the definition of a probabilistic metric space is to consider a counterpart of triangle inequality. Triangle functions will be used for this purpose. We review them below.

Definition 11. *[10] Let Δ^+ be the set of all distance distribution functions.*

A binary operation on Δ^+ is a triangle function if it is commutative, associative, and nondecreasing in each place, and has ϵ_0 as the identity.

It is important to underline the link between triangle functions and t-norms [1]. In particular, for a t-norm $\top$, we have $\tau_\top(F, G)(x) = \top(F(x), G(x))$ is a triangle function. See Def. 7.1.3 and Section 7.1 in [10]. The maximal triangle function is $\tau_{\min}$ (Theorem 7.1.4 in [10]).

We are now in conditions to define probabilistic metric spaces.

Definition 12. *[10] Let $(S, \mathcal{F}, \tau)$ be a triple where S is a nonempty set, $\mathcal{F}$ is a function from $S \times S$ into Δ^+, τ is a triangle function; then $(S, \mathcal{F}, \tau)$ is a probabilistic metric space (PM space) if the following conditions are satisfied for all p, q, and r in S:*

(i) $\mathcal{F}(p, p) = \epsilon_0$
(ii) $\mathcal{F}(p, q) \neq \epsilon_0$ *if* $p \neq q$
(iii) $\mathcal{F}(p, q) = \mathcal{F}(q, p)$
(iv) $\mathcal{F}(p, r) \geq \tau(\mathcal{F}(p, q), \mathcal{F}(q, r))$.

Given a probabilistic metric space $(S, \mathcal{F}, \tau)$, we say that $(S, \mathcal{F})$ is a probabilistic metric space under τ.

A probabilistic pseudometric space (PPM space) $(S, \mathcal{F}, \tau)$ is defined as above but not requiring condition (ii). When all conditions above apply but (iv) is not

required we have a probabilistic semimetric space. When all conditions apply but (iii) is not required we have a probabilistic quasimetric space.

We prefer to use F_{pq} instead of $\mathcal{F}(p,q)$. Then, we express the value of the latter at x simply as $F_{pq}(x)$.

We consider in this paper particular probabilistic metric spaces. The next definition introduces Menger spaces.

Definition 13. *[10] Let $(S,\mathcal{F},\tau)$ be a probabilistic metric space. Then $(S,\mathcal{F},\tau)$ is proper if*

$$\tau(\epsilon_a,\epsilon_b) \geq \epsilon_{a+b}$$

for all a,b in $\mathbb{R}^+$.

If $\tau=\tau_\top$ for some t-norm $\top$, then $(S,\mathcal{F},\tau)$ is a Menger space, or equivalently $(S,\mathcal{F})$ is a Menger space under $\top$.

Example 2. Let $(S,\mathcal{F},\tau)$ be a probabilistic metric space and $a,b\in\mathbb{R}^+$ with $a\geq b$.

- Suppose that τ is minimum, that is, $\tau=\wedge$.
  Since $a\wedge b<a+b$, we have $\tau(\epsilon_a,\epsilon_b)\geq\epsilon_{a+b}$. Therefore $(S,\mathcal{F},\tau)$ is a proper Menger space.
- Suppose that τ is algebraic product.
  Since $\epsilon_a\cdot\epsilon_b=\epsilon_a$, we have $\tau(\epsilon_a,\epsilon_b)\geq\epsilon_{a+b}$. Therefore $(S,\mathcal{F},\tau)$ is a proper Menger space.
- Suppose that τ is the bounded difference, that is, $\tau=W$.
  Since $0\vee(\epsilon_a+\epsilon_b-1)=\epsilon_a$, we have that $(S,\mathcal{F},\tau)$ is a proper Menger space.

3.2 E-spaces

This is a family of probabilistic metric spaces [11,12] that are constructed in terms of a set of functions and a probability space. For any pair of functions, and any x we can compute the measure of the points that are at a distance at most x. The definition of the E-spaces uses a probability to measure the set of points. As discussed by Schweizer and Sklar this can be seen as a generalization of just using the Lebesgue measure on the $I=[0,1]$ interval.

The definition of E-spaces follows. Here, $L_1^+(\Omega)$ is the set of all positive a.e. finite Lebesgue measurable functions on Ω.

Definition 14. *[9] Let $(\Omega,\mathcal{A},P)$ be a probability space, let (M,d) be a metric space, let S be a set of functions from Ω into M and let $\mathcal{F}$ be a mapping from $S\times S$ into Δ^+. Then, $(S,\mathcal{F})$ is an E-space with base $(\Omega,\mathcal{A},P)$ and target (M,d) if*

- *(i) for all p,q in S and all x in $\mathbb{R}^+$ the set*

  $$\{\omega\in\Omega | d(p(\omega),q(\omega))<x\}$$

  belongs to $\mathcal{A}$; i.e., the composite function $d(p,q)$ from Ω into $\mathbb{R}^+$ is P-measurable and therefore in $L_1^+(\Omega)$; and

– *(ii) for all p, q in S, $\mathcal{F}(p,q) = F_{pq}$ defined by*

$$F_{pq}(x) = P(\{\omega \in \Omega | d(p(\omega), q(\omega)) < x\}). \quad (1)$$

Equation 1 implies that $\mathcal{F}$ satisfies Properties (i) and (iii) in Definition 12. If $\mathcal{F}$ also satisfies Property (ii), then $(S, \mathcal{F})$ is a canonical E-space.

The following can be proven for E-spaces. The proof of this theorem is given in [9] and also in [12].

Theorem 3. *[9] Let $(S, \mathcal{F})$ be an E-space. Then $(S, \mathcal{F})$ is a probabilistic pseudometric space under τ_W. If $(S, \mathcal{F})$ is canonical, then it is a Menger space under W.*

3.3 F-space

We have introduced [7] a generalization of E-spaces by means of replacing the probability function in Equation 1 by a non-additive measure. This measure evaluates the set of ω that are at most at a given *distance* x. We provide the definition below.

Definition 15. *[7] Let $(\Omega, \mathcal{A})$ be a measurable space, and let μ a fuzzy measure on $(\Omega, \mathcal{A})$. Let (M, d) be a metric space, let S be a set of functions from Ω into M and let $\mathcal{F}$ be a mapping from $S \times S$ into Δ^+. Then, $(S, \mathcal{F})$ is an F-space with base $(\Omega, \mathcal{A}, \mu)$ and target (M, d) if*

– *(i) For all p, q in S and all x in $\mathbb{R}^+$ the set*

$$\{\omega \in \Omega | d(p(\omega), q(\omega)) < x\}$$

belongs to $\mathcal{A}$.

– *(ii) For all p, q in S, $\mathcal{F}(p,q) = F_{pq}$ with*

$$F^{\mu}_{pq}(x) = \mu(\{\omega \in \Omega | d(p(\omega), q(\omega)) < x\}). \quad (2)$$

In our previous paper we have proven the following two results.

Theorem 4. *[7] Let $(\Omega, \mathcal{A})$ be a measurable space, and let μ be a non-additive measure on $(\Omega, \mathcal{A})$ and $(S, \mathcal{F})$ be an F-space with base $(\Omega, \mathcal{A}, \mu)$.*

If μ is a supermodular non-additive measure on $(\Omega, \mathcal{A})$, then $(S, \mathcal{F})$ is a probabilistic pseudometric space under bounded difference τ_W.

The proposition below follows from this theorem and Proposition 1.

Proposition 2. *[7] Let $(\Omega, \mathcal{A})$ be a measurable space, and let P be a probability on $(\Omega, \mathcal{A})$, φ be as increasing convex function on closed interval on [0,1] with $\varphi(0) = 0, \varphi(1) = 1$, and $(S, \mathcal{F})$ be an F-space with base $(\Omega, \mathcal{A}, \varphi \circ P)$.*

Then, $(S, \mathcal{F})$ is a probabilistic pseudometric space under bounded difference τ_W.

4 F-space and Choquet integral

In this section we discuss F-spaces when we consider functions from $X \times \Omega \to \mathbb{R}^+$, for appropriate X and Ω. Then, we consider a distance between functions in terms of the Choquet integral. Let us start introducing the distance.

Definition 16. *Let $(X, \mathcal{B})$ be a measurable space and ν be a fuzzy measure on $\mathcal{B}$. Then, for $p, q \in L_\nu^+(X)$, the distance d_ν between p and q by means of the Choquet integral with respect to ν is defined by $d_\nu(p,q) = (C)\int |p-q| d\nu$.*

The next proposition is immediate from the previous definition and the subadditivity theorem.

Proposition 3. *Let ν be a fuzzy measure. Then, let d_ν be the distance defined by means of the Choquet integral with respect to ν as above. Then,*

(i) *$(L_\nu^+(X), d_\nu)$ is a symmetric space; and*
(ii) *if ν is submodular, then $(L_\nu^+(X), d_\nu)$ is a pseudometric space.*

4.1 Results

Let us now consider two measurable spaces $(\Omega, \mathcal{A})$ and $(X, \mathcal{B})$, and fuzzy measures μ on $\mathcal{A}$ and ν on $\mathcal{B}$. Suppose that X is finite and $\mathcal{B} = 2^X$.

Let $f : X \times \Omega \to \mathbb{R}^+$ be measurable for fixed $x \in X$ and fixed $\omega \in \Omega$. Then, the set of the above mentioned functions is denoted by M. Then, we denote by f_ω the function $f(x, \omega)$. For p, q in M, then $d_\nu(p_\omega, q_\omega)$ corresponds to $||p_\omega - q\omega||_\nu$ for $p_\omega, q_\omega \in L_\nu^+(X)$.

We can then prove the next proposition.

Proposition 4. *Let X be a finite set and ν be a fuzzy measure on 2^X, $L_\nu^+(X)$ be as above, and d_ν be the distance by means of the Choquet integral with respect to ν for functions in $L_\nu^+(X)$.*

Then, for a given $\omega \in \Omega$, and $p_\omega, q_\omega \in L_\nu^+(X)$, we have that $d_\nu(p_\omega, q_\omega)$ is $\mathcal{A}-$measurable.

The next proposition is obtained from Theorem 4.

Proposition 5. *Consider two measurable spaces $(\Omega, \mathcal{A})$ and $(X, \mathcal{B})$, and fuzzy measures μ on $\mathcal{A}$ and ν on $\mathcal{B}$. Let $M := \{f | f : X \times \Omega \to \mathbb{R}^+,$ both $\mathcal{A}, \mathcal{B}$ measurable$\}$.*

Let $S = L_\nu^+(X)$ and $p_\omega, q_\omega \in S$. Then, let

$$F_{pq}^\mu(x) = \mu(\{\omega \in \Omega | d_\nu(p_\omega, q_\omega) < x\}).$$

If μ is super modular and ν is submodular, then $(S, F_{p,q}^\mu)$ is a probabilistic pseudometric space under bounded difference τ_W.

Next we will consider the Choquet integral with respect to μ as the average of the distances on the $\omega \in \Omega$.

Define the μ-average $A_\mu(p,q)$ of $p_\omega, q\omega$ for $\omega \in \Omega$ by

$$A_\mu(p,q) = (C)\int d_\nu(p,q)d\mu \tag{3}$$

Note that here we can consider fuzzy measures ν that depend on $\omega \in \Omega$. That is, $\nu(\omega)$ and, thus, $d_{\nu(\omega)}(p_\omega, q_\omega)$. This can be used in Equation 3 as well.

The next proposition follows from the definition of the conjugate.

Proposition 6. *Let $p_\omega, q_\omega \in L^+_\nu(X)$ for $\omega \in \Omega$ and $s = \sup_{\omega\in\Omega}\{d_\nu(p(\omega), q(\omega)) < x\}$. Then,*

$$A_\mu(p,q) = s - \int_0^s F^\mu_{pq}(x)dx.$$

4.2 Example

In this section we consider an example. Let $\Omega := \{\omega_1, \omega_2, \omega_3, \omega_4\}$, $X = \{x_1, x_2\}$, $\mathcal{A} := 2^\Omega$, and $\mathcal{B} := 2^X$. Then $(\Omega, \mathcal{A})$ and $(X, \mathcal{B})$ are measurable spaces.

Let us define a function $f : X \times \Omega \to \mathbb{R}^+$ according to the following table.

	ω_1	ω_2	ω_3	ω_4
x_1	1	1	2	4
x_2	1	4	1	4

Now, let us define the following fuzzy measures ν_k for $k = 0, 1, 2, \infty$ defined on $(X, \mathcal{B})$ as the table below indicates.

	$\{x_1\}$	$\{x_2\}$	$\{x_1, x_2\}$
ν_0	0	0	1
ν_1	3/4	1/4	1
ν_2	$\sqrt{3}/2$	1/2	1
ν_∞	1	1	1

Now, let us define μ_a on $(\Omega, \mathcal{A})$ by $\mu_a := \lambda^a$ where $\lambda(\{\omega_k\}) = 1/4$ for $k = 1, 2, 3, 4$. That is, λ is additive.

For each ω_k for $k = 1, 2, 3, 4$, the distance by means of the Choquet integral with respect to μ_l denoted by $d_l(0, f)$ $l = 0, 1, 2, \infty$ corresponds to the table below.

Therefore for $l = 0, 1, 2, \infty$, we have $\{\omega | d_l(0, f(\omega)) < x\}$ as the tables below.

	ω_1	ω_2	ω_3	ω_4
d_0	1	1	1	4
d_1	1	7/4	7/4	4
d_2	1	5/2	$(2+\sqrt{3})/2$	4
d_∞	1	4	2	4

x	$0 \leq x \leq 1$	$1 < x \leq 4$	$x > 4$
$\{\omega \mid d_0(0, f(\omega)) < x\}$	$\emptyset$	$\{\omega_1, \omega_2, \omega_3\}$	Ω

x	$0 \leq x \leq 1$	$1 < x \leq 7/4$	$7/4 < x \leq 4$	$x > 4$
$\{\omega \mid d_1(0, f(\omega)) < x\}$	$\emptyset$	$\{\omega_1\}$	$\{\omega_1, \omega_2, \omega_3\}$	Ω

x	$0 \leq x \leq 1$	$1 < x \leq (2+\sqrt{3})/2$	$(2+\sqrt{3})/2 < x \leq 5/2$	$5/2 < x \leq 4$	$x > 4$
$\{\omega \mid d_2(0, f(\omega)) < x\}$	$\emptyset$	$\{\omega_1\}$	$\{\omega_1, \omega_3\}$	$\{\omega_1, \omega_2, \omega_3\}$	Ω

x	$0 \leq x \leq 1$	$1 < x \leq 2$	$2 < x \leq 4$	$x > 4$
$\{\omega \mid d_\infty(0, f(\omega)) < x\}$	$\emptyset$	$\{\omega_1\}$	$\{\omega_1, \omega_3\}$	Ω

Then, we have that the averages of the distances $A_l(0, f)$ for $l = 0, 1, 2, \infty$ are calculated using Equation 3 as follows.

(i) $A_0(0, f) = (4-1)\mu_a(\{\omega_4\}) + +(1-0)\mu(\Omega)$
$= 3 * (1/4)^a + 1$

(ii) $A_1(0, f) = (4 - 7/4)\mu_a(\{\omega_4\}) + (7/4 - 1)\mu_a(\{\omega_2, \omega_3, \omega_4\}) + (1-0)\mu(\Omega)$
$= (9/4) * (1/4)^a + (3/4) * (3/4)^a + 1$

(iii) $A_2(0, f) = (4-5/2)\mu_a(\{\omega_4\}) + (5/2 - (2+\sqrt{3})/2)\mu_a(\{\omega_2, \omega_4\}) + ((2+\sqrt{3})/2 - 1)\mu_a(\{\omega_2, \omega_3, \omega_4\}) + (1-0)\mu(\Omega)$
$= (3/2) * (1/4)^a + ((3-\sqrt{3})/2) * (1/2)^a + (\sqrt{3}/2) * (1/2)^a + 1$

(iv) $A_\infty(0, f) = (4-2)\mu_a(\{\omega_2, \omega_4\}) + (2-1)\mu_a(\{\omega_2, \omega_3, \omega_4\}) + (1-0)\mu(\Omega)$
$= 2 * (1/2)^a + 1 * (3/4)^a + 1$

Note the following, let $a \to 0$, then we have $A_l(0, f) = 4 = \max(d_l(0, f))$ and let $a \to \infty$, $A_l(0, f) = 1 = \min(d_l(0, f))$ for $l = 0, 1, 2, \infty$.

5 Conclusions

In this paper we have reviewed the definition of F-spaces and provided new results for these spaces. In particular, we have considered the case of functions p from $X \times \Omega$ into $\mathbb{R}$. We have also provided an example of their computation.

As future work we plan to provide new result in the same direction. One of our motivations to work on probabilistic metric spaces is to compare machine learning models built from databases. Some results in this direction appear in our previous paper [16]. Another future direction is to consider F-spaces in this setting.

Acknowledgements

This study was partially funded by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

1. Alsina, C., Frank, M. J., Schweizer, B. (2006) Associative Functions: Triangular Norms and Copulas, World Scientific.
2. Choquet, G. (1953/54) Theory of capacities, Ann. Inst. Fourier 5 131-295.
3. Denneberg, D. (1994). Non-Additive Measure and Integral, Theory and Decision Library (Vol. 27). Dordrecht: Kluwer Academic Publilshers.
4. Drossos, C. A. (1977) Stochastic Menger spaces and convergence in probability, Rev. Roumaine Math. Pures Appl. 22 1069-1076.
5. Grabisch, M. (2016) Set Functions, Games and Capacities in Decision Making, Springer.
6. Muštari, D. H., Šerstnev (1966) On methods of introducing a topology into random metric spaces, Izv. Vysh. Mat. 6:55 99-106. (in Russian)
7. Narukawa, Y., Taha, M., Torra, V. (2023) On the definition of probabilistic metric spaces by means of fuzzy measures, Fuzzy Sets and Systems, in press.
8. E. Pap, *Null-Additive set functions,* Kluwer Academic Publishers, Dordorecht, 1995.
9. Schweizer, B., Sklar, A. (1960) Statistical metric spaces, Pacific J. Math. 10 313-334.
10. Schweizer, B., Sklar, A. (1983) Probabilistic Metric Spaces, Elsevier-North-Holland.
11. Sherwood, H. (1969) On E-spaces and their relation to other classes of probabilistic metric spaces, J. London Math. Soc. 44 441-448.
12. Stevens, S. S. (1968) Metrically generated probabilistic metric spaces, Fund. Math. 61 259-269.
13. Torra, V., Narukawa, Y. (2007) Modeling decisions: information fusion and aggregation operators, Springer.
14. Torra, V., Narukawa, Y., Sugeno, M. (eds.) (2013) Non-additive measures: theory and applications, Springer.
15. Torra, V., Navarro-Arribas, G. (2018) Probabilistic metric spaces for privacy by design machine learning algorithms: modeling database changes, Proc. DPM 2018, LNCS 11025 422-430.
16. Torra, V., Taha, M., Navarro-Arribas, G. (2021) The space of models in machine learning: using Markov chains to model transitions. Prog. Artif. Intell. 10(3): 321-332.
17. Zadeh, L. A. (1978) Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems 1 3-28.

Three Point Comparison of Interval Priority Weight Estimation Methods in Alternative Ranking*

Masahiro Inuiguchi[1,2], Akiko Hayashi[1], and Shigeaki Innan[1,3]

[1] Osaka University, Osaka 560-8531, Japan
[2] inuiguti@sys.es.osaka-u.ac.jp
[3] innan@inulab.sys.es.osaka-u.ac.jp

Abstract. By estimating interval weights reflecting the vagueness of evaluation from a crisp pairwise comparison matrix, we may analyze the potential solutions and give a well-considered suggestion in multiple criteria decision problems. In this paper, we compare several interval priority weight estimation methods by numerical experiments for finding the most appropriate one as well as their characters. For comparison, the accuracy in ranking alternatives is adopted. To cope with the non-uniqueness of the solution to the interval priority weight estimation problem, we compare the accuracy at three representative solutions, i.e., the standard, minimum, and maximum solutions. We show the results under different numbers of criteria when maximin and maximax rules are used for ranking alternatives. Several characteristic tendencies of estimation methods are revealed.

Keywords: Interval analysis, AHP, maximin rule, minimax rule, ranking alternatives

1 Introduction

The analytic hierarchy process (AHP) [1] is one of the useful tools for multiple criteria decision problems. The problem is solved by using a crisp priority weight vector (PWV) estimated from a pairwise comparison matrix (PCM). To reflect the vagueness in decision maker's (DM's) evaluation, interval priority weight (PW) estimation has been investigated [2–4]. By estimating interval PWV, we may analyze the potential solutions and give a well-considered suggestion for making decisions. Then interval PW estimation methods have been proposed and investigated their usefullness and proporties.

To find the most appropriate one and their characters, various methods for estimating the interval PWV have been compared by numerical experiments [2–4]. It is shown that the estimation methods using the minimum ranges perform well in many cases. As multiple interval PWVs associate with a consistent interval PCM, any interval PW estimation problem usually has multiple solutions [5].

* Supported by JSPS KAKENHI Grant Number JP23K04272

Therefore, the comparisons of solution sets are more reasonable than those of one-point solutions in the evaluation of interval PW estimation methods.

In this paper, instead of the comparisons of solution sets, we compare three representative solutions to the interval PW estimation problems as we did earlier in [4]. The three representative solutions are (i) the standard solution, the solution such that the sum of centers of interval PWs is one, (ii) the minimum solution such that the sum of centers of interval PWs takes the minimum, and (iii) the maximum solution such that the sum of centers of interval PWs takes the maximum. We compare the accuracy in ranking alternatives, i.e., the number of alternative pairs whose pairwise comparisons by the estimated interval PWs produce the correct results. For ranking alternatives, we use the maximin rule and the maximax rule because the total scores of alternatives become intervals under an interval PWV. In [4], the following facts are demonstrated:

a) In difficult ranking problems such that many alternatives take similar total utility scores, the accuracies in the maximin rule tends to be larger than those in the maximax rule.
b) In easy ranking problems, interval PW estimation methods often perform worse than the classical crisp PW estimation methods under the maximum true interval PWs in the maximin rule and also under the minimum true interval PWs in the maximax rule.
c) In difficult ranking problems, accuracies of estimated interval PWs are larger than those of estimated crisp weights. Especially, the estimation method by maximizing the minimum ranges with minimizing the sum of widths often perform best.
d) Accuracies of the estimation method by averaging the minimum ranges with minimizing the sum of deviations are larger than those of estimated crisp PWs in all problem settings.

However, the differences among the settings of true interval PWs have not yet been shown, and the comparisons in different numbers of criteria have not yet been done. In this paper, those results are shown and investigated the characteristic tendencies of interval PW estimation methods.

In the next section, the conventional interval PW estimation method is reviewed. The other estimation methods, representative solutions, and methods for ranking alternatives are described in Section 3. In Section 4, numerical experiments are explained. The results of numerical experiments are shown in Section 5. In Section 6, the conclusion is given.

2 The Conventional Interval AHP

Assume that the DM's PW of each item X_i is evaluated vaguely and represented by interval PWs $W_i = [w_i^{\mathrm{L}}, w_i^{\mathrm{U}}]$, $i \in N = \{1, 2, \ldots, n\}$. Then the (i, j)-component a_{ij} of the given PCM A satisfies $a_{ij} \in [w_i^{\mathrm{L}}/w_j^{\mathrm{U}}, w_i^{\mathrm{U}}/w_j^{\mathrm{L}}]$, $i, j \in N$, $i \neq j$. The interval PW estimation problem is formulated as the following linear

programming (LP) problem [6]:

$$\begin{aligned} \text{minimize} \ & \sum_{i\in N}(w_i^{\mathrm{U}} - w_i^{\mathrm{L}}), \\ \text{sub. to} \ & a_{ij}w_j^{\mathrm{L}} \le w_i^{\mathrm{U}}, \ i,j \in N \ i \ne j, \\ & \sum_{i\in N\setminus j} w_i^{\mathrm{U}} + w_j^{\mathrm{L}} \ge 1, \ \sum_{i\in N\setminus j} w_i^{\mathrm{L}} + w_j^{\mathrm{U}} \le 1, \ j \in N, \\ & w_i^{\mathrm{U}} \ge w_i^{\mathrm{L}} \ge \epsilon, \ i \in N. \end{aligned} \tag{1}$$

We note that the second and third constraints imply the normality condition of interval weight vector composed of W_i, $i \in N$. The first constraints are equivalent to $w_j^{\mathrm{L}} \le a_{ji}w_i^{\mathrm{U}}$, $i,j \in N$, $i \ne j$ from the reciprocity of PCM A as well as to $a_{ij} \in [w_i^{\mathrm{L}}/w_j^{\mathrm{U}}, w_i^{\mathrm{U}}/w_j^{\mathrm{L}}]$, $i,j \in N$, $i \ne j$. As the objective function is the sum of the widths of interval weights, this estimation problem is called "the minimization problem of the sum of widths (MSW)".

3 Non-uniqueness, Methods and Ranking Alternatives

There are usually multiple feasible interval PWVs whose deviations from the given PCM A are the same. Namely, the solution to the estimation problem is usually non-unique [5].

To cope with the non-uniqueness, the most natural one, i.e., the standard solution such that the sum of centers of interval PWs is one, is used as the representative solution. We note that the existence of the standard solution is guaranteed by the normality constraints of interval PWs. The optimal solution set of the estimation problem usually forms a line segment from the optimal solution minimizing the sum of centers to the optimal solution maximizing the sum of centers. In this paper, we extend the accuracy analysis by employing the minimum and maximum solutions located at the edges of the line segment of solutions.

Given a standard solution $W_i = [w_i^{\mathrm{L}}, w_i^{\mathrm{U}}]$, $i \in N$, the minimum and maximum solutions are obtained as $t^{\mathrm{L}}W_i$, $i \in N$ and $t^{\mathrm{U}}W_i$, $i \in N$, respectively, where t^{L} and t^{R} are calculated by

$$t^{\mathrm{L}} = \max_{i\in N} \frac{1}{w_i^{\mathrm{L}} + \sum_{j\in N\setminus i} w_j^{\mathrm{U}}}, \ t^{\mathrm{U}} = \min_{i\in N} \frac{1}{w_i^{\mathrm{U}} + \sum_{j\in N\setminus i} w_j^{\mathrm{L}}}. \tag{2}$$

The interval PWV estimated by Problem (1) do not reflect appropriately the vagueness of the DM's evaluation [2, 3]. Then, various interval PW estimation methods have been proposed by authors [2–4]. To reflect the vagueness appropriately to the solution, the variety of potential solutions is considered by introducing the minimum ranges of interval PWs. As one of the various estimation methods using minimum ranges, we describe the method of averaging the minimal ranges with deviations, $\mathrm{AMR_D}$. The method for obtaining the standard solution is composed of the following four steps:

⟨1⟩ For each $k \in N$, obtain the optimal value $\hat{d}_{\bar{k}}$ of LP problem,

$$\begin{aligned} \text{minimize } & d_{\bar{k}} = \sum_{i \in N \setminus k} \sum_{j \in N \setminus k, i} d_{ij}, \\ \text{sub. to } & \sqrt{a_{ij}} w_j^{\mathrm{L}} + d_{ij} = \sqrt{a_{ji}} w_i^{\mathrm{U}},\ i, j \in N,\ i \neq j \\ & \sum_{i \in N \setminus j} w_i^{\mathrm{U}} + w_j^{\mathrm{L}} \geq 1,\ \sum_{i \in N \setminus j} w_i^{\mathrm{L}} + w_j^{\mathrm{U}} \leq 1,\ j \in N, \\ & \sum_{i \in N} (w_i^{\mathrm{L}} + w_i^{\mathrm{U}}) = 2,\ w_i^{\mathrm{U}} \geq w_i^{\mathrm{L}} \geq \epsilon,\ d_{ij} \geq 0,\ i, j \in N,\ i \neq j. \end{aligned} \tag{3}$$

⟨2⟩ For each $k \in N$, obtain the optimal value $\check{d}_{\bar{k}}$ of LP problem,

$$\begin{aligned} \text{minimize } & \tilde{d}_{\bar{k}} = \sum_{j \in N \setminus k} (d_{kj} + d_{jk}), \\ \text{sub. to } & \text{constraints of (3)}, d_{\bar{k}} = \hat{d}_{\bar{k}}. \end{aligned} \tag{4}$$

⟨3⟩ For each $k \in N$, obtain optimal solutions $\hat{W}_i(k),\ i \in N$ and $\check{W}_i(k),\ i \in N$ to the following two LP problems, respectively:

$$\begin{aligned} \text{maximize } & w_k^{\mathrm{U}} \ / \ \text{minimize } \ w_k^{\mathrm{L}}, \\ \text{sub. to } & \text{constraints of (3)}, d_{\bar{k}} = \hat{d}_{\bar{k}}, \tilde{d}_{\bar{k}} = \check{d}_{\bar{k}}. \end{aligned} \tag{5}$$

⟨4⟩ Then the interval PWs are obtained as

$$W_i = \frac{1}{2n} \sum_{k \in N} (\hat{W}_i(k) + \check{W}_i(k)),\ i \in N. \tag{6}$$

By replacing the averaging step ⟨4⟩ with a step taking the union of all normalized interval PWVs and dividing it by its sum of center values, we obtain the method of maximizing the minimal range with deviations, $\mathrm{MMR_D}$. Moreover, by replacing the objective functions showing deviations of LP problems solved at Steps ⟨1⟩ and ⟨2⟩ with the objective function showing the sum of widths, we obtain the method of averaging the minimal ranges with widths $\mathrm{AMR_W}$ and the method of maximizing the minimal range with widths $\mathrm{MMR_W}$. Furthermore, determining the center values by the classical methods for estimating a crisp PWV, e.g., eigenvalue method (EM) and geometric mean method (GM) in interval PW estimation methods, we obtain other methods.

Given marginal utility score $u_i(o_p)$ of Alternative o_p in view of the i-th criterion, the total utility score is obtained as an interval bounded by the following minimum total utility score $u^{\min}(o_p)$ and maximum total utility score $u^{\max}(o_p)$ under interval PWs $W_i = [w_i^{\mathrm{L}}, w_i^{\mathrm{U}}],\ i \in N$;

$$u^{\min}(o_p) = \min \left\{ \sum_{i \in N} u_i(o_p) w_i \;\middle|\; \sum_{i \in N} w_i = 1,\ w_i^{\mathrm{L}} \leq w_i \leq w_i^{\mathrm{U}},\ i \in N \right\}, \tag{7}$$

$$u^{\max}(o_p) = \max \left\{ \sum_{i \in N} u_i(o_p) w_i \;\middle|\; \sum_{i \in N} w_i = 1,\ w_i^{\mathrm{L}} \leq w_i \leq w_i^{\mathrm{U}},\ i \in N \right\}. \tag{8}$$

In this paper, for ranking alternatives, we adopt the maximin rule and the maximax rule. The maximin rule arranges alternatives in descending order of the minimum total utility score while the maximax rule arranges alternatives in descending order of the maximum total utility score.

4 Numerical Experiment

As described earlier, the solution to the interval PW estimation problem is not unique. Accordingly, even if we assume a true normalized interval PWV, we would not be able to estimate it by a unique normalized interval PWV. Moreover, we do not know whether the DM has a true normalized interval PWV or a true consistent interval PCM in her/his mind, where consistent interval PCM implies an interval PCM whose interval components $[a_{ij}^{\mathrm{L}}, a_{ij}^{\mathrm{U}}]$, are expressed as $a_{ij}^{\mathrm{L}} = w_i^{\mathrm{L}}/w_j^{\mathrm{U}}$ and $a_{ij}^{\mathrm{L}} = w_i^{\mathrm{U}}/w_j^{\mathrm{L}}$, $i, j \in N$, $i \neq j$ with a normalized interval PWV $\boldsymbol{W} = (W_1, W_2, ..., W_n)^{\mathrm{T}}$, $W_i = [w_i^{\mathrm{L}}, w_i^{\mathrm{U}}]$, $i \in N$. In the numerical experiment, we assume that the DM has a unique true consistent interval PCM in her/his mind. We prepare five different settings of consistent interval PCMs for each number n of criteria ($n = 4, 5, ..., 8$). Each consistent interval PCM can be represented by a normalized interval PWV with normalized center values. The normalized interval PWV with normalized center values associated with the five different settings of true consistent interval PCMs are given in Table 1 when $n = 6$. As shown in Table 1, setting A is a case where the widths increase as the center values of interval PWs decrease, setting B is a case where the widths decrease as the center values of interval PWs decrease, setting C is a case where the widths decrease as the center values of interval PWs approach to their median, setting D is a case where the widths are constant regardless of the center values of interval PWs, and finally, setting E is a case where the widths increase as the center values of interval PWs approach their median. Those properties of five settings A to E are the same in other number of criteria, $n = 4, 5, ..., 8$. Therefore, the true consistent interval PCMs are denoted by a combination of the number of criteria and the alphabet showing the setting. For example, 5E stands for the true consistent interval PCMs of setting E with five criteria. We use 1,000 PCMs randomly generated from a true consistent interval PCMs for each of those twenty-five settings, 4A, 4B,..., 8E. Each component a_{ij}, $i < j$ of a PCM is generated by $\exp(rand)$ with a random number $rand$ obeying a uniform distribution $[\log(a_{ij}^{\mathrm{L}}), \log(a_{ij}^{\mathrm{U}})]$ of the true consistent interval PCM. The components a_{ij}, $i > j$ and a_{ii} of the PCM is determined by $a_{ij} = 1/a_{ji}$ and 1, respectively.

The high accuracy in ranking alternatives would be more significant than the high accuracy of interval PW estimation in the setting of multiple criteria decision making problems. From this point of view, we would like to compare the results in ranking alternatives based on solutions to the various estimation problems to the results in ranking alternatives based on all normalized interval PWVs obtained from the true consistent interval PCM. Unfortunately, this comparison becomes complex because we have infinitely many solutions and infinitely many

Table 1. Five kinds of normalized interval PWVs ($n = 6$)

	A	B	C	D	E
T_1	[0.21, 0.25]	[0.16, 0.30]	[0.16, 0.30]	[0.18, 0.28]	[0.20, 0.26]
T_2	[0.17, 0.23]	[0.14, 0.26]	[0.15, 0.25]	[0.15, 0.25]	[0.15, 0.25]
T_3	[0.14, 0.24]	[0.13, 0.23]	[0.15, 0.21]	[0.13, 0.23]	[0.11, 0.25]
T_4	[0.10, 0.20]	[0.11, 0.19]	[0.12, 0.18]	[0.10, 0.20]	[0.08, 0.22]
T_5	[0.07, 0.19]	[0.10, 0.16]	[0.08, 0.18]	[0.08, 0.18]	[0.08, 0.18]
T_6	[0.04, 0.18]	[0.09, 0.13]	[0.04, 0.18]	[0.06, 0.16]	[0.08, 0.14]

normalized interval PWVs. Then we select the three representative normalized interval PWVs, i.e., standard, minimum and maximum ones from the set of normalized interval PWVs. We compare the standard, minimum and maximum solutions of the estimation problem to the standard, minimum, and maximum normalized interval PWVs corresponding to the true consistent interval PCM, respectively. The accuracy of the estimated ranking alternatives is defined by the number of alternative pairs whose order is correctly estimated. For ranking alternatives, we use the maximin rule and the maximax rule. In those rules. the order between two alternatives is independent from any irrelevant alternatives.

For the numerical experiment, we consider multiple criteria decision problems with five alternatives under twenty-five settings, 4A, 4B,..., 8E. We generate two kinds of ranking problems, i.e., easy and difficult ones, in each setting. Each problem is defined by the marginal utility scores of five alternatives. In the 'easy ranking problems', the marginal scores of an alternative are generated randomly so that their sum becomes 1. By doing this five times, we obtain a set of five alternatives, that corresponds to a problem. We generate 100 easy ranking problems in each setting. In the 'difficult ranking problems', we generate five alternatives having similar total utility scores for a problem. The marginal scores of an alternative are generated by the following three steps: (i) Generate $g_i \geq 0$, $i \in N$ randomly such that $\sum_{i \in N} g_i = 1$, (ii) Calculate $g'_i = 2g_i/(t_i^{\mathrm{L}} + t_i^{\mathrm{U}})$, $i \in N$. (iii) Determine marginal utilities by $u_i = g'_i \times rand_i$, $i \in N$ with random numbers $rand_i \sim U(0.95, 1.05)$, $i \in N$, where $T_i = [t_i^{\mathrm{L}}, t_i^{\mathrm{U}}]$ is the i-th component of the normalized interval PWV corresponding to a true consistent interval PCM, and $U(0.95, 1.05)$ stands for a uniform distribution over the interval $[0.95, 1.05]$. By this procedure, the total utility score of each alternative locates around 1. Generating five alternatives, we obtain a difficult ranking problem. We generate 100 difficult ranking problems in each setting.

5 Results

We applied twenty-seven interval PWCV estimation methods and two classical crisp PW estimation methods to each of the prepared PCMs ($25 \times 1,000$ PCMs). Due to limitations of space, we show the results of two interval PW estimation methods, i.e., $\mathrm{MMR_W}$ and $\mathrm{AMR_W}$ as well as a classical crisp PW estimation method, i.e., EV. $\mathrm{MMR_W}$ and $\mathrm{AMR_W}$ are top two methods. The performances of EV and GM are similar and EV is selected for comparing to the interval PW estimation methods. In each setting, we examined 100 problems for each of 1,000

Table 2. Accuracies of the estimated ranking alternatives (1)

Set.	Easy ranking problem: Maximin rule								
	minimum			standard			maximum		
	EV	MMR_W	AMR_W	EV	MMR_W	AMR_W	EV	MMR_W	AMR_W
4A	852,282	858,864	876,859	858,212	876,348	881,507	877,856	884,993	889,614
4B	857,996	842,764	859,070	861,732	861,352	863,486	873,980	864,189	866,606
4C	853,135	839,654	862,756	867,371	859,125	870,851	869,007	872,670	874,157
4D	853,970	858,897	867,864	860,952	874,358	874,796	882,220	879,362	880,706
4E	854,835	848,103	866,855	858,773	863,145	868,521	870,749	870,381	875,530
5A	836,132	848,871	885,168	856,640	863,001	895,823	876,408	886,194	902,218
5B	856,350	826,301	867,104	866,076	848,100	874,597	886,338	869,752	879,251
5C	829,665	835,211	861,543	857,553	847,613	882,042	874,039	868,749	885,101
5D	847,313	846,554	877,003	867,603	858,442	888,014	890,399	876,708	890,663
5E	857,835	835,930	883,996	865,023	864,080	888,125	880,189	885,775	892,451
6A	772,543	773,819	843,328	812,785	799,984	867,221	845,791	842,898	876,508
6B	777,868	735,861	796,346	779,932	776,929	797,731	822,326	792,523	801,042
6C	744,416	759,682	789,649	795,744	790,311	830,605	823,786	815,623	838,977
6D	777,687	782,059	822,987	803,370	801,941	842,737	843,760	822,221	848,190
6E	769,224	749,754	819,168	793,092	798,071	845,545	818,321	826,142	844,364
7A	678,947	831,783	821,490	756,041	802,113	849,113	815,895	811,907	849,862
7B	697,520	747,218	782,606	702,141	774,796	766,769	769,715	733,539	739,995
7C	703,491	777,268	803,726	746,607	785,218	812,611	800,873	782,670	808,016
7D	691,502	799,428	800,226	739,040	786,838	808,787	803,661	766,306	795,183
7E	671,045	786,823	794,418	725,195	791,014	813,500	788,080	773,554	806,010
8A	664,908	839,781	828,897	743,664	816,890	859,056	803,772	810,001	848,582
8B	690,605	733,712	785,877	680,731	785,965	773,366	737,707	725,182	715,812
8C	671,217	770,669	785,956	720,509	789,586	800,878	777,159	745,908	776,627
8D	685,176	792,004	803,264	715,108	799,118	815,176	783,960	746,764	778,635
8E	660,040	780,654	793,991	705,560	798,779	823,659	753,107	774,252	800,189

PCMs. Accordingly, we have 100,000 kinds of problems in each setting. For each problem, we have 10 alternative pairs. Therefore, the maximum number of the correctly ordered pairs becomes $1,000,000$ in each setting.

The results are shown in Tables 2–5. Tables 2 and 3 show the accuracies in ranking alternatives by the estimated interval PWV when the maximin rule is adopted under easy ranking problems and under difficult ranking problems, respectively. On the other hand, Tables 4 and 5 show the accuracies in ranking alternatives by the estimated interval PWV when the maximax rule is adopted under easy ranking problems and under difficult ranking problems, respectively.

From those tables, we observe that the accuracies in easy ranking problems are less than those in difficult ranking problems. In easy ranking problems, the accuracies of EV in the ranking by maximin rule increases in the order corresponding to minimum, standard, and maximum solutions while those of EV in the ranking by maximax rule decreases in that order. As the minimum and maximum total utility scores increase in the order corresponding to minimum, standard, and maximum solutions. The maximum total utility score of the minimum solution and the minimum total utility score of the maximum solution approach to the center value of all conceivable total utility scores. From these

Table 3. Accuracies of the estimated ranking alternatives (2)

Set.	Difficult ranking problem: Maximin rule								
	minimum			standard			maximum		
	EV	MMR_W	AMR_W	EV	MMR_W	AMR_W	EV	MMR_W	AMR_W
4A	584,483	769,915	717,395	595,159	765,270	721,733	608,861	756,899	721,210
4B	542,330	694,414	628,632	555,066	708,109	636,977	565,816	659,693	623,281
4C	577,733	752,875	693,871	592,721	749,023	691,995	607,263	734,615	700,404
4D	557,758	763,615	680,618	570,722	759,991	684,920	587,332	732,857	680,812
4E	555,905	716,651	659,760	554,741	710,877	642,283	561,999	680,158	636,640
5A	599,506	809,320	767,476	636,624	808,570	790,859	660,917	799,804	798,108
5B	549,706	694,259	664,979	549,174	760,742	668,959	565,229	709,048	653,774
5C	604,943	782,658	744,531	635,759	798,685	764,716	658,737	772,461	768,155
5D	574,604	793,455	738,924	589,072	796,187	737,682	611,276	751,088	729,074
5E	552,347	730,586	708,966	565,591	774,161	709,176	581,508	736,455	699,897
6A	539,567	776,812	736,412	599,117	782,593	783,686	616,457	754,386	779,994
6B	522,855	703,219	654,368	525,225	759,264	644,817	541,909	611,390	551,884
6C	538,392	762,722	709,705	585,207	780,194	742,211	618,182	723,210	729,078
6D	533,651	812,875	726,601	544,121	816,293	731,944	571,619	710,231	704,951
6E	521,793	739,911	693,503	543,265	781,095	717,983	560,036	713,327	701,932
7A	519,211	835,027	769,125	573,847	822,082	783,067	616,247	750,565	759,080
7B	523,389	776,790	728,714	516,145	838,102	705,111	547,936	687,568	598,763
7C	529,633	806,386	750,013	567,323	803,306	744,456	603,743	708,597	700,793
7D	525,000	837,801	756,610	535,546	841,131	739,258	573,879	702,172	673,799
7E	512,270	812,079	748,578	528,234	842,226	745,006	561,001	709,825	692,621
8A	513,153	874,015	793,036	582,481	811,696	805,593	629,568	749,049	788,175
8B	514,530	777,048	747,898	514,818	849,724	721,268	535,544	673,323	585,378
8C	525,251	821,699	768,142	568,063	810,331	762,055	621,209	704,471	711,987
8D	522,710	869,887	780,250	531,976	845,880	756,083	570,312	680,212	675,057
8E	505,044	822,180	765,016	527,172	843,841	762,749	560,128	713,415	700,069

observations, we guess that ranking alternatives based on the crisp PWV estimated by EV may work well around the center value of all conceivable total utility scores. The accuracies of EV in the ranking by the maiximax rule with minimum solutions attain the best in several settings.

In difficult ranking problems, the accuracies in the ranking alternatives by the minimax rule are a little larger than those in ranking alternatives by the maximax rule. To generate difficult ranking problems, we make the centers of total utility intervals close to one another. Then the total utility intervals of five alternatives can often be nested. From this conceivable fact, the order of alternatives obtained by the minimax rule would be similar to the reverse order of that obtained by the maximax rule, and vice versa. Therefore, if the accuracies of EV, or more generally, a crisp priority estimation method, in the ranking alternatives by the maximin rule is large, those in the ranking alternatives by the maximax rule becomes small, and if the accuracies of EV, or more generally, a crisp priority estimation method, in the ranking alternatives by the maximax rule is large, those in the ranking alternatives by the maximin rule is small. Indeed, the accuracy of EV in the maximin rule plus the accuracy of EV in the maximax rule becomes near 1,000,000. As the orders of two alternatives can

Table 4. Accuracies of the estimated ranking alternatives (3)

Set.	Easy ranking problem: Maximax rule								
	minimum			standard			maximum		
	EV	MMR_W	AMR_W	EV	MMR_W	AMR_W	EV	MMR_W	AMR_W
4A	862,604	874,384	874,815	855,796	874,194	877,426	855,518	875,782	878,433
4B	882,070	856,707	869,101	865,716	855,930	866,418	864,788	853,948	867,786
4C	870,997	851,915	849,269	866,981	853,279	862,621	865,615	860,465	872,516
4D	885,362	870,465	875,429	860,040	871,505	871,875	860,966	872,477	874,058
4E	857,139	842,641	862,359	846,023	852,733	858,743	849,803	855,676	861,726
5A	878,528	862,054	889,557	860,290	867,432	891,801	850,890	863,775	894,737
5B	893,792	833,509	873,012	880,474	833,787	876,224	862,036	836,101	870,868
5C	864,729	840,397	854,849	856,158	847,096	871,127	843,730	855,945	876,879
5D	893,024	846,914	877,767	876,318	851,442	886,190	855,093	858,466	881,350
5E	882,675	824,034	880,764	870,231	838,603	883,273	861,647	850,291	882,963
6A	809,809	782,358	826,410	785,434	808,566	846,170	781,016	826,470	859,753
6B	841,656	670,741	793,772	818,934	701,521	804,194	771,266	765,194	800,720
6C	795,506	707,597	769,672	780,982	762,413	806,394	763,324	820,517	833,629
6D	842,003	745,058	810,753	792,707	783,733	829,720	765,947	821,373	832,664
6E	790,075	674,321	783,007	772,069	733,985	806,570	762,283	795,885	826,864
7A	793,431	753,292	797,178	705,191	810,697	820,603	714,339	821,351	841,627
7B	793,300	583,720	733,501	738,844	675,799	762,426	675,804	787,323	779,764
7C	794,397	654,916	763,979	732,663	739,548	799,328	711,083	803,398	822,152
7D	804,963	667,469	761,251	727,888	752,790	799,094	700,236	802,331	806,905
7E	760,342	635,234	733,396	713,052	729,818	779,383	688,368	800,823	813,073
8A	770,128	767,111	781,164	672,608	826,645	808,107	698,694	830,996	849,510
8B	773,879	558,833	722,441	721,174	659,294	755,844	653,232	809,088	791,043
8C	745,815	608,987	715,970	705,743	713,772	769,218	689,385	810,404	817,502
8D	786,581	650,372	737,464	695,514	756,356	785,671	668,545	830,683	814,227
8E	718,923	612,859	700,274	661,157	746,596	763,352	657,903	819,330	815,177

be the same in both the maximin rule and the maximax rule, the sums do not exactly equal to 1,000,000. Then the advantage of interval priority estimation can be observed because it potentially enlarges the accuracy in the maximin rule and the accuracy in the maximax rule at the same time. Indeed, accuracies of MMR_W and AMR_W in both maximin rule and maximax rule take values more than 650,000 on average. MMR_W performs better than AMR_W.

It is a little surprising that accuracies in the maximin rule are always larger than those in the maximax rule. However, this fact may come from the generation method for marginal utility scores. In our method, the marginal utility scores are generated by dividing random numbers by center values of interval PWs corresponding to the true consistent interval PCM. Therefore, the marginal score corresponding to a large interval PW tends to be small. To evaluate the minimum total utility score, the largest value in the interval PW is assigned in order of increasing marginal utility, as far as the sum of weights is not larger than one. On the contrary, to evaluate the maximum total utility score, the largest value in the interval PW is assigned in order of decreasing marginal utility, as far as the sum of weights is not larger than one. From these facts, the PWs assigned for calculating the minimum total utility score can be closer to the PWs estimated

Table 5. Accuracies of the estimated ranking alternatives (4)

Set.	Difficult ranking problem: Maximax rule								
	minimum			standard			maximum		
	EV	MMR_W	AMR_W	EV	MMR_W	AMR_W	EV	MMR_W	AMR_W
4A	456,689	720,511	664,845	468,563	736,540	669,066	487,121	746,656	671,237
4B	557,004	636,150	623,858	541,784	689,569	643,268	528,804	706,493	656,926
4C	461,873	700,783	634,935	467,647	735,967	650,827	482,261	738,941	663,967
4D	503,226	728,127	673,867	500,996	744,924	671,901	506,970	749,486	668,396
4E	529,465	622,983	633,789	527,021	680,559	649,814	527,131	704,636	656,509
5A	377,937	781,314	675,512	397,822	806,113	685,563	449,526	799,639	701,111
5B	541,895	690,820	673,638	512,831	743,624	683,342	506,345	786,026	705,840
5C	383,618	750,488	643,460	398,260	798,325	666,021	439,241	806,406	684,494
5D	457,202	793,052	714,422	456,497	817,535	717,700	486,347	802,644	710,314
5E	487,100	676,050	673,874	496,562	735,446	693,337	508,812	779,587	713,891
6A	423,783	720,341	676,135	443,308	771,402	700,574	500,254	802,715	725,830
6B	568,005	509,290	536,349	540,713	665,914	630,984	526,673	791,528	703,109
6C	429,863	668,109	642,114	453,403	754,423	686,999	497,907	818,036	720,793
6D	507,161	761,055	707,695	499,672	812,862	721,358	502,870	848,015	720,227
6E	493,690	599,017	620,192	500,166	710,449	676,673	515,800	794,604	718,036
7A	434,535	744,030	689,382	466,709	826,467	743,445	517,807	860,342	771,712
7B	562,766	606,756	614,043	527,436	765,023	713,321	516,764	852,278	747,027
7C	460,169	686,646	667,063	480,531	805,450	739,525	514,285	857,120	761,582
7D	507,375	765,923	706,858	504,751	838,468	749,891	518,355	870,931	757,164
7E	500,258	657,460	644,643	507,220	787,321	725,002	516,473	864,341	764,653
8A	406,392	791,986	685,542	446,706	847,017	749,538	512,300	866,892	811,107
8B	565,082	516,820	563,957	531,829	738,015	710,784	509,615	878,990	789,218
8C	443,169	691,090	643,046	463,766	811,858	734,195	507,026	875,327	783,696
8D	503,438	800,275	700,598	501,369	856,926	761,642	505,559	898,958	793,646
8E	493,114	659,610	628,658	503,163	788,238	730,822	518,021	880,059	80,5406

by EM. Therefore, the accuracies of EM in the maximin rule can be larger than those in the maximax rule.

To see the case of estimated interval PWVs, we should consider the following tendency [7] of the estimated interval PWs: the larger interval PW, the smaller its width. The error of the estimated interval PW corresponding to a larger marginal utility score makes the error of the total utility score bigger. The error of the estimated interval PW corresponding to a larger marginal utility score could be large because of the tendency of the estimated interval PWV as well as the property of the marginal utility score generation in difficult ranking problem. From the calculations of minimum and maximum total utility scores, the maximum total utility score would include bigger errors than the minimum total utility score. Therefore, the accuracies in the maximax rule are smaller than those in the maximin rule in difficult ranking problems.

The accuracies of MMR_W and AMR_W become better than those of EV as the number of criteria increases even in easy ranking problems. In difficult ranking problems, the accuracies of MMR_W and AMR_W are better than those of EV in all settings. In general, the estimated interval priority weight vectors perform better than the estimated crisp PWV.

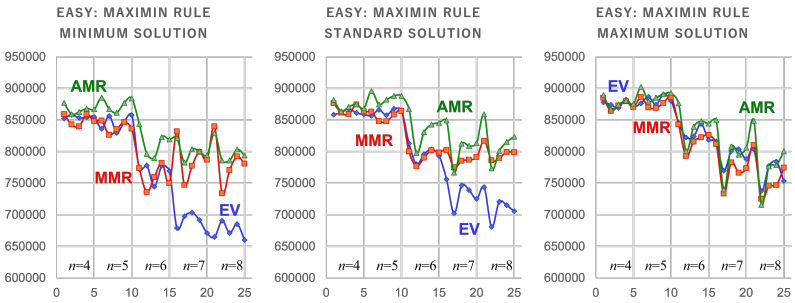


Fig. 1. Easy ranking problem: Maximin rule

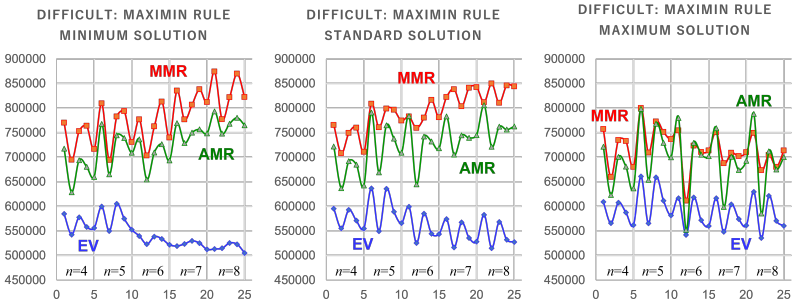


Fig. 2. Difficult ranking problem: Maximin rule

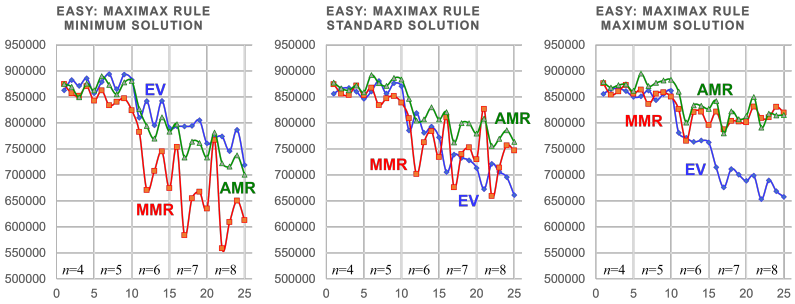


Fig. 3. Easy ranking problem: Maximax rule

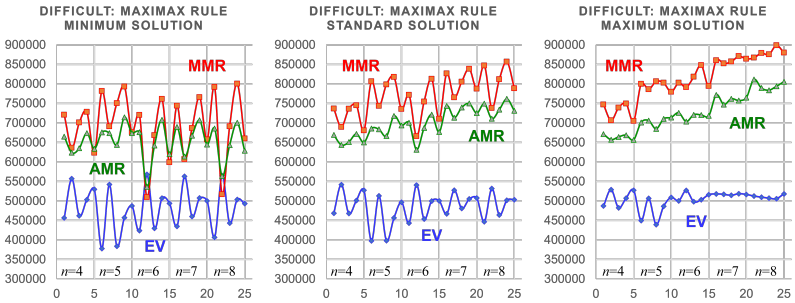


Fig. 4. Difficult ranking problem: Maximax rule

To see the differences in accuracies by the number of criteria, we depict graphs showing the relation between accuracy and the number of criteria in Figures 1–4. In those figures, the vertical axis shows the accuracy while the horizontal axis shows the settings in the order corresponding to 4A,4B,..., 8E. The interval PW estimation methods estimate $2n$ parameters from ${}_nC_2 = n(n-1)/2$ data under n criteria. When $n = 4$, the number of data is smaller than the number of parameters. When $n = 5$, the number of data equals the number of parameters. When $n \geq 6$, the number of data is larger than the number of parameters. Therefore, we may guess that the estimation problem becomes easier as n increases. However, as shown in Figures 1 and 3 corresponding to easy ranking problems, the accuracies decrease as n increases. This may imply that the correctness of the parameter estimation is required much more as n increases because the normality is assumed for the PWV. On the other hand, in Figures 2 and 4 corresponding to difficult ranking problems, the accuracies increase as n increases. Although the reason for this result is hard to explain, this may come from the fact that the correctness of only a few key parameter estimation would be required in difficult ranking problems and multiple data can work well.

Finally, from Figures 1–4, we find that $\mathrm{MMR_W}$ performs well in difficult ranking problems although the fluctuation by the setting is large. On the other hand, $\mathrm{AMR_W}$ frequently performs best in easy ranking problems.

6 Conclusion

We compared several interval PW estimation methods by numerical experiments, using three representative solutions to cope with the non-uniqueness of the solution. Characters of the top two estimation methods are observed.

References

1. Saaty, T. L.: *The Analytic Hierarchy Process*, McGraw-Hill, New York, 1980.
2. Innan S., Inuiguchi, M.: Advantages of minimum range based methods for the interval weight estimation in the setting of AHP, *Proc. 22th ISIS*, GO7-2, 2021.
3. Inuiguchi, M., Torisu, I.: The advantage of interval weight estimation over the conventional weight estimation in AHP in ranking alternatives, *Proc, IUKM 2020*, LNAI 12482, Berlin, Heidelberg, Springer, pp.38–49, 2020.
4. Inuiguchi, M., Hayashi, A., Innan, S.: Comparing the ranking accuracies among interval weight estimation methods at the standard, minimum and maximum solutions under crisp pairwise comparison matrices, *Proc. Joint 12th SCIS & 23rd ISIS*, Ise, Japan, pp.1–4, 2022.
5. Inuiguchi, M.: Non-uniqueness of interval weight vector to consistent interval pairwise comparison matrix and logarithmic estimation methods, *Proc, IUKM 2016*, LNAI 9978, Berlin, Heidelberg, Springer, pp.39–50, 2016.
6. Sugihara, K., Tanaka, H.: Interval evaluations in the analytic hierarchy process by possibility analysis, *Computational Intelligence*, 17(3), 567–579 (2001)
7. Inuiguchi, M., Innan, S.: Improving interval weight estimations in interval AHP by relaxations, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 21(7), 1135–1143 (2017)

A fuzzy-based method to boost short time-series to solve class imbalance in health care data

Jordi Pascual-Fontanilles[1][0000−0002−7528−5819], Aida Valls[1,3][000−0003−3616−7809], and Pedro Romero-Aroca[2,3][0000−0002−7061−8987]

[1] ITAKA, Dept. Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili, Tarragona, Catalonia, Spain
[2] Servei d'Oftalmologia, Hospital Universitari Sant Joan de Reus, Catalonia, Spain
[3] Institut d'Investigació Sanitària Pere Virgili, Tarragona, Catalonia, Spain

Abstract. When working with historical medical data from patient's electronic health record (EHR), we may have sequences of very different lengths, as some patients are visited more frequently than others. Moreover, when screening the patients for specific diseases, the number of patients that test positive is usually much smaller than the ones that test negative. Therefore, there is a high class imbalance towards the negative class. In a previous work, we presented a method for pre-processing medical multivariate time-series data from EHR in order to have a set of sequences of the same length. Patients with very short EHR were discarded. In this paper, we propose a novel technique to make use of short EHR series to minimize class imbalance. Long time series are used to synthetically complete the short time series. For numerical data, a fuzzy-based approach is proposed to generate additional similar positive examples. The proposed method has been tested with the problem of Diabetic Retinopathy classification. Results show that it improves the performance obtained by applying random oversampling to the data.

Keywords: Fuzzy Logic · Time Series · Class Imbalance · Diabetic Retinopathy

1 Introduction

Diabetic Retinopathy (DR) is an ocular complication due to diabetes. Its progression leads to the eye blood vessels break and may also generate small blood spots, hemorrhages and exudates. These lesions produce vision loss and may even cause blindness if they are not detected and treated at an early stage [8]. The incidence of DR in the diabetic population is about 12%. Due the increasing number of diabetic people, the identification of the patients with risk of developing DR is of high importance and proper screening tests must be done (i.e. control with eye-fundus images). However, diabetic people with no risk does not need to perform such tests, saving time and resources. The assessment of Diabetic Retinopahy risk can be addressed using clinical and analytical data stored in the electronic health records (EHR) of a patient, as done in systems like

Retiprogram [10, 5, 6]. This system uses 9 risk factors to determine the DR level. According to the ETDRS standard classification [11], the categories in which DR patients can be classified are: no retinopathy ($DR = 0$), mild ($DR = 1$), moderate ($DR = 2$) and severe ($DR = 3$). They are ordered from the best to the worst medical condition.

The DR classification problem from EHR data is quite difficult to solve for several reasons. First, patients with similar values on the risk factors can have different risk levels of DR. The low incidence of DR leads to a scarce availability of positive DR examples. Thus, there is a high imbalance of the data towards the negative class. Recently, we began to study the possibility of using the information of the past data available in the EHR to make the classification of patients. In a previous work [6], we explained a method for transforming the historical EHR data into equal-length multivariate time series. Then, by using multivariate time series classifiers, we were able to improve the prediction of the current DR risk level. The sequences used had between 6 and 10 entries, and they were turned into time series of length 10 (with interpolation methods).

In this work, we propose a novel fuzzy-based method to compensate for class imbalance on DR time series data. In the previous work, we discarded the sequences with less than 6 values to avoid inferring too much data that could end being incorrect. However, due to the scarcity of DR patients, we propose now to use the short series data to generate synthetic data that can be used as positive examples in the training of the classification methods.

A fuzzy approach has been used during the generation of new data values, because doctors reason qualitatively on the attribute values when assessing the patients' conditions (e.g., age: child/young/old; body mass: underweight/normal/overweight; hypertension: good control/bad control, etc.). For health treatment, a difference of one year in age, or of one kilogram makes no difference in the diagnosis, as it is done at a more general level (with labels corresponding to intervals). Fuzzy approaches can be used to reason qualitatively. In the literature, several fuzzy-based clinical decision support systems can be found. For instance, Hamedan et al. [2] use fuzzy linguistic variables to predict a chronic kidney disease and Nazari et al. [4] to diagnose heart diseases. The Retiprogram system is also a fuzzy-based classifier [10, 5, 6]. Therefore, in this work, we take advantage of the fuzzy linguistic model to generate different numerical values for fictive patients, which correspond to the same labels of real patients. We generate synthetic values making use of fuzzy linguistic variables in order to introduce some degree of variability o the new samples, without assigning unreal values. The numerical risk factors have been transformed to fuzzy linguistic variables with the knowledge of specialized ophthalmologists. By means of the fuzzy sets defined, we introduce some variability to the new generated examples, without affecting the patterns that the classifier must learn.

The rest of the paper is organized as follows. A short review of related work is done in Section 2. Next, Section 3 presents the proposed approach for boosting short time series. In Section 4, the data used to perform the experiments is

presented, and the obtained results are shown. Finally, Section 5 presents the conclusions and the future work.

2 Related work

Most state-of-the-art time series classifiers are not suited to solve problems with imbalanced class distributions, thus, methods to balance the amount of data at each class are commonly used. Three main approaches can be used to avoid class imbalance on time series data:

1. Sampling methods: some techniques are applied to the data on the original dataset to oversample and/or undersample it. For instance, in random oversampling, examples of the positive (minority) classes are replicated to balance the class distribution. On the contrary, undersampling consists on randomly removing examples from the majority class.
2. Synthetic data generation: the examples introduced to compensate the class imbalance are artificially generated from the existing data. The most common method is SMOTE (Synthetic Minority Over-Sampling Technique) [1]. Synthetic data points are generated by taking one of the k-nearest neighbours of a sample, and randomly choosing one point of the vector that unites the sample and the selected nearest neighbour. On the literature, several variations or methods based on the methodology of SMOTE can be found. For instance, T-SMOTE [13] is a variation for time series which takes into account the temporal characteristics of the data to select the nearest neighbours. T-SMOTE can be used on both univariate and multivariate time series.
3. Data augmentation: slightly modified copies of the data or synthetic examples created from the existing data are introduced to compensate for the class imbalance. Methods are highly dependent on the data types that have to be augmented. In the time series case, Iwana and Uchida [3] analysed over 50 data augmentation methods for time series, and they proposed a taxonomy with 4 families of methods: Random transformation methods apply a transformation function with some randomness to the time series; Pattern mixing combines patterns to generate new ones, which overcomes the assumption that all random transformations are possible on the data; Generative models use either statistical or neural network models to sample time series from feature distributions; Time series decomposition uses feature extraction techniques to extract features or underlying patterns, which are then used to generate new examples.

In medical diagnosis, the patients' values of the different risk factors are not totally independent. Although doctors know that there are some underlying relations, they are not usually completely defined. For instance, doctors may know that some combinations of values are not possible. Consequently, it is important that the balancing method used does not generate examples that may not be not real, as this may hamper the quality of the classifier built. This

paper proposes a new method that combines both synthetic data generation and random transformation data augmentation. On the one hand, short series are extended by synthetically generating the missing data. On the other hand, we are also conditioning the generated data to be similar to other existing examples (i.e. to a real patient), which is something that cannot be assured when using interpolation without introducing further pre-processing.

3 Time series generation

At each visit, doctors collect some clinical and analytical data about the diabetic patient. The most relevant risk factors for DR consist of six numerical and three categorical variables, which were selected by experts [9]. Numerical variables include the current age, body mass index (BMI), duration of Type-2 diabetes (EVOL), HbA1c, CKDEPI and microalbuminuria (MA). Categorical variables include gender, treatment of Type-2 diabetes (TTM) and control of arterial hypertension (HTAR). Sometimes they also have an eye fundus image, captured with special non-mydriatic cameras. However, we will not consider image data in this work, as our goal is to avoid the cost of obtaining such images. With this information, the ophthalmologist determines the degree of DR, which is stored in the EHR.

Diabetic patients are visited every 6 to 24 months, depending on the level of risk of developing the retinopathy disease. After some years, we can collect a sequence of entries with the medical data of each of the visits, stored in the hospital EHR. This inforamtion can be structured as a time series dataset that can be used to train a DR classifier. The DR variable should also be included as a categorical variable in each entry, except for the last one, which is the value we want to predict using an automatic classifier. The DR value of the last visit is taken, then, as the ground truth value. In our case study, after some pre-processing [6], we obtained a set with 2108 patients, each of them with a multivariate time series of length 10, where each variable is encoded as a different time series. For convenience, the interval between two entries was set to be 1 year. During pre-processing, some patient's data was discarded, concretely the ones with sequences with too short length, as the interpolation method could not find appropriate values. The data of these patients is now used to generate partially-synthetic instances for the minority classes.

In the procedure to generate new examples for short time series, we distinguish two subsets:

- C_p are sets of complete time series for the minority classes. In the case of DR, it contains the positive categories, $p \in \{1, 2, 3\}$. All the series have the same length, l_c.
- I is the set with incomplete time series, i_j, each one with a short length l_j. The length must be in the range $l_{min} < l_j < l_c$, where l_{min} must be determined by the characteristics of the data.

In short, we propose to generate new examples of length l_c by means of extending (i.e. boosting) the information available in short series. For each minority class p, the additional entries added at the end of the existing sequence will take into account the information available in the set C_p. In that way, we introduce data values they are feasible, as some other patient has had similar values. In the following subsections, the method to boost the incomplete set I using the complete set C_p is explained. In the procedure, variables are treated differently according to their nature. The method is applied to all examples of the incomplete set, $i_j \in I$, for each of the minority classes $p \in \{1, 2, 3\}$. Examples in I do not have a ground truth value, hence, they can be completed using examples from different classes, generating different sequences for each class.

3.1 Demographic variables

First, we consider the demographic variables, whose progression is known in advance. In the DR case study, they are age, gender and EVOL (duration of diabetes). Age and EVOL are numerical and they are measured in years, so at each time point in the series (yearly intervals), they increase in 1 unit. Gender is a categorical variable with a value fixed along the time, so the same category (woman/man) is maintained equal in all the new entries for each given time series i_j.

3.2 Medical variables

These are variables that store clinical and analytical information related to health. For the medical variables, we calculate the distance between a given incomplete series $i_j \in I$ (with length l_j) and a complete series $c_{p,k} \in C_p$ (with length l_c). As $l_j < l_c$, for the complete series we only consider the first l_j entries for the distance calculation. Dynamic Time Warping (DTW) is proposed as the distance measure for comparing the sequences, which is a well-known measure for this kind of data. The dependent version of DTW, DTW_D, has been applied in the case study [7].

This comparison is performed for each of the minority classes p. For each class, we find the example from the complete set with the least distance to i_j, i.e. the most similar in class p, denoted c_{p,sim_j}.

$$c_{p,sim_j} = argmin_k(DTW(i_j, c_{p,k})) \ \forall c_{p,k} \in C_p \tag{1}$$

Once we know the most similar complete series to an incomplete one, the procedure for assigning the following missing values of the sequence depends on being a numerical or categorical variable.

Categorical variables. We have three categorical variables: TTM, HTAR, and the class label DR. Their values in the incomplete entries of i_j are completed using the categorical values of the most similar series, c_{p,sim_j}. This corresponds to the missing time points $t \in (l_j, l_c]$. Regarding the class variable DR, which must

be monotonic non-decreasing according to the medical specialists, a forward fill is applied in the time points where copying the value would produce a decrease from the previous DR level. This process mainly affects the first generated time points, where some discrepancies between the incomplete and complete sequences could be found.

Numerical variables. For the management of numerical values, we propose a procedure based on fuzzy sets. Doctors usually work with ranges of values with fuzzy boundaries rather than with precise numerical values. We consider that for each numerical variable a, we can define a linguistic fuzzy variable f_a with a fixed set of ordered labels. Each label has a fuzzy set μ_a. In our case study, ophthalmologists provided appropriate linguistic labels and fuzzy sets for the numerical variables $A = \{CKDEPI, HbA1c, MA, BMI\}$. For each variable ($a \in A$) and for all missing time points $t \in (l_j, l_c]$, the following procedure is applied.

First, a forecasting method is used to predict the next numerical value for the incomplete time series, $i_j(a,t)$. Drift forecasting has been chosen because of its simplicity. Moreover, the amount of available past data is limited, so more complex forecasting techniques were not needed. It fits a line between the first and last points of the series, and extrapolates them to the future.

Second, the same time point is obtained from the nearest complete time series, $c_{p,sim_j}(a,t)$. The fuzzy sets of the variable f_a are then used to obtain the label with maximum activation for both the incomplete and complete time series values, x and y, respectively. If $x = y$, the forecasted value is stored in $i_j(a,t)$. Otherwise, a random value with maximum activation on the fuzzy term y is set in $i_j(a,t)$.

By forcing the forecasted value to be similar to one in the complete sequence, we can generate new examples that, although not having the same values, are similar. The use of fuzzy sets permits to assign values that are fuzzified with the same label, which means that they are falling the same category according to the vocabulary given by the ophthalmologists.

4 Experimental results

This section presents the obtained experimental results, where random oversampling is compared to this proposed approach on the DR dataset. In subsection 4.1 the dataset is presented. Subsection 4.2 shows and discusses the results.

4.1 Dataset

This subsection presents the data that has been used for the experiments. It comes from Catalan diabetic population from period 2010 to 2021. It is a private dataset provided to us in the framework of a national research project. Table 1 shows the number of patients of the complete set for the 4 possible DR classes. The dataset was split between training (20%) and testing (80%). All the

sequences in these sets have length $l_c = 10$. Although the negative class is not used in the example generation process, it is also included on Table 1 to show the clear imbalance towards the negative class.

Table 1. Diabetic retinopathy patients with complete time series

Class/Dataset	**Training (20%)**	**Testing (80%)**	**Total**
DR = 0	354 (84.1%)	1376 (81.6%)	1730 (82.1%)
DR = 1	41 (9.7%)	168 (10%)	209 (9.9%)
DR = 2	22 (5.2%)	111 (6.6%)	133 (6.3%)
DR = 3	4 (1%)	32 (1.8%)	36 (1.7%)
Total	421	1687	2108

To balance the training set, we applied the presented method for completing short sequences for the three minority classes. Here we use incomplete set I. It is composed by previously discarded time series because of their short length. In this work, we just considered the incomplete series of the maximum length available, $l_j = 5$, which correspond to 4547 patients. This is because the shorter the incomplete time series are, more fictive data has to be introduced, increasing the probabilities of introducing erroneous data.

After generating new examples for each minority class, we have oversampled these classes on the training set to reach the number of elements in the majority one (i.e. each class has 354 examples).

4.2 Results

The validity of the balanced data obtained with the proposed method has been studied by training and testing a time series classifier. According to the results we obtained in [6], the Convolutional Neural Network (CNN) is the time series classifier that performs better in the DR dataset. We used a CNN architecture specifically designed for time series classification [12]. The CNN configuration used for the tests is the following one: 20 epochs, batch size of 16, kernel size of 5, average pool size of 2, softmax activation, and categorical crossentropy loss.

The obtained results have been compared with a baseline balancing method consisting on making random oversampling. Even being a simple technique, it is still quite used for balancing.

Figure 1 depicts the confusion matrices obtained with the CNN classifier for both balancing methods. From the results, it can be seen how the proposed method improves the random oversampling. It is able to correctly classify more examples for classes $DR = 0$ and $DR = 1$. Moreover, it is clearly more capable of detecting patients with severe retinopathy $DR = 3$.

Several standard performance metrics for multiclass classification have been used to evaluate both methods, as shown in Table 2. Random oversampling results are quite high, but the proposed method obtains better results in all the metrics. Taking into account that the testing set is imbalanced, some of the

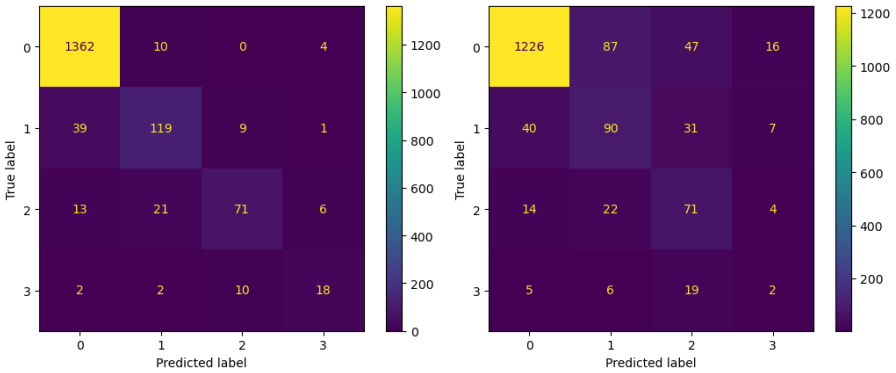


Fig. 1. Confusion matrices for proposed approach (left) and random oversample (right)

metrics may be biased towards the performance of the largest class. Even so, accuracy, weighted recall and weighted F1 have an improvement of at least 10% over random oversampling when using the proposed approach to balance the training set. In macro metrics, for recall we have an improvement of 40% and of 50% in macro F1, with respect to the baseline random oversampling. It is worth to highlight the improvement in the macro metrics, which indicates that the trained classifier with the proposed balancing method can better identify the positive patients. Finally, the quadratic weighted kappa increase is specially remarkable, reaching a value of 0.85. It indicates a substantial strength of agreement between the predictions and the ground truth.

Table 2. Performance of the proposed approach and random oversampling

Metric/Balancing	**Proposed balancing**	**Random oversampling**
Accuracy (%)	93	82
Quadratic Weighted Kappa	0.85	0.59
Macro Recall (%)	73	53
Weighted Recall (%)	93	82
Macro F1 (%)	75	49
Weighted F1 (%)	93	83

5 Conclusions and future work

In this paper, we presented a new fuzzy-based approach to boost short time series to solve class imbalance in health care data. We have generated completed short sequences by using information from similar completed ones. Three types of variables have been distinguished when completing their values. For medical

numerical values, a method based on the use of linguistic fuzzy variables has been proposed. By using the membership functions, we can find new input values that generate series similar enough to real examples. The resulting DR dataset has been compared to a set balanced with random oversampling, by using a CNN time series classifier. The obtained results clearly indicate that our proposed method for generating new positive examples is better on compensating for class imbalance than using random oversampling. The time series classifier, CNN, is able to better learn the underlying patterns of the minority DR examples, as metrics depict. Moreover, as the quadratic weighted kappa indicates, when it commits mistakes, they are closer to the ground truth.

As future work, we plan to test the proposed method on other time series classifiers and other datasets to confirm the observations. A comparison with other balancing techniques (T-SMOTE) will be made. The possibility of using or adapting the proposed method for shorter incomplete time series should also be studied.

Acknowledgements This study has been funded by Instituto de Salud Carlos III (ISCIII) through the project PI21/00064 and co-funded by the European Union. Also by the URV projects 2022PFR-URV-41 and 2021PFR-B2-103. The first author has a pre-doctoral FI grant (2022 FI_B1 00036) from Generalitat de Catalunya and Fons Social Europeu.

References

1. Chawla, N. v, Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P.: SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research **16**, 321–357 (2002)
2. Hamedan, F., Orooji, A., Sanadgol, H., Sheikhtaheri, A.: Clinical decision support system to predict chronic kidney disease: A fuzzy expert system approach. International Journal of Medical Informatics **138**, 104134 (2020)
3. Iwana, B. K., Uchida, S.: An empirical survey of data augmentation for time series classification with neural networks. PLOS ONE **16**(7), 0254841 (2021)
4. Nazari, S., Fallah, M., Kazemipoor, H., Salehipour, A.: A fuzzy inference- fuzzy analytic hierarchy process-based clinical decision support system for diagnosis of heart diseases. Expert Systems with Applications **95**, 261–271 (2018)
5. Pascual-Fontanilles, J., Lhotska, L., Moreno, A., Valls, A.: Adapting a Fuzzy Random Forest for Ordinal Multi-Class Classification. Frontiers in Artificial Intelligence and Applications **356**, 181–190 (2022)
6. Pascual-Fontanilles, J., Valls, A., Romero-Aroca, P.: A Diabetic Retinopathy classifier based on time-series clinical and analytical patient's data. The 26th European Conference on Artificial Intelligence. Submitted
7. Pasos Ruiz, A., Flynn, M., Large, J., Middlehurst, M., Bagnall, A.: The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery **35**, 401–449 (2021)
8. Romero-Aroca, P., de La Riva-Fernandez, S., Valls-Mateu, A., Sagarra-Alamo, R., Moreno-Ribas, A., Soler, N.: Changes observed in diabetic retinopathy: eight-year

follow-up of a Spanish population. British Journal of Ophthalmology **100**(10), 1366–1371 (2016)
9. Romero-Aroca, P., Valls-Mateu, A., Moreno-Ribas, A., Sagarra-Alamo, R., Basora-Gallisa, J., Saleh, E., Baget-Bernaldiz, M., Puig, D.: A Clinical Decision Support System for Diabetic Retinopathy Screening: Creating a Clinical Support Application. Telemedicine and e-Health **25**(1), 31–40 (2019)
10. Saleh, E., Błaszczyński, J., Moreno, A., Valls, A., Romero-Aroca, P., de la Riva-Fernández, S., Słowiński, R.: Learning ensemble classifiers for diabetic retinopathy assessment. Artificial Intelligence in Medicine **85**, 50–63 (2018)
11. Wilkinson, C. P., Ferris, F. L., Klein, R. E., Lee, P. P., Agardh, C. D., Davis, M., Dills, D., Kampik, A., Pararajasegaram, R., Verdaguer, J. T.: Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. Ophthalmology **110**(9), 1677–1682 (2003)
12. Zhao, B., Lu, H., Chen, S., Liu, J., Wu, D.: Convolutional neural networks for time series classification. Journal of Systems Engineering and Electronics **28**(1), 162–169 (2017)
13. Zhao, P., Luo, C., Qiao, B., Wang, L., Rajmohan, S., Lin, Q., Zhang, D.: T-SMOTE: Temporal-oriented Synthetic Minority Oversampling Technique for Imbalanced Time Series Classification. Proceedings of IJCAI (2022)

Fuzzy approach to differential entropy*

Zuzana Ontkovičová[1] [0000−0001−8443−5543]

Institute of Information Engineering, Automation and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Slovak Republic
zuzana.ontkovicova@stuba.sk

Abstract. The paper offers a fuzzy insight into differential entropy. Its formula is revised for fuzzy measures using extended Choquet integrals and Choquet-Radon-Nikodym derivatives as a fuzzy alternative to the additive Radon-Nikodym derivatives. The computational aspect of entropy is examined for the particular class of distorted Lebesgue measures, and also some of the basic entropy properties are studied resulting from modifying one of the measures in the formula. Derivation from Kullback-Leibler divergence is introduced, and other types of entropy, such as Rényi and Tsallis, are generalised on the same basis to the fuzzy setup.

Keywords: Differential entropy · Fuzzy measures · Extended Choquet integrals · Choquet-Radon-Nikodym derivatives.

1 Introduction - Additive case

The notion of entropy originated in the 1870s in thermodynamics and statistical physics, where it was derived from thermodynamical considerations based upon the second law of thermodynamics [12]. Later, entropy was applied by Hartley in his study of random signals and also served as one of the main concepts in the information theory developed by Shannon [8] in the 1940s. Since then, it has been a very important concept in several areas of scientific research and has been widely used in many distinct applications, such as engineering, finance, decision making, optimisation, system modelling or image processing.

This article focuses on the original mathematical insight into entropy in the field of information theory, according to Shannon. Our aim is to make an analogy from the already introduced differential entropy in the additive (probability) case to the fuzzy (nonadditive) setup. The main difference is replacing classic probability with fuzzy measure, which does not possess the key property of additivity. This replacement leads to more necessary and not so trivial changes, namely in the integral, RN derivatives and even the formula itself, which we study one by one in the following parts of the article. To our best knowledge, the only article dealing with differential entropy for fuzzy measures the way we would like to is [11], where the authors mainly focus on a more general case of entropy called Kullback-Leibler divergence.

Since information theory can be simply considered a branch of applied probability theory, let us recall some basic probabilistic concepts necessary for this paper. Consider a probability space $(\Omega, \mathcal{A}, P)$, with a probability measure P. Then a mapping $\xi : \Omega \to [-\infty, +\infty]$ is called a random

*This paper was supported by Agency of the Ministry of Education, Science, Research and Sports of the Slovak Republic and the Slovak Academy of Sciences under the contract No. VEGA 1/0267/21.

variable (RV) if for all $t \in \mathbb{R}$ it holds $\{\omega \in \Omega : \xi(\omega) < t\} \in \mathcal{A}$. For a description of RV ξ, cumulative distribution function (cdf) $\mathcal{H}_\xi : \mathbb{R} \to \mathbb{R}$ is commonly used, having the form

$$\mathcal{H}_\xi(t) = P\left(\{\omega \in \Omega : \xi(\omega) \leq t\}\right), \quad t \in \mathbb{R}.$$

The cdf is said to be absolutely continuous if there exists a function $h : \mathbb{R} \to [0, \infty]$ such that $\int_{-\infty}^{\infty} h(t)\,\mathrm{d}t = 1$ and $\mathcal{H}_\xi(t) = \int_{-\infty}^{t} h(x)\,\mathrm{d}x$. Then h is called probability density function (pdf), and it holds $h(t) = \dfrac{\mathrm{d}\mathcal{H}_\xi(t)}{\mathrm{d}t} = \mathcal{H}'_\xi(t)$ if the derivative exists.

It is obvious that probability measure, cdf and pdf are all closely interconnected, so a direct link between probability measure and pdf exists, namely in the form of Radon-Nikodym derivative.

Theorem 1 (Radon-Nikodym theorem). *On a measurable space $(\Omega, \mathcal{A})$, let us assume two σ-finite[1] measures μ, ν, where $\nu \ll \mu$[2]. Then there exists a measurable function $f : \Omega \to [0, \infty]$, such that for any measurable set $A \in \mathcal{A}$ it holds*

$$\nu(A) = \int_A f\,\mathrm{d}\mu.$$

Function f from Theorem 1 is the Radon-Nikodym (RN) derivative, commonly written as $f = \dfrac{\mathrm{d}\nu}{\mathrm{d}\mu}$. Measure μ is also called reference measure.

Pdf can be written in our proposed setup as RN derivative $h = \dfrac{\mathrm{d}P}{\mathrm{d}\lambda}$, where λ is Lebesgue measure. With all this in mind, differential entropy of continuous RV ξ with corresponding pdf h is defined with the formula

$$\mathsf{Ent}(\xi) = -\int h(t) \ln h(t)\,\mathrm{d}t,$$

with a convention $0 \cdot \ln 0 = 0$ for a proper definition, resulting from the L'Hospital rule. In case when corresponding RN derivative is used instead of pdf, the formula for differential entropy changes to the form

$$\mathsf{Ent}(P) = -\int \frac{\mathrm{d}P}{\mathrm{d}\lambda} \ln\left(\frac{\mathrm{d}P}{\mathrm{d}\lambda}\right) \mathrm{d}\lambda \tag{1}$$

and after modification resulting from (change of variables) property of RN derivatives, it can be shortened as

$$\mathsf{Ent}(P) = -\int \ln\left(\frac{\mathrm{d}P}{\mathrm{d}\lambda}\right) \mathrm{d}P. \tag{2}$$

The notation for entropy changed because it is more convenient to assume measure in the argument than RV, which even does not appear in the formula. These two formulas (1) and (2) are our starting point for fuzzy insight into the definition of differential entropy.

It is necessary to mention that there are significant differences between discrete and differential (continuous) entropy. One is that the final value of differential entropy cannot be directly interpreted. It is only interpretable when comparing values or after division with the pre-agreed reference

[1] measure μ is σ-finite if $(\forall (A_n)_{n\in\mathbb{N}} \in \mathcal{A} : A_i \cap A_j = \emptyset\ i \neq j, \Omega = \bigcup_{n\in\mathbb{N}} A_n)\ \ \mu(A_n) < \infty$

[2] $\nu \ll \mu$ is notation for absolute continuity of ν with respect to μ or $(\forall A \in \mathcal{A})\ \mu(A) = 0 \Rightarrow \nu(A) = 0$.

entropy. Another difference is that the final value of entropy can be any real number, even a negative one, as shown in Example 1. The reason is that there is a pdf in the argument of the logarithm, while in the discrete case, there is probability measure bounded in the interval $[0, 1]$, resulting in a nonnegative value.

Example 1. Let ξ be a random variable described through Laplace distribution with pdf $h_\lambda(x) = \frac{\lambda}{2}e^{-\lambda x}$. Then its differential entropy is given as

$$\mathsf{Ent}(\xi) = -2\int_0^\infty \frac{\lambda}{2}e^{-\lambda x}\,\ln\left(\frac{\lambda}{2}e^{-\lambda x}\right)\mathrm{d}x = \ln\left(\frac{2e}{\lambda}\right).$$

Taking e.g. $\lambda = 10$, $\mathsf{Ent}(\xi) \doteq -0.6094$, so even for additive case entropy can be negative.

2 Main results - Fuzzy case

On a measurable space $(\Omega, \mathcal{A})$, a fuzzy measure $\mu : \mathcal{A} \to [0, 1]$ is a set function which satisfies the following properties:

- (groundedness) $\mu(\emptyset) = 0$
- (normalisation) $\mu(\Omega) = 1$
- (monotonicity) $(\forall A, B \in \mathcal{A} : A \subseteq B)\, \mu(A) \leq \mu(B)$
- (continuity from below) $(\forall n \in \mathbb{N})\,(\forall C, C_n \in \mathcal{A} : C_n \nearrow C)\ \mu(C_n) \nearrow \mu(C)$
- (continuity from above) $(\forall n \in \mathbb{N})\,(\forall C, C_n \in \mathcal{A} : C_n \searrow C,\ \mu(C_1) < \infty)\ \mu(C_n) \searrow \mu(C)$

Using fuzzy measures in the integration leads to a transition from additive Lebesgue integral to its fuzzy equivalent Choquet integral. For a nonnegative RV ξ and fuzzy measure μ, it is defined as the indefinite Riemann integral [1], [2] in the form

$$(C)\int \xi\,\mathrm{d}\mu = \int_0^\infty \mathcal{S}_{\mu,\xi}(t)\,\mathrm{d}t,$$

where

$$\mathcal{S}_{\mu,\xi}(t) = \mu(\{\omega \in \Omega : \mu(\omega) > t\}),\ t \in \mathbb{R}$$

is the corresponding survival function, emphasising the use of the fuzzy measure in the lower index. When assuming general RV with both positive and negative values, there are two different ways to extend the Choquet integral, namely symmetric (C_s) and asymmetric (C_a) versions given as

$$(C_a)\int \xi\,\mathrm{d}\mu = (C)\int \xi^+\,\mathrm{d}\mu - (C)\int \xi^-\,\mathrm{d}\overline{\mu},$$
$$(C_s)\int \xi\,\mathrm{d}\mu = (C)\int \xi^+\,\mathrm{d}\mu - (C)\int \xi^-\,\mathrm{d}\mu.$$

Apparently, these two Choquet integrals differ in the formula only for the nonpositive part, where the crucial is the dual measure[3] appearing only in the asymmetric integral. This fact also influences the integrals' properties, which can be found in [1] mainly for nonnegative RV and in [7] for general RV.

Further, we can generalise RN derivatives for fuzzy measures with the introduced Choquet integral, as in [6].

[3] Dual measure with notation $\overline{\mu}$ is defined as $\overline{\mu}(A) = \mu(\Omega) - \mu(A^c)$ for all $A \in \mathcal{A}$.

Definition 1 (Choquet-Radon-Nikodym derivatives). *On a measurable space $(\Omega, \mathcal{A})$, let us assume two σ-finite fuzzy measures μ, ν and Choquet integral of nonnegative measurable function g for any measurable set $A \in \mathcal{A}$ given as*

$$\nu(A) = (C) \int_A g \, \mathrm{d}\mu.$$

Then g is a Choquet-Radon-Nikodym (CRN) derivative with notation $g = \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}$ distinguished from the RN derivatives.

CRN derivatives are always nonnegative functions, so there is no need to use extended Choquet integrals. Moreover, both assumed measures need to be fuzzy, otherwise we obtain the additive case. Regarding their existence, absolute continuity is not satisfactory here as for additive measures, so more conditions need to be assumed. Either submodularity of both measures is added [4] or subadditivity of measures together with the strong decomposition property (derived from the Hahn decomposition of measures) is further assumed [3]. There is also an additional condition on measures assuming the uniqueness of CRN derivatives because dealing with Choquet integral, as given in [6].

Proposition 2 *On a measurable space $(\Omega, \mathcal{A})$, let us have two fuzzy measures μ, ν.*

i) If μ is subadditive[4]*, $g = \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}$ and f is a $\mathcal{A}$-measurable function on Ω such that $f = g$ μ-a.e., then $f = \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}$.*

ii) If $f = \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}$ and $g = \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}$, then $f = g$ μ-a.e.

In the following two propositions, we focus on properties adopted from [6] necessary for the paper. The first one is related to the change of variables in the Choquet integral using CRN derivatives, and the second one deals with their basic calculus.

Proposition 3 *On a measurable space $(\Omega, \mathcal{A})$, let us assume two fuzzy measures μ, ν with μ being σ-finite, and existence of the CRN derivative $\frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}$. Then for a nonnegative random variable ξ, it holds*

$$(C) \int \xi \, \mathrm{d}\nu = (C) \int \xi \, \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu} \, \mathrm{d}\mu,$$

where the existence of one side implies that of the other.

As a generalisation of Proposition 3, we take general RV ξ with positive and negative values. The equation then holds only for symmetric, but not for asymmetric Choquet integral because positive as well as negative homogeneity needs to be satisfied.

Lemma 1. *Let μ, ν, τ be fuzzy measures and let us assume the existence of all the necessary CRN derivatives. Then*

i) (homogeneity) for $k > 0$ it holds $\frac{\mathfrak{d}(k\nu)}{\mathfrak{d}\mu} = k \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}$ μ-a.e. and $\frac{\mathfrak{d}\nu}{\mathfrak{d}(k\mu)} = \frac{1}{k} \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}$ μ-a.e.

[4] Subadditive measure $(\forall A, B \in \mathcal{A})$ $\mu(A) + \mu(B) \geq \mu(A \cup B)$.

ii) (chain rule) for σ-finite μ it holds $\dfrac{\mathfrak{d}\nu}{\mathfrak{d}\tau}\dfrac{\mathfrak{d}\tau}{\mathfrak{d}\mu} = \dfrac{\mathfrak{d}\nu}{\mathfrak{d}\mu}$ μ-*a.e.*

iii) (inverse) it holds $\dfrac{\mathfrak{d}\nu}{\mathfrak{d}\mu} = \left(\dfrac{\mathfrak{d}\mu}{\mathfrak{d}\nu}\right)^{-1}$

iv) (duality) for subadditive [superadditive[5]*] μ it holds* $\dfrac{\mathfrak{d}\nu}{\mathfrak{d}\mu} \leq \dfrac{\mathfrak{d}\nu}{\mathfrak{d}\overline{\mu}}$ μ-*a.e.* $\left[\dfrac{\mathfrak{d}\nu}{\mathfrak{d}\mu} \geq \dfrac{\mathfrak{d}\nu}{\mathfrak{d}\overline{\mu}}\right.$ $\left.\mu\text{-a.e.}\right]$

2.1 Definition

We have done all the necessary modifications from the probability case to the fuzzy setup, namely from probability to fuzzy measure, from Lebesgue to Choquet integral with extensions symmetric and asymmetric Choquet integrals, and from RN derivatives to CRN derivatives. Now, we can finally generalise the differential entropy formula itself, derived from the additive formulas (1) and (2).

Definition 2 (Fuzzy differential entropy). *On a measurable space $(\Omega, \mathcal{A})$, let us have two fuzzy measures μ, ν and let CRN derivative* $\dfrac{\mathfrak{d}\nu}{\mathfrak{d}\mu}$ *exist. Then fuzzy differential entropy of ν with respect to μ with asymmetric Choquet integral is given as*

$$\mathsf{Ent}^{a}_{\mu}(\nu) = -(C_a)\int \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\ln\left(\frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\right)\mathrm{d}\mu \tag{3}$$

and with symmetric Choquet integral by the formula

$$\mathsf{Ent}^{s}_{\mu}(\nu) = -(C_s)\int \ln\left(\frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\right)\mathrm{d}\nu. \tag{4}$$

The notation is adapted to the fuzzy case, so there is the reference measure in the lower index and a type of Choquet integral in the upper index. We propose two definitions (3) and (4), which differ in the type of Choquet integral resulting in their different forms. The first one with asymmetric integral is a direct analogy to the longer additive formula (2). The second one with symmetric integral is shortened according to Proposition 3 and comments below, so there is no CRN derivative multiplying the logarithm and the integration measure has changed. It is also a direct analogy to the shorter additive formula (2).

2.2 Computation

After properly defining differential entropy in the fuzzy case, it is interesting to look at the practical computational aspect. First, we focus on the Choquet integral. To avoid a complicated reordering process, a simplification is used only assuming monotone functions and subsets of the real line, so when $\Omega \subseteq \mathbb{R}$. Particularly, in [9] and [10], only nondecreasing integrands and $\Omega = [0, \tau]$ are used. Here, we study both nondecreasing and nonincreasing functions on more general $\Omega = [s, \tau]$, where $0 \leq s \leq \tau$.

Proposition 4 *Let us assume $\Omega = [s, \tau] \subseteq \mathbb{R}$, $0 \leq s \leq \tau$ and fuzzy measure μ, generating functions $\mu([t, \tau])$ and $\mu([s, t])$, which are supposed to be differentiable with respect to t. Then the Choquet integral of nonnegative monotone continuous function g with respect to μ on $[s, \tau]$ have the form*

[5] Superadditive measure $(\forall A, B \in \mathcal{A})$ $\mu(A) + \mu(B) \leq \mu(A \cup B)$.

- *for nondecreasing g* $\quad (C)\displaystyle\int_s^\tau g\,\mathrm{d}\mu = s\,\mu([s,\tau]) - \int_s^\tau \frac{\partial}{\partial t}\mu([t,\tau])\,g(t)\,\mathrm{d}t,$
- *for nonincreasing g* $\quad (C)\displaystyle\int_s^\tau g\,\mathrm{d}\mu = s\,\mu([s,\tau]) + \int_s^\tau \frac{\partial}{\partial t}\mu([s,t])\,g(t)\,\mathrm{d}t.$

If we restrict ourselves only to distorted Lebesgue measures[6] as a special case of fuzzy measures, the corresponding Choquet integrals on $\Omega = [s,\tau]$ are given as follows.

Corollary 5 *Let $\Omega = [s,\tau] \subseteq \mathbb{R}$, $0 \le s \le \tau$ and λ_m be a distorted Lebesgue measure. Then the Choquet integral of nonnegative monotone continuous function g with respect to λ_m on $[s,\tau]$ has the form*

- *for nondecreasing g* $\quad (C)\displaystyle\int_s^\tau g\,\mathrm{d}\lambda_m = s\; m(\tau - s) + \int_s^\tau m'(\tau - t)g(t)\,\mathrm{d}t,$
- *for nonincreasing g* $\quad (C)\displaystyle\int_s^\tau g\,\mathrm{d}\lambda_m = s\; m(\tau - s) + \int_s^\tau m'(t - s)g(t)\,\mathrm{d}t,$

It is obvious that taking $s = 0$ in both proposition and corollary, we obtain already existing results for $\Omega = [0,t]$ as a special case.

Our main task regarding computation is CRN derivatives, which can be seen as the argument of the corresponding Choquet integral. Because it is quite challenging in general, we only restrict ourselves to subsets of the real line $\Omega = [0,\tau]$, monotone functions and distorted Lebesgue measures, so only formulas from Corollary 5 are needed.

In the case of a nondecreasing derivative, the Choquet integral coincides with the Riemann integral in the form of convolution. The best way for its computation is therefore using Laplace transform and its inverse. So putting $\xi_1(\tau) = \displaystyle\int_0^\tau m'(\tau - t)g(t)\,\mathrm{d}t$, function g as the corresponding CRN derivative is computed as

$$g(t) = \mathcal{L}^{-1}\left\{\frac{\mathcal{L}[\xi_1]}{s\mathcal{L}[m]}\right\}. \tag{5}$$

Even though this method with Laplace transform seems easy for computation, it yields some practical problems as shown in particular examples in [6]. In case of nonincreasing CRN derivative, let us put $\xi_2(\tau) = \displaystyle\int_0^\tau m'(t)g(t)\,\mathrm{d}t$. From the integral form, it is clear that the CRN derivative can be expressed through the fundamental theorem of integral calculus as

$$g(t) = \frac{\xi_2'(t)}{m'(t)}. \tag{6}$$

Regardless of the type of monotonicity, there is one particular inconvenience in this approach. At the beginning of the computation, we do not know if the derivative is nondecreasing or nonincreasing. So, we need to guess first and then check if the guess was correct or if we need to repeat the whole procedure.

[6] Distorted Lebesgue measure λ_m is a fuzzy measure, where $\lambda_m = m \circ \lambda$ with λ being Lebesgue measure and $m : \mathbb{R}^+ \to \mathbb{R}^+$ a distortion, so differentiable increasing function satisfying $m(0) = 0$ and $m(1) = 1$.

Example 2. For nondecreasing CRN derivative, let us assume $\xi_1(t) = \frac{1}{10}t^5 + \frac{1}{3}t^3$ and distorted Lebesgue measure with distortion $m(t) = t^2$. Then using (5), it holds

$$g_1(t) = \mathcal{L}^{-1}\left\{\frac{\mathcal{L}[\frac{1}{10}t^5 + \frac{1}{3}t^3]}{s\,\mathcal{L}[t^2]}\right\} = \mathcal{L}^{-1}\left\{\frac{6}{s^4} + \frac{1}{s^2}\right\} = t^3 + t.$$

For nonincreasing CRN derivative, we assume $\xi_2(t) = -4te^{-\frac{t}{2}} - 8e^{-\frac{t}{2}}$ with the same distorted Lebesgue measure. Then following (6), it holds

$$g_2(t) = \frac{\left(-4te^{-\frac{t}{2}} - 8e^{-\frac{t}{2}}\right)'}{(t^2)'} = e^{-\frac{t}{2}}.$$

For both monotonicities in Example 2, the form of the CRN derivative can be checked by inserting it into the formula for Choquet integral with respect to distorted Lebesgue measure λ_{t^2} in Corollary 5 and compare the result with corresponding ξ function.

After showing the computation of the Choquet integral and CRN derivatives in some restricted setup, we compute the entropy itself from formulas (3) and (4) with the use of integrals' definitions and Corollary 5.

Example 3. Let $\Omega = [0,1]$, $\mu = \lambda_m$ with $m(t) = t^3$ and $\nu = \lambda_n$ with $m(t) = t^2$. Our task is to compute the entropy of ν with respect to μ with both extended Choquet integrals.
First, CRN derivative is $\frac{\mathfrak{d}\nu}{\mathfrak{d}\mu} = \frac{(t^2)'}{(t^3)'} = \frac{2}{3t}$ because it is nonincreasing.
Then entropy with symmetric Choquet integral is computed as follows

$$\mathsf{Ent}^s_\mu(\nu) = -(C_s)\int_{[0,1]} \ln\left(\frac{2}{3t}\right) \mathrm{d}\lambda_{t^2} = -(C)\int_{\left[0,\frac{2}{3}\right]} \ln\left(\frac{2}{3t}\right) \mathrm{d}\lambda_{t^2} + (C)\int_{\left[\frac{2}{3},1\right]} \ln\left(\frac{3t}{2}\right) \mathrm{d}\lambda_{t^2} =$$

$$= -\int_0^{\frac{2}{3}} (t^2)' \ln\left(\frac{2}{3t}\right) \mathrm{d}t + \frac{2}{3}\left(1 - \frac{2}{3}\right)^2 - \int_{\frac{2}{3}}^1 \left((1-t)^2\right)' \ln\left(\frac{2}{3t}\right) \mathrm{d}t = \frac{13}{54} - \ln\left(\frac{3}{2}\right) \doteq -0.1647.$$

For asymmetric Choquet integral, the computation is very similar, with the one important change in measure, because the dual measure for nonpositive values is needed

$$\mathsf{Ent}^a_\mu(\nu) = -(C_a)\int_{[0,1]} \frac{2}{3t} \ln\left(\frac{2}{3t}\right) \mathrm{d}\lambda_{t^3} = -(C)\int_{\left[0,\frac{2}{3}\right]} \frac{2}{3t} \ln\left(\frac{2}{3t}\right) \mathrm{d}\lambda_{t^3} +$$

$$+(C)\int_{\left[\frac{2}{3},1\right]} \frac{2}{3t} \ln\left(\frac{3t}{2}\right) \mathrm{d}\overline{\lambda}_{t^3} = -\int_0^{\frac{2}{3}} \left(t^3\right)' \frac{2}{3t} \ln\left(\frac{2}{3t}\right) \mathrm{d}t + \frac{2}{3}\left(1 - \lambda_{t^3}\left(\left[0,\frac{2}{3}\right]\right)\right) -$$

$$-\int_{\frac{2}{3}}^1 (1 - \lambda_{t^3}([0,t]))' \frac{2}{3t} \ln\left(\frac{2}{3t}\right) \mathrm{d}t = \frac{85}{162} - \ln\left(\frac{3}{2}\right) \doteq 0.1192.$$

2.3 Properties

Next, we study basic properties of fuzzy differential entropy definitions (3) and (4) regarding both the reference measure and the measure in the argument. We change these two input measures and observe what happens with the resulting entropy. For their proof, the properties of integrals as well as CRN derivatives are taken into account.

First, we look at the uniqueness of the proposed entropy definition, so how could the measures resp. CRN derivatives differ without the change of the final entropy. For a more simple form of this proposition, we use notation $\mathsf{Ent}_\mu(f)$ with CRN derivative f in the argument.

Proposition 6 *Let μ, ν be fuzzy measures and f_1, f_2 two different CRN derivatives of ν with respect to μ. If μ, ν are subadditive and $f_1 = f_2$ ν-a.e., then*

$$\mathsf{Ent}^s_\mu(f_1) = \mathsf{Ent}^s_\mu(f_2).$$

If μ is subadditive and $f_1 = f_2$ μ-a.e., then

$$\mathsf{Ent}^a_\mu(f_1) = \mathsf{Ent}^a_\mu(f_2).$$

Proof. From the uniqueness property of CRN derivatives in Proposition 2, measure μ needs to be subadditive for both entropy versions. For asymmetric integral, when $f_1 = f_2$ μ-a.e. and μ as a measure in the integral is subadditive, also $f_1 \ln(f_1) = f_2 \ln(f_2)$ μ-a.e. and from [1, Example 9.1] $(C_a)\int f_1 \ln(f_1)\,\mathrm{d}\mu = (C_a)\int f_2 \ln(f_2)\,\mathrm{d}\mu$. Analogously, it can be done for symmetric integral with ν as a measure in the integral, so assuming further condition of its subadditivity.

The next two properties can be seen as a variation of positive homogeneity regarding both measures. The first one, with respect to the reference measure, is in the additive case (emphasising additive reference measure in the lower index) given as $\mathsf{Ent}_{k\lambda}(P) = \ln k + \mathsf{Ent}_\lambda(P)$ for all $k > 0$, $k \neq 1$. Dealing with fuzzy measures, this does not hold in general.

Proposition 7 *Let μ, ν be fuzzy measures, $k > 0$, $k \neq 1$, and let all the necessary CRN derivatives exist. Then for both extended Choquet integrals*

- $\mathsf{Ent}_{k\mu}(\nu) \leq \mathsf{Ent}_\mu(\nu)$ *for* $k \in (0,1)$
- $\mathsf{Ent}_{k\mu}(\nu) \geq \mathsf{Ent}_\mu(\nu)$ *for* $k > 1$.

Moreover, further inequalities can be obtained for asymmetric Choquet integral, so

- $\mathsf{Ent}^a_{k\mu}(\nu) \geq \ln k + \mathsf{Ent}^a_\mu(\nu)$ *for* $k \in (0,1)$ *and submodular* μ
- $\mathsf{Ent}^a_{k\mu}(\nu) \leq \ln k + \mathsf{Ent}^a_\mu(\nu)$ *for* $k \in (0,1)$ *and supermodular* μ

Proof. Let us focus on asymmetric Choquet integral.

$$\mathsf{Ent}^a_{k\mu}(\nu) = -(C_a)\int \frac{\mathfrak{d}\nu}{\mathfrak{d}(k\mu)} \ln\left(\frac{\mathfrak{d}\nu}{\mathfrak{d}(k\mu)}\right) \mathrm{d}(k\mu) = -(C_a)\int \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\left[-\ln k + \ln\left(\frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\right)\right] \mathrm{d}\mu$$

where positive homogeneity of CRN derivatives from Lemma 1 i) and of the integral regarding measure is used together with logarithm properties. The first approach is an estimation of the

constant $(-\ln k)$. Since if $k \in (0,1)$ then $-\ln k > 0$ and if $k > 1$ then $-\ln k < 0$, estimating it with zero (from below or above respectively) gives us the first two inequalities in the proposition. The same can be done for symmetric Choquet integral. The second approach is the use of weaker versions of the linearity (additivity) of the integral, where the further condition of submodularity of μ resulting in $\geq$ or supermodularity of μ resulting in $\leq$ is assumed (only submodular measure is taken in the derivation)

$$\geq -(C_a)\int \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\,(-\ln k)\;\mathrm{d}\mu - (C_a)\int \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\ln\left(\frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\right)\mathrm{d}\mu = -(C_a)\int \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\,(-\ln k)\;\mathrm{d}\mu + \mathsf{Ent}^a_\mu(\nu).$$

In the last integral, $-\ln k > 0$ for $k \in (0,1)$, so with positive homogeneity and CRN derivatives definition

$$-(C_a)\int \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\,(-\ln k)\;\mathrm{d}\mu = \ln k\,(C_a)\int \frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\,\mathrm{d}\mu = \ln k.$$

For $k > 1$, $-\ln k < 0$ and the dual measure needs to be assumed from negative homogeneity. Even though there exist inequalities comparing original and the dual measure, assuming both estimations results in supermodular and subadditive measure or submodular and superadditive measure, respectively. That leads to the additive case, which is not in our interest here. Estimations for linearity cannot be used for symmetric Choquet integral because weaker linearity holds with more strict conditions, which are not satisfied in general.

The second variation of homogeneity property is done regarding measure in the argument. The additive form of this property is given as $\mathsf{Ent}(kP) = -k\ln k + k\mathsf{Ent}(P)$ for $k > 0$, $k \neq 1$. For fuzzy measures, results are summarised in the following proposition.

Proposition 8 *Let μ, ν be fuzzy measures, $k > 0$, $k \neq 1$, and let all the necessary CRN derivatives exist. Then for both extended Choquet integrals*

- $\mathsf{Ent}_\mu(k\nu) \geq k\mathsf{Ent}_\mu(\nu)$ *for* $k \in (0,1)$
- $\mathsf{Ent}_\mu(k\nu) \leq k\mathsf{Ent}_\mu(\nu)$ *for* $k > 1$.

Furthermore, better boundaries can be obtained for asymmetric Choquet integral

- $\mathsf{Ent}^a_\mu(k\nu) \leq -k\ln k + k\mathsf{Ent}^a_\mu(\nu)$ *for* $k > 1$ *and supermodular* μ
- $\mathsf{Ent}^a_\mu(k\nu) \geq -k\ln k + k\mathsf{Ent}^a_\mu(\nu)$ *for* $k > 1$ *and submodular* μ

The proof can be done analogously to the previous proposition, with similar derivation steps and the same explanations, so we will not repeat it here.

The following property arises from the situation when the original measure is replaced with its dual. We observe if there is any relation between these two entropies.

Proposition 9 *Let μ, ν be fuzzy measures, $\overline{\mu}$ the dual measure to μ (also fuzzy) and let all the necessary CRN derivatives exist.*

- *If μ is subadditive, then* $\mathsf{Ent}^s_{\overline{\mu}}(\nu) \leq \mathsf{Ent}^s_\mu(\nu)$.
- *If μ is superadditive, then* $\mathsf{Ent}^s_{\overline{\mu}}(\nu) \geq \mathsf{Ent}^s_\mu(\nu)$.

Proof. The proof is straightforward, so

$$\mathsf{Ent}^s_{\overline{\mu}}(\nu) = -(C_s)\int \ln\left(\frac{\mathfrak{d}\nu}{\mathfrak{d}\overline{\mu}}\right)\mathrm{d}\nu \leq -(C_s)\int \ln\left(\frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\right)\mathrm{d}\nu = \mathsf{Ent}^s_{\mu}(\nu),$$

where the inequality follows from Lemma 1 iv), increasingness of logarithm and monotonicity of symmetric Choquet integral.

For asymmetric Choquet integral, there is a problem in comparison because the estimation for $\overline{\mu}$ needs to be done in CRN derivative as well as in the integral measure, which yields measure to be both subadditive and superadditive, so additive.

Dealing with the nonnegativity property of entropy, in Example 1 we already discussed that it is not satisfied for the additive differential entropy. Moreover, as could be expected, the same is true also for fuzzy differential entropy, with particular illustration for extended Choquet integrals in Example 3.

The entropy formula can also be derived from Kullback-Leibler divergence, which is a particular case of ϕ-divergence being one special class of information divergences. Introducing the derivation step by step, general ϕ-divergence in the most extended form is written as

$$\mathsf{D}^a_{\phi}(\nu_1,\nu_2:\mu) = (C_a)\int \frac{\mathfrak{d}\nu_2}{\mathfrak{d}\mu}\phi\left(\frac{\mathfrak{d}\nu_1/\,\mathfrak{d}\mu}{\mathfrak{d}\nu_2/\,\mathfrak{d}\mu}\right)\mathrm{d}\mu$$

introduced for the fuzzy setup in [6] for asymmetric Choquet integral indicated in the upper index. It can be modified to the form

$$\mathsf{D}^a_{\phi}(\nu_1,\nu_2:\mu) = (C_a)\int \frac{\mathfrak{d}\nu_2}{\mathfrak{d}\mu}\phi\left(\frac{\mathfrak{d}\nu_1}{\mathfrak{d}\nu_2}\right)\mathrm{d}\mu$$

because with Lemma 1 ii) and iii) it holds $\dfrac{\mathfrak{d}\nu_1/\,\mathfrak{d}\mu}{\mathfrak{d}\nu_2/\,\mathfrak{d}\mu} = \dfrac{\mathfrak{d}\nu_1}{\mathfrak{d}\mu}\dfrac{\mathfrak{d}\mu}{\mathfrak{d}\nu_2} = \dfrac{\mathfrak{d}\nu_1}{\mathfrak{d}\nu_2}$. Going from general to particular divergence, $\phi(t) = t\ln t$ for Kullback-Leibler divergence, so

$$\mathsf{D}^a_{KL}(\nu_1,\nu_2:\mu) = (C_a)\int \frac{\mathfrak{d}\nu_1}{\mathfrak{d}\mu}\ln\left(\frac{\mathfrak{d}\nu_1}{\mathfrak{d}\nu_2}\right)\mathrm{d}\mu$$

with the argument modification $\dfrac{\mathfrak{d}\nu_2}{\mathfrak{d}\mu}\dfrac{\mathfrak{d}\nu_1}{\mathfrak{d}\nu_2}\ln\left(\dfrac{\mathfrak{d}\nu_1}{\mathfrak{d}\nu_2}\right) = \dfrac{\mathfrak{d}\nu_1}{\mathfrak{d}\mu}\ln\left(\dfrac{\mathfrak{d}\nu_1}{\mathfrak{d}\nu_2}\right)$, again according to Lemma 1 ii) and iii). Because of Proposition 3, it can be shortened for symmetric integral to the form

$$\mathsf{D}^s_{KL}(\nu_1,\nu_2) = (C_s)\int \ln\left(\frac{\mathfrak{d}\nu_1}{\mathfrak{d}\nu_2}\right)\mathrm{d}\nu_1,$$

where there is no need to use the original reference measure. The last two integrals are the final forms for Kullback-Leibler divergence in the fuzzy case. Comparing them to fuzzy entropy formulas (3) and (4), it can be easily seen that the entropy is obtained as their special case.

Lemma 2. *Let us assume fuzzy measures μ,ν_1,ν_2 and existence of all the necessary CRN derivatives. Then, there exists a relation between entropy and Kullback-Leibler divergence in the form*

$$\mathsf{Ent}^a_{\mu}(\nu_1) = -\mathsf{D}^a_{KL}(\nu_1,\nu_2:\mu),$$
$$\mathsf{Ent}^s_{\mu}(\nu_1) = -\mathsf{D}^s_{KL}(\nu_1,\nu_2),$$

for $\nu_2 = \mu$, so taking the second input measure as the reference measure in both cases.

2.4 Other types of entropy

Besides the already mentioned entropy derived from original Shannon's concept in the information theory, there also exist some other types modified to particular applications. We focus here on the two most common Tsallis entropy and Rényi entropy, which are parametrised entropies with parameter $q \in \mathbb{R}$. In the additive setup, taking $q \to 1$ leads to Shannon entropy in both cases, so they are its generalised versions. As explained in [5], the generalisation provides information about the importance of specific events, for example, outliers or rare events.

Tsallis entropy can be directly modified into the nonadditive case with fuzzy measures and Choquet integral in the form

$$ {}^{T}_{q}\mathsf{Ent}_{\mu}(\nu) = \frac{1}{q-1}\left(1 - (C)\int \left(\frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\right)^{q} \mathrm{d}\mu\right), $$

where the existence of CRN derivative $\dfrac{\mathfrak{d}\nu}{\mathfrak{d}\mu}$ is assumed. Notation is taken to correspond the whole article with reference measure in the lower index, and to follow the usual notation with parameter q in the lower index and T referring to Tsallis in the upper index. Because the integrand is always nonnegative, only Choquet integral (without extensions to real functions) is sufficient to use and, together with Proposition 3, it leads to possible shortening of the formula given as

$$ {}^{T}_{q}\mathsf{Ent}_{\mu}(\nu) = \frac{1}{q-1}\left(1 - (C)\int \left(\frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\right)^{q-1} \mathrm{d}\nu\right). $$

Rényi entropy can be expressed in the fuzzy case similarly to Tsallis entropy as

$$ {}^{R}_{q}\mathsf{Ent}_{\mu}(\nu) = \frac{1}{1-q}\ln\left((C)\int \left(\frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\right)^{q} \mathrm{d}\mu\right), $$

again with assumed existence of the corresponding CRN derivative and the same explanation for the notation. Similarly, since integrand is nonnegative, Proposition 3 allows shortening the formula to

$$ {}^{R}_{q}\mathsf{Ent}_{\mu}(\nu) = \frac{1}{1-q}\ln\left((C)\int \left(\frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\right)^{q-1} \mathrm{d}\nu\right). $$

The formula could be further modified with Jensen-like inequality for Choquet integral and concave function. Assuming Choquet integral and convex function, it was already studied in [6] and from that it can be easily derived that assuming nondecreasing concave function (as is logarithm) the inequality holds with the opposite inequality sign. Since translation invariance needs to be satisfied, this is true only for asymmetric Choquet integral. Applying this inequality to the fuzzy Rényi entropy formula above leads to modifications given as

$$ {}^{R}_{q}\mathsf{Ent}_{\mu}(\nu) \geq \frac{1}{1-q}(C_a)\int \ln\left(\frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\right)^{q-1} \mathrm{d}\nu = -(C_a)\int \ln\left(\frac{\mathfrak{d}\nu}{\mathfrak{d}\mu}\right) \mathrm{d}\nu. $$

It is evident that changing order of the integral and logarithm causes that the integrand is no longer nonnegative and an extended Choquet integral need to be used, in this case only asymmetric Choquet integral because of Jensen inequality. Besides, the final form after applying inequality resembles the original definition of fuzzy entropy, and they coincide for nonnegative integrand.

3 Conclusions

The central notion of the article was differential entropy for fuzzy measures; its theoretical background was studied with proper definition, computational aspect and properties. From the practical point of view, introduced entropy can be used in all the application problems originally suitable for additive entropy. Thanks to the fuzzy measures, this entropy could better illustrate the situations where interactions are present. From the theoretical perspective, it is interesting to study other properties of entropy, which are mostly derived from the properties of CRN derivatives. It would also be beneficial to look at the maximum entropy principle in the fuzzy case. This principle is based on the premise that when estimating the probability distribution, you should select the distribution which leaves you the largest remaining uncertainty consistent with your constraints; that way no additional assumptions or biases are introduced into the calculations. However, the results are not so straightforward because even zero as a minimal entropy value does not hold assuming fuzzy measures.

References

1. Denneberg, D.: Non-additive measure and integral. vol 27. Springer Science & Business Media, (2013)
2. Grabisch, M.: Set functions, games and capacities in decision making. vol 46. Berlin: Springer, (2016)
3. Graf, S.: A Radon-Nikodym theorem for capacities. Journal für die reine und angewandte Mathematik **1980**(320), 192–214 (1980)
4. Huber, P., Strassen, V.: Minimax tests and the Neyman-Pearson lemma for capacities. Annals of Statistics, 251–263 (1973)
5. Maszczyk, T., Duch, W.: Comparison of Shannon, Renyi and Tsallis entropy used in decision trees. In Artificial Intelligence and Soft Computing–ICAISC 2008: 9th International Conference Zakopane, Poland, 11295–11314 (2008)
6. Ontkovičová, Z., Kisel'ák, J.: A way to proper generalization of ϕ-divergence based on Choquet derivatives. Soft Computing, **26**(21), 11295–11314 (2022)
7. Ontkovičová, Z., Kisel'ák, J., Hutník, O.: On quadruplets of nonadditive integrals. Fuzzy Sets and Systems, **451**, 297–319 (2022)
8. Shannon, C.: A mathematical theory of communication. The Bell system technical journal, **27**(3), 379–423 (1948)
9. Sugeno, M.: A note on derivatives of functions with respect to fuzzy measure. Fuzzy Sets and Systems, **222**, 1–17 (2013)
10. Sugeno, M.: A way to Choquet calculus. IEEE Transactions on Fuzzy Systems, **23**(5), 1439–1457 (2014)
11. Torra, V.: Entropy for non-additive measures in continuous domains. Fuzzy Sets and Systems, **324**, 49–59 (2017)
12. Wehrl, A.: General properties of entropy. Reviews of Modern Physics, **50**(2), 221–260 (1978)

Overall Fuzzy Weight of Alternatives for Partial Inner Dependence AHP

Shin-ichi Ohnishi* and Takahiro Yamanoi**

*Faculty of Engineering, Hokkai-Gakuen University,
**Graduate School of Health Scis., Hokkaido University,
{ohnishi, yamanoi}@hgu.jp

Abstract. Saaty's method, Analytic Hierarchy Process (AHP), is a popular process in the multi-criterion decision making. The Hierarchy used in AHP usually has an object, some criteria and alternatives (over 3 levels). In the criteria and alternatives level, plural elements are contained. Each elements in a same level must be independent enough in classical original AHP. For cases in which elements in a levels are not sufficiently independent, the extended inner dependence AHP is useful. We treat the "partial" inner dependence AHP, i.e. only some elements in a level are not independent. On the other hand, in many examples, data of AHP do not have the reliability, because data matrix themselves are often inconsistent. We first analyze sensitivity of consistency and weight, then attempt to present a weight using results from sensitivity analyses and a concept of fuzzy set when the data matrix is not sufficiently reliable in partial inner dependence AHP.

Keywords: decision-making, Analytic Hierarchy Process (AHP), sensitivity analysis, fuzzy set, inner dependence structure.

1 Introduction

The Analytic Hierarchy Process (AHP) proposed by T.L. Saaty in 1977 [1][2] is widely used in decision making for selecting alternatives. It is useful for the system containing humans, because it can reflects humans feelings naturally. A classical AHP assumes independence among both criteria and alternatives, although it is difficult to choose enough independent elements. Inner dependence method AHP [3] is used to solve this problem even for criteria or alternatives having dependence. The inner dependence AHP employs influence and dependency matrices to solve this problem. In this paper, we treat the "partial" inner dependence AHP, i.e. only some elements are not independent.

When AHP or inner dependence AHP is used, some comparison matrices may not have enough consistency because, for instance, a hierarchical structure may contain too many criteria or alternatives for decision making. In such a case, answers from decision-makers, i.e., comparison matrix components, do not have enough reliability, since they are too ambiguous or too fuzzy [4]. To avoid this problem, we usually have to revise again or abandon the data, but it takes a lot of time and costs [2][3]. Then, we consider that weights should also have ambiguity or fuzziness. Therefore, it is necessary

to represent these weights using fuzzy sets. Then fuzzy weights using results of sensitivity analyses are proposed. They are efficient even if hierarchy do not have enough independence in some levels

Our research first applied sensitivity analyses [5] to AHP to analyze how much the components of a pairwise comparison matrix influence the weights or consistency of a matrix [6]. This may enable us to show the magnitude of fuzziness in weights. We previously proposed new representation for criteria and alternatives weights in AHP, also representation for criteria weights for inner dependence, as L-R fuzzy numbers [7]. In the next step, we deal with partial inner dependence structure and consider compositions of weights to obtain overall alternative weights for partial inner dependence structure AHP, using results from sensitivity analyses and fuzzy operations when a comparison matrix does not have enough consistency.

In section 2, we introduce the AHP methodology and its inner dependence method. The sensitivity analyses for AHP are described in section 3. The calculation of fuzzy weight in partial inner dependence AHP are defined in section 4, and section 5 is conclusions.

2 AHP Methodology and Inner dependence Structure

2.1 Process of Classical AHP

In this section, we introduce the process of classical AHP and consistency index proposed by Saaty [1][2]

(Process 1) Representation of structure by a hierarchy. The problem under consideration can be represented in a hierarchical structure. The highest level of the hierarchy consists of a unique element that is the overall objective. At the lower levels, there are multiple criterion (i.e. elements within a single level) with relationships among elements of the adjacent higher level to be considered. The criterion are evaluated using subjective judgments of a decision maker. Elements that lie at the upper level are called parent elements while those that lie at lower level are called child elements. Alternative elements are put at the lowest level of the hierarchy

(Process 2) Paired comparison between elements at each level. A pairwise comparison matrix A is created from a decision maker's answers. Let n be the number of elements at a certain level. The upper triangular components of the comparison matrix a_{ij} ($i< j$ = 1,…,n) are 9, 8, .. , 2, 1, 1/2, …, or 1/9. These denote intensities of importance from element i to j. The lower triangular components a_{ji} are described with reciprocal numbers as follows

$$a_{ji} = 1/a_{ij} \tag{1}$$

In addition, for diagonal elements, let a_{ii} = 1. The lower triangular components and diagonal elements are occasionally omitted from the written equation as they are evident if upper triangular components are shown. The decision maker should make $n(n\text{-}1)/2$ paired comparisons at a level with n elements.

(Process 3) Calculations of weight at each level. The weights of the elements, which represent grade of importance among each element, are calculated from the pairwise comparison matrix. The normalized eigenvector that corresponds to a positive maximum eigenvalue of the matrix is used in calculations throughout in this paper.

(Process 4) Priority of an alternative by a composition of weights. The composite weight can be calculated from the weights of one level lower. With repetition, the weights of the alternative, which are the priorities of the alternatives with respect to the overall objective, are finally found.

2.2 Consistency

Since components of the comparison matrix are obtained by comparisons between two elements, coherent consistency is not guaranteed. In AHP, the consistency of the comparison matrix A is measured by the following consistency index (C.I.)

$$\text{C.I.} = \frac{\lambda_A - n}{n-1}, \tag{2}$$

where n is the order of matrix A, and λ_A is its maximum eigenvalue.

It should be noted that $\text{C.I.} \geq 0$ holds. And if the value of C.I. becomes smaller, then the degree of consistency becomes higher, and vice versa. The comparison matrix is consistent if the following inequality holds.

$$\text{C.I.} \leq 0.1 \tag{3}$$

2.3 Inner Dependence Structure

The conventional AHP ordinarily assumes independence among criteria and alternatives, although it is difficult to choose sufficiently independent elements in practice. This dependency indicates some kind of interaction among the elements. An inner dependence AHP [3] is used to solve this type of problem even when criteria have dependency.

In the method, using a dependency matrix $F=\{ f_{ij} \}$, we can calculate real weights $\boldsymbol{w}^{(N)}$ as follows,

$$\boldsymbol{w}^{(N)} = F\boldsymbol{w} \tag{4}$$

where $\boldsymbol{w}$ is weights from independent criteria or alternatives, i.e. normal weights of classical AHP, F consists of eigenvectors of influence matrices showing dependency among criteria or alternatives.

However, the inner dependence method requires a dependency matrix for all elements, even if some criteria are independent. In this research, we employ a partial inner dependence structure, i.e. only some elements are not independent, that enables us to easily understand the relationships among elements. In partial inner dependence AHP, we can divide an element set $A=\{X_1, X_2, ..., X_n\}$ into 2 subsets, the dependent part $A_1=\{X_1^{(1)}, X_2^{(1)}, ..., X_{n1}^{(1)}\}$ and independent part $A_2=\{X_1^{(2)}, X_2^{(2)}, ..., X_{n2}^{(2)}\}$, where $n_1+n_2=n$. The parts consist of elements that are dependent and independent criteria, respectively. Let the weights of A_1 be $\boldsymbol{w}^{(1)}=(w_{k_1}^{(1)}),\ k_1=1,\ldots,n_1$, and the weights of A_2 be $\boldsymbol{w}^{(2)}=(w_{k_2}^{(2)}),\ k_2=1,\ldots,n_2$.

3 Sensitivity Analyses of AHP

When AHP is used, the comparison matrix is often inconsistent or large differences among the overall weights of the alternatives do not appear. Thus, it is very important to investigate how the components of a pairwise comparison matrix influence the consistency or weights. Sensitivity analysis is used to analyze how results are influenced when certain variables change. Therefore, it is necessary to establish a sensitivity analysis of AHP.

In our research, a previously proposed method [7] is used to evaluate the fluctuation of the consistency index and weights when a comparison matrix is perturbed. This method is useful as it does not change the structure of the data.

Evaluating the consistency index and the weights of a perturbed comparison matrix are performed as follows.

(1) Perturbations $\varepsilon a_{ij}d_{ij}$ are imparted to component a_{ij} of a comparison matrix, and the fluctuation of the consistency index and the weight are expressed by the power series of ε.

(2) Fluctuations of the consistency index and the weights are represented by the linear combination of d_{ij}.

(3) By the coefficient of d_{ij}, it can be shown that how the component of the comparison matrix gives influence on the consistency index and the weight.

Since the pairwise comparison matrix A is a positive square matrix, the following Perron- Frobenius theorem [4] holds.

Theorem 1 (Perron – Frobenius) *For a positive square matrix A, the following holds true.*

1. *Matrix A has a positive eigenvalue. If λ_A is the largest eigenvalue then λ_A is a simple root. The positive eigenvector $\boldsymbol{w}$, corresponding to λ_A, exists. λ_A is called the Frobenius root of A.*
2. *Any positive eigenvectors of A are the constant multiples of $\boldsymbol{w}$.*
3. *The absolute value of the eigenvalues of A, except for λ_A, is smaller than λ_A.*
4. *The Frobenius root of the transposed matrix A' is equivalent to the Frobenius*

root of A.

This theorem ensures the existence of a weight vector in a pairwise comparison matrix.

From Theorem 1, the following theorem regarding a perturbed comparison matrix holds true [7].

Theorem2 *Let* $A = (a_{ij})$, $i,j = 1,\ldots,n$ *be a comparison matrix and let* $A(\varepsilon) = A+\varepsilon D_A$, $D_A=(a_{ij}d_{ij})$ *be a matrix that has been perturbed. Moreover, let* λ_A *be the Frobenius root of A with* $\boldsymbol{w}_1$ *being the corresponding eigenvector. Let* $\boldsymbol{w}_2$ *be the eigenvector corresponding to the Frobenius root of transposed matrix A', then, the Frobenius root* $\lambda(\varepsilon)$ *of* $A(\varepsilon)$ *and the corresponding eigenvector* $\boldsymbol{w}_1(\varepsilon)$ *can be expressed as follows*

$$\lambda(\varepsilon) = \lambda_A + \varepsilon\lambda^{(1)} + o(\varepsilon), \tag{5}$$

$$\boldsymbol{w}_1(\varepsilon) = \boldsymbol{w}_1 + \varepsilon\boldsymbol{w}^{(1)} + \boldsymbol{o}(\varepsilon), \tag{6}$$

where

$$\lambda^{(1)} = \frac{\boldsymbol{w}_2' D_A \boldsymbol{w}_1}{\boldsymbol{w}_2' \boldsymbol{w}_1}, \tag{7}$$

$\boldsymbol{w}^{(1)}$ *is an n-dimension vector that satisfies*

$$(A - \lambda_A I)\boldsymbol{w}^{(1)} = -(D_A - \lambda^{(1)} I)\boldsymbol{w}_1, \tag{8}$$

where $\boldsymbol{o}(\varepsilon)$ *denotes an n-dimension vector in which all components are* $o(\varepsilon)$.

Proof of this theorem can be found in Ohnishi's paper [7].

3.1 Sensitivity analysis for consistency index

Regarding a fluctuation of the consistency index, the following corollary can be obtained from Theorem 2.

Corollary 1 *Using an appropriate* g_{ij}*, we can represent the consistency index* C.I.(ε) *of the perturbed comparison matrix as follows*

$$\text{C.I.}(\varepsilon) = \text{C.I.} + \varepsilon \sum_{i}^{n} \sum_{j}^{n} g_{ij} d_{ij} + o(\varepsilon). \tag{9}$$

(Proof)

From the definition of the consistency index (4) and (5),

$$\mathrm{C.I.}(\varepsilon) = \mathrm{C.I.} + \varepsilon \frac{\lambda^{(1)}}{n-1} + o(\varepsilon).$$

Let $\boldsymbol{w}_1=(w_{1i})$ and $\boldsymbol{w}_2=(w_{2i})$ from (7). $\lambda^{(1)}$ is can now be represented as

$$\lambda^{(1)} = \frac{1}{\boldsymbol{w}_2' \boldsymbol{w}_1} \sum_i^n \sum_j^n w_{2i} a_{ij} w_{1j} d_{ij},$$

therefore, the second part of the right side is expressed by a linear combination of d_{ij}. (Q.E.D)

g_{ij} in equation (9) in Corollary 1 shows the influence of comparison matrix components on the consistency.

On the other hand, since the comparison matrix $A(\varepsilon) = (a_{ij}(\varepsilon))$ is reciprocal, then

$$d_{ji} = -d_{ij} \tag{10}$$

is obtained. The impact on the consistency can be easily shown by use of this property.

3.2 Sensitivity analysis for weights

With regards to the fluctuation in weighs, the following corollary can also be obtained from Theorem 2.

Corollary 2 *Using an appropriate $h_{ij}^{(k)}$, we can represent the fluctuation $\boldsymbol{w}^{(1)}=(w_k^{(1)})$ of the weight* (*i.e. the eigenvector corresponding to the Frobenius root*) *as follows*

$$w_k^{(1)} = \sum_i^n \sum_j^n h_{ij}^{(k)} d_{ij} \tag{11}$$

(Proof)

The *k*-th row component of the right side of (7) in Theorem 2 is represented as

$$\sum_i^n \sum_j^n \{ \frac{w_{1k} w_{2i} a_{ij} w_{1j}}{\boldsymbol{w}_2' \boldsymbol{w}_1} - \delta(i,k) a_{ij} w_{1j} \} d_{ij},$$

and is expressed by a linear combination of d_{ij}. Here,$\delta(i,k)$ is Kronecker's symbol

$$\delta(i,k) = \begin{cases} 1 & (i=k), \\ 0 & (i \neq k). \end{cases}$$

In contrast, since λ_A is a simple root, Rank(A-$\lambda_A I$) = n-1. Accordingly, the weight vector is normalized as

$$\sum_{k}^{n}(w_k + \varepsilon w_k^{(1)}) = \sum_{k}^{n} w_k = 1,$$

Then the condition is as follows.

$$\sum_{k}^{n} w_k^{(1)} = 0. \tag{12}$$

By using an elementary transformation to formula (8) in the condition above, we also can represent $w_k^{(1)}$ by linear combinations of d_{ij}. (Q.E.D)

As seen in equation (6) in Theorem 2, the component that has a great influence on weight $\boldsymbol{w}_1(\varepsilon)$ is the component which has the greatest influence on $\boldsymbol{w}^{(1)}$. $h_{ij}^{(k)}$ in equation (11) from Corollary 2 shows how the influence by the components of a comparison matrix on the weights can be calculated.

The influence can also be shown easily by use of equation (10).

4 Fuzzy Weight Representation

The comparison matrix often has poor consistency (i.e. 0.1<C.I.<0.2) because it encompasses too many criteria or alternatives. In these cases, the components of a comparison matrix are considered to have fuzziness since they result from the fuzzy judgment of humans. Therefore, weights should be treated as fuzzy numbers.

4.1 Fuzzy Weight of Criteria or Alternatives in Classical AHP

From the fluctuation of the consistency index, the multiple coefficient $g_{ij}h_{ij}^{(k)}$ in Corollaries 1 and 2 is considered as the influence on a_{ij}.

Since g_{ij} is always positive, if the coefficient $h_{ij}^{(k)}$ is positive, the real weight of criterion k is considered to be larger than w_{1k}. Conversely, if $h_{ij}^{(k)}$ is negative, the real weight of element k is considered to be smaller. Therefore, the sign of $h_{ij}^{(k)}$ represents the direction of the fuzzy number spread. The absolute value $g_{ij}|h_{ij}^{(k)}|$ represents the size of the influence. On the other hand, if C.I. becomes bigger, then the judgment becomes fuzzier.

Consequently, multiple C.I. $g_{ij}|h_{ij}^{(k)}|$ can be regarded as a spread of a fuzzy weight $\tilde{w}_k$ concerned with a_{ij}.

Definition 1 (fuzzy weight) *Let w_k be a crisp weight of element k, and $g_{ij}\ |h_{ij}^{(k)}|$ denote the coefficients found in* Corollaries 1 and 2. *If* 0.1<C.I.<0.2, *then a fuzzy weight $\tilde{w}_k$ is defined by*

$$\tilde{w}_k = (w_k, \alpha_k, \beta_k)_{LR} \tag{13}$$

where

$$\alpha_k = \text{C.I.}\sum_i^n \sum_j^n s(-, h_{kij}) g_{ij} \mid h_{ij}^{(k)} \mid, \tag{14}$$

$$\beta_k = \text{C.I.}\sum_i^n \sum_j^n s(+, h_{kij}) g_{ij} \mid h_{ij}^{(k)} \mid, \tag{15}$$

$$s(+, h) = \begin{cases} 1, (h \geq 0) \\ 0. (h < 0) \end{cases}, \quad s(-, h) = \begin{cases} 1, (h < 0) \\ 0. (h \geq 0) \end{cases}$$

Using same way, we can define fuzzy weight for inner dependence structure AHP.

Definition 2 (fuzzy weight for dependence AHP) *Let* $w^{(N)}{}_k$ *be a crisp weight of element k of inner dependence structure, and* g_{ij} $|h_{ij}{}^{(k)}|$ *denote the coefficients found in* Corollaries 1 and 2. *If* 0.1<C.I.<0.2, *then a fuzzy weight* $\tilde{w}_k^{(N)}$ *is defined by*

$$\tilde{w}_k^{(N)} = (w_k^{(N)}, \alpha_k, \beta_k)_{LR} \tag{16}$$

4.2 Overall Fuzzy Weights of Alternatives for Partial Inner Dependence

We consider about overall fuzzy weight of alternatives for partial inner dependence structure AHP. In this subsection, we assume inner dependence structure among criteria and partial inner dependence structure among alternatives.

Let $\boldsymbol{u}_k = (u_{kl}),\ (l = 1....., m)$ be weights of alternatives with only respect to criterion *k*, and divide the element set $A = \{X_1, X_2, ..., X_m\}$ into 2 subsets, the dependent part $A_1 = \{X_1^{(1)}, X_2^{(1)}, ..., X_{m1}^{(1)}\}$ and independent part $A_2 = \{X_1^{(2)}, X_2^{(2)}, ..., X_{m2}^{(2)}\}$, where $m_1 + m_2 = m$. Let the weights of A_1 be $\boldsymbol{u}_k^{(1)} = (u_{kl_1}^{(1)}),\ l_1 = 1, \ldots, m_1$, and the weights of A_2 be $\boldsymbol{u}_k^{(2)} = (u_{kl_2}^{(2)}),\ l_2 = 1, \ldots, m_2$, we calculate the modified weight of dependent subset $\boldsymbol{u}_k^{(N1)} = (u_{kl_1}^{(N1)})$ using dependency matrix F_A as follows:

$$\boldsymbol{u}_k^{(N2)} = F_A \boldsymbol{u}_k^{(2)}. \tag{17}$$

Then, the partial crisp (i.e., not yet fuzzy) weight $\boldsymbol{u}_k^{(PN)} = (u_{kl}^{(PN)})$, $k = 1, ..., n$ is made by the following concatenation.

$$\boldsymbol{u}_k^{(\mathrm{PN})} = (u_{kl}^{(PN)}) = (u_{k1}^{(\mathrm{N1})}, \ldots, u_{km_1}^{(\mathrm{N1})}, u_{k1}^{(2)}, \ldots, u_{km_2}^{(2)}) \tag{18}$$

Then, using definitions 1 and 2, we can make fuzzy local weight $\tilde{u}_{kl}^{(PN)}$ of alternative *l* from crisp weight $u_{kl}^{(PN)}$.

Let the modified fuzzy local weight of criteria, $\tilde{\boldsymbol{w}}^{(\mathrm{N})} = (\tilde{w}_k^{(\mathrm{N})})$, $k = 1, ..., n$, using dependency matrix F_C, at last, we can calculate fuzzy overall weights of alternative *l*, $\tilde{v}_l^{(\mathrm{N})}$ can be calculated as follows:

$$\tilde{v}_l^{(\mathrm{N})} = \sum_k^n \tilde{w}_k^{(\mathrm{N})} \otimes \tilde{u}_{kl}^{(\mathrm{PN})} \tag{19}$$

where $\otimes$ denotes fuzzy multiplication defined by extension principal.

5 Conclusions

When we use AHP, there are a lot of cases that data of AHP do not have enough consistency or reliability. For these cases, we propose fuzzy weight of alternatives using representation of fuzzy set and compositions for partial inner dependence AHP.

Our approach can show how to represent weights. And also it will be efficient to investigate how the result of partial inner dependence AHP has fuzziness even if data are not sufficiently consistent or reliable and element of hierarchy do not have enough independence in some levels.

References

1. Saaty, T. L.: A scaling method for priorities in hierarchical structures. J. Math. Psy., 15(3), 234—281 (1977)
2. Saaty, T. L.: The Analytic Hierarchy Process. McGraw-Hill, New York (1980)
3. Saaty, T. L.: Scaling the membership function. European J. of O.R., 25:320--329 (1986)
4. Saito, M.: An Introduction to Linear Algebra, Tokyo University Press (1966)
5. Tanaka, Y.: Recent advance in sensitivity analysis in multivariate statistical methods, J. Japanese Soc. Comp. Stat., 7(1):1--25 (1994)
6. Tone, K.: The Game Feeling Decision Making. Nikka-giren Press, Tokyo (1986)
7. Ohnishi, S., Imai, H., Kawaguchi, M.: Evaluation of a Stability on Weights of Fuzzy Analytic Hierarchy Process using a sensitivity analysis. J. Japan Soc. for Fuzzy Theory and Sys., 9(1), 140—147 (1997)
8. Ohnishi, S., Imai, H., Yamanoi, T.: Weights Representation of Analytic Hierarchy Process by use of Sensitivity Analysis, IPMU 2000 Proceedings (2000)
9. Dubois, D., Prade, H.: Possibility Theory An Approach to Computerized Processing of Uncertainty, Plenum Press, New York (1988)
10. Saaty, T. L.: Inner and Outer Dependence in AHP, University of Pittsburgh (1991)

11. Ohnishi, S., Yamanoi, T., Imai, H.: A Fuzzy Representation for Weights of Alternatives in AHP, New Dimensions in Fuzzy Logic and Related Technologies, Vol.II, 311--316 (2007)
12. Ohnishi, S., Dubois, D., Prade, H., Yamanoi, T.: A Fuzzy Constraint-based Approach to the Analytic Hierarchy Process, Uncertainty and Intelligent Information Systems, 2171--228 (2008)
13. Ohnishi, S., Yamanoi, T., Imai, H.: A Fuzzy Weight Representation for Inner Dependence AHP, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol.15, No.3, 329-335 (2011)
14. Ohnishi, S., Yamanoi, T. : On a Representation of Fuzzy Weight for Triple Inner Dependence AHP, MDAI2015 (2015)
15. Ohnishi, S., Yamanoi, T. : On a Fuzzy AHP Weight for Partial Inner Dependence Structure, Proceedings of DATAANAL2018 (2018)

Fusion functions based on a soft-penalty function*

Zdenko Takáč[1][0000−0003−0767−4756], Ľubomíra Horanská[1][0000−0002−6428−3439], Iosu Rodriguez-Martinez[2][0000−0002−9960−0203], and Humberto Bustince[2][0000−0002−1279−6195]

[1] Slovak University of Technology in Bratislava, Radlinskeho 9, 81237 Bratislava, Slovakia
[2] Public University of Navarre, Av. Cataluña s/n, 31006 Pamplona, Spain
zdenko.takac@stuba.sk

Abstract. In the context of information fusion, penalty-based functions have proven to be a powerful tool for selecting the best out of a set of possible reductions, in terms of minimizing the distance between the output and the input values. However, different reductions can help preserve different information, and the combination of their outputs can produce a more informed and useful final representative. In this work, we show how penalty functions can be used to combine different fusion functions, weighting the contribution of each of them according to the chosen penalty. We present an extension of penalty-based functions which we call wPA-functions and offer construction methods for tuning their behaviour. We illustrate the usefulness of the proposed functions in the context of an image classification problem, for reducing the features extracted by a Deep Learning model, obtaining favourable results.

Keywords: Penalty functions · Aggregation functions · Convolutional Neural Networks · Pooling functions.

1 Introduction

Data fusion is a crucial task in most areas of computational sciences [4–6, 13]. When working with multivalued data, fusion and aggregation functions can be used in order to reduce the available information into a single representative. However, choosing the best possible aggregation is usually a task-specific problem, and the used function ends up behaving as a hyperparameter of the problem.

Penalty functions can alleviate this problem, since they offer a measure of the spread between n values and another one [2]. They have been used with success for tasks such as decision making [3] and image downsampling [10], in order to find the best aggregation function to perform either of those tasks.

However, this strategy only takes into account one of the possible reductions and discards the contribution of the rest. Furthermore, it has been proven that

* Supported by the grants VEGA 1/0267/21, VEGA 1/0545/20 and Public University of Navarre.

each averaging aggregation function minimizes a particular penalty function, which limits the applicability of the method [2, 10].

We consider that the capability of measuring the suitability of a fusion function can be further exploited in order to combine the information of all reductions. Penalty functions could be used to measure the fitness of each of a set of different reductions, in order to assign more weight to the best outputs.

In this paper, we present wPA-functions, an extension of the concept of penalty based function which generates a convex combination of fusion functions, according to the extent that each of them minimize the chosen penalty. We present a construction method which allows to set the relationship between penalty value and the contribution of each fusion function, offering more drastic or soft behaviours.

Similar to how penalty-based functions have been succesful in image downsampling tasks, we show how both this functions and wPA-functions can improve the behaviour of Convolutional Neural Networks (CNNs) [7, 9]. This family of artificial neural networks perform image reduction in order to reduce the dimensionality of the features they extract from a given image. We have trained several variants of a well-known model of CNN and checked that the proposed family of function outperforms traditional techniques.

The remainder of this paper is structured as follows: Section 2 recalls some well-known concepts about fusion and aggregation functions; Section 3 presents the generalization of penalty-based functions and studies its main properties in depth; in Section 4, the suitability of the method is proven according to the performed experiments; finally, Section 5 closes the work with some conclusions and ideas for future research lines.

2 Preliminaries

We will denote vectors by $\mathbf{x} = (x_1, \ldots, x_n)$. In particular, we will make use of the vectors $\mathbf{0} = (0, \ldots, 0)$ and $\mathbf{1} = (1, \ldots, 1)$.

In this work we will focus on the information fusion task. The more general family of functions which can be used to this end are fusion functions. Let $n \in \{2, 3, \ldots\}$. An n-ary fusion function is an arbitrary function $A : [0, 1]^n \to [0, 1]$.

If we add the monotonicity property and bounding conditions to the previous functions we end up with the concept of aggregation function.

Definition 1. *An aggregation function is any function $A : [0, 1]^n \to [0, 1]$ which satisfies*

(A1) A is increasing;
(A2) $A(\mathbf{0}) = 0$ and $A(\mathbf{1}) = 1$.

Fuzzy integrals are examples of aggregation functions which model the interaction among subsets of the input values through fuzzy measures.

A fuzzy measure on N is a map $\nu : 2^N \to [0, \infty[$ such that $\nu(\emptyset) = 0$ and that, if $S \subseteq T \subseteq N$, then $\nu(S) \leq \nu(T)$.

Example 2. *Let σ be the permutation of $\{1,\dots,n\}$ such that $x_{\sigma(1)} \leq \dots \leq x_{\sigma(n)}$. We will denote by $N_i^\sigma = \{\sigma(i),\dots,\sigma(n)\}$. Then:*

- *The Choquet integral C_ν associated to the fuzzy measure ν is the map $C_\nu : \mathbb{R}^n \to \mathbb{R}$ given by $C_\nu(\mathbf{x}) = \sum_{i=1}^n x_{\sigma(i)}(\nu(N_i^\sigma) - \nu(N_{i+1}^\sigma))$. We set $x_{\sigma(0)} = 0$.*
- *The Sugeno integral S_ν associated to the fuzzy measure ν is the map $S_\nu : \mathbb{R}^n \to \mathbb{R}$ given by $S_\nu(\mathbf{x}) = \max_{i=1}^n(\min(x_{\sigma(i)}, \nu(N_i^\sigma))$.*
- *The Sugeno-like function D_ν associated to the fuzzy measure ν is the map $D_\nu : \mathbb{R}^n \to \mathbb{R}$ given by $D_\nu(\mathbf{x}) = \sum_{i=1}^n x_{\sigma(i)}\nu(N_i^\sigma)$.*

The requirement of monotonicity for aggregation functions leaves out several important fusion functions such as the mode. A more relaxed condition can be found in the concept of directional monotonicity.

Let $\mathbf{r}$ be a real n-dimensional vector, $\mathbf{r} \neq \mathbf{0}$. A fusion function $A : [0,1]^n \to [0,1]$ is $\mathbf{r}$-increasing ($\mathbf{r}$-decreasing) if for all $\mathbf{x} \in [0,1]^n$ and all $c > 0$ such that $\mathbf{x} + c\mathbf{r} \in [0,1]^n$, it holds that

$$A(\mathbf{x} + c\mathbf{r}) \geq A(\mathbf{x}) \qquad (A(\mathbf{x} + c\mathbf{r}) \leq A(\mathbf{x})).$$

Substituting property $(A1)$ in the definition of aggregation function by monotonicity in a direction $\mathbf{r}$, we end up with the concept of pre-aggregation function [1]

Definition 3. *A pre-aggregation function is a function $A : [0,1]^n \to [0,1]$ such that A is $\mathbf{r}$-increasing for some real vector $\mathbf{r} \in [0,1]^n$, $\mathbf{r} \neq \mathbf{0}$ and $A(\mathbf{0}) = 0$, $A(\mathbf{1}) = 1$.*

3 Fusion functions based on penalty function

3.1 *P*-functions

We say that a function $f : \mathbb{R}^n \to \mathbb{R}$ is quasi-convex in the k-th variable if

$$f\big(x_1,\dots,x_{k-1},\lambda u + (1-\lambda)v, x_{k+1},\dots,x_n\big) \leq \max\Big(f\big(x_1,\dots,x_{k-1},u,x_{k+1},\dots,x_n\big), f\big(x_1,\dots,x_{k-1},v,x_{k+1},\dots,x_n\big)\Big),$$

for all $\lambda \in [0,1]$ and all $x_1,\dots,x_{k-1},x_{k+1},\dots,x_n,u,v \in \mathbb{R}$.

Definition 4. *A function $P : [0,1]^{n+1} \to [0,\infty[$ is a penalty function if it satisfies:*

(P1) *$P(\mathbf{x},y) = 0$ if $x_i = y$ for all $i \in \{1,\dots,n\}$;*
(P2) *$P(\mathbf{x},y)$ is quasi-convex in y for any $\mathbf{x}$.*

Penalty functions can be used to construct fusion functions, some of them are aggregation functions, according to the following definition.

Definition 5. *Let $P : [0,1]^{n+1} \to [0,\infty[$ be a penalty function. Then the penalty based function f (P-function, for short) is*

$$f(\mathbf{x}) = arg \min_{y} P(\mathbf{x}, y)$$

if y is the unique minimizer, and $f(\mathbf{x}) = \frac{a+b}{2}$ if the set of minimizers is the interval [a, b].

3.2 *PA*-functions

Given k fusion (or aggregation) functions $A_i : [0,1]^n \to [0,1]$, a penalty function P can be used to choose the "best" fusion (or aggregation) function for given input vector $\mathbf{x} \in [0,1]^n$ in such a way that the fusion (or aggregation) function with the smallest penalty is applied.

Definition 6. *Let $P : [0,1]^{n+1} \to [0,\infty[$ be a penalty function and $A_i : [0,1]^n \to [0,1]$, for $i \in \{1,\ldots,k\}$, be fusion functions. Let $\mathbf{x} \in [0,1]^n$. Then the PA-function $f : [0,1]^n \to [0,1]$ defined by*

$$f(\mathbf{x}) = \frac{\sum\limits_{i \in M^{\mathbf{x}}} A_i(\mathbf{x})}{|M^{\mathbf{x}}|} \tag{1}$$

where $M^{\mathbf{x}} \subseteq \{1,\ldots,k\}$ is such that $j \in M^{\mathbf{x}}$ if and only if $P(\mathbf{x}, A_j(\mathbf{x}))$ is a minimum of the set $\{P(\mathbf{x}, A_1(\mathbf{x})), \ldots, P(\mathbf{x}, A_k(\mathbf{x}))\}$.

Proposition 7. *Under the assumptions of Definition 6 it holds $f(\mathbf{0}) = 0$ whenever $A_i(\mathbf{0}) = 0$ for all $i \in \{1,\ldots,k\}$, and $f(\mathbf{1}) = 1$ whenever $A_i(\mathbf{1}) = 1$ for all $i \in \{1,\ldots,k\}$.*

Proof: First observe that $M^{\mathbf{0}} = M^{\mathbf{1}} = \{1,\ldots,k\}$. Then the first equality follows from the consideration $P\big(\mathbf{0}, A_i(\mathbf{0})\big) = 0$ for all $i \in \{1,\ldots,k\}$ and the second from $P\big(\mathbf{1}, A_i(\mathbf{1})\big) = 1$ for all $i \in \{1,\ldots,k\}$. □

According to Proposition 7, each PA-function satisfies border conditions, however, even if all considered fusion functions $A_1,\ldots,A_k$ are increasing, the induced PA-function need not be increasing. Thus, taking aggregation functions $A_1,\ldots,A_k$, the induced PA-function need not be an aggregation function.

Proposition 8. *Under the assumptions of Definition 6, the PA-function f is idempotent whenever A_i is idempotent for each $i \in \{1,\ldots,k\}$ and f is averaging whenever A_i is averaging for each $i \in \{1,\ldots,k\}$.*

Proof: The idempotency as well as averagingness directly follows from Equation (1). □

Definition 9. *We say that a penalty function* $P : [0,1]^{n+1} \to [0,\infty[$ *is shift-invariant if, for all* $\mathbf{x} \in [0,1]^n$, $y \in [0,1]$, $r \in]0,1]$ *such that* $\mathbf{x} + r\mathbf{1} \in [0,1]^n$ *and* $y + r \in [0,1]$, *it holds*

$$P(\mathbf{x} + r\mathbf{1}, y + r) = P(\mathbf{x}, y).$$

Recall that a function $f : [0,1]^n \to [0,1]$ is shift-invariant if, for any $s \in [0,1]$, we have $f(\mathbf{x} + s\mathbf{1}) = f(\mathbf{x}) + s$ for all $\mathbf{x} \in [0,1]^n$ such that $\mathbf{x} + s\mathbf{1} \in [0,1]^n$.

Theorem 10. *Under the assumptions of Definition 6, the PA-function* f *is* $\vec{1}$*-increasing and shift-invariant whenever* $A_1, \ldots, A_k$ *and* P *are shift-invariant.*

Proof: First observe that from the shift-invariancy of a fusion function A_i and penalty function P it follows:

$$P(\mathbf{x} + r\mathbf{1}, A_i(\mathbf{x} + r\mathbf{1})) = P(\mathbf{x} + r\mathbf{1}, A_i(\mathbf{x}) + r) = P(\mathbf{x}, A_i(\mathbf{x})),$$

for all $\mathbf{x} \in [0,1]^n$ and $r \in]0,1]$ such that $\mathbf{x} + r\mathbf{1} \in [0,1]^n$. Thus, the order of penalties is preserved in the following sense:

$$P(\mathbf{x} + r\mathbf{1}, A_i(\mathbf{x} + r\mathbf{1})) \leq P(\mathbf{x} + r\mathbf{1}, A_j(\mathbf{x} + r\mathbf{1}))$$

whenever

$$P(\mathbf{x}, A_i(\mathbf{x})) \leq P(\mathbf{x}, A_j(\mathbf{x})).$$

It means that $M^{\mathbf{x}} = M^{\mathbf{x}+r\mathbf{1}}$ and consequently

$$f(\mathbf{x} + r\mathbf{1}) = \frac{\sum\limits_{i \in M} A_i(\mathbf{x} + r\mathbf{1})}{|M^{\mathbf{x}+r\mathbf{1}}|} = \frac{\sum\limits_{i \in M} (A_i(\mathbf{x}) + r)}{|M^{\mathbf{x}}|} = \frac{\sum\limits_{i \in M} A_i(\mathbf{x})}{|M^{\mathbf{x}}|} + r = f(\mathbf{x}) + r$$

which proves that f is shift-invariant. The proof of $\vec{1}$-increasingness is similar. □

Corollary 11. *Under the assumptions of Definition 6, the PA-function is a pre-aggregation function.*

3.3 *wPA*-functions

The definition of PA-function is based on the idea that we consider a set of fusion functions $A_1, \ldots, A_k$ and, for a given input $\mathbf{x}$, we choose the fusion function(s) with the smallest penalty, i.e., with the smallest value $P(\mathbf{x}, A_i(\mathbf{x}))$. In the following step, we change our approach in the sense that we do not choose only a single fusion function (or a few fusion functions) with the smallest penalty, but we consider all of them and assign them weights in such a way that the lesser the value $P(\mathbf{x}, A_i(\mathbf{x}))$ is, the greater weight is assigned.

Definition 12. *Let* $P : [0,1]^{n+1} \to [0,\infty[$ *be a penalty function and* $A_i : [0,1]^n \to [0,1]$*, for* $i \in \{1,\dots,k\}$*, be fusion functions. Let* $\mathbf{x} \in [0,1]^n$*. Let* $\mathbf{w}^{\mathbf{x}} = (w_1^{\mathbf{x}},\dots,w_k^{\mathbf{x}})$ *be a vector such that* $\sum_{i=1}^{k} w_i^{\mathbf{x}} = 1$ *and, for all* $i,j \in \{1,\dots,n\}$*,* $w_i^{\mathbf{x}} \le w_j^{\mathbf{x}}$ *whenever* $P(\mathbf{x},A_j(\mathbf{x})) \le P(\mathbf{x},A_i(\mathbf{x}))$*. Then the function* $f : [0,1]^n \to [0,1]$ *defined by*

$$f(\mathbf{x}) = \sum_{i=1}^{k} w_i^{\mathbf{x}} A_i(\mathbf{x}), \tag{2}$$

is called a wPA-function.

Remark 13. *Considering Definition 6 and Definition 12, it is easy to see that PA-function is a special case of wPA-function for the following weights:*

$$w_i^{\mathbf{x}} = \begin{cases} \frac{1}{|M^{\mathbf{x}}|}, & \text{if } i \in M^{\mathbf{x}}, \\ 0, & \text{if } i \notin M^{\mathbf{x}}. \end{cases}$$

Taking into account the assertion in Remark 13 we obtain that, in general, a wPA-function need not be an aggregation function (need not be increasing) even if A_i are aggregation functions (increasing) for all $i \in \{1,\dots,k\}$. However, as can be seen in the following proposition, the border conditions, idempotency and averagingness of functions A_i imply the same properties of the induced wPA-function.

Proposition 14. *Under the assumptions of Definition 12 it holds:*

1. *the PA-function* f *satisfies* $f(\mathbf{0}) = 0$ *whenever* $A_i(\mathbf{0}) = 0$ *for each* $i \in \{1,\dots,k\}$*, and* $f(\mathbf{1}) = 1$ *whenever* $A_i(\mathbf{1}) = 1$ *for each* $i \in \{1,\dots,k\}$*;*
2. *the PA-function* f *is idempotent whenever* A_i *is idempotent for each* $i \in \{1,\dots,k\}$*;*
3. *the PA-function* f *is averaging whenever* A_i *is averaging for each* $i \in \{1,\dots,k\}$*.*

Proof: The properties directly follow from Equation (2) and the properties of weighted averages. □

Theorem 15. *Under the assumptions of Definition 12, the wPA-function* f *is* $\vec{1}$*-increasing and shift-invariant whenever* $A_1,\dots,A_k$ *and* P *are shift-invariant.*

Proof: The proof is straightforward. □

Corollary 16. *Under the assumptions of Definition 12, the wPA-function is a pre-aggregation function.*

Remark 17. *In order to obtain a wPA-function* f *as given in Definition 12, the problem of obtaining a weighting vector* $\mathbf{w}^{\mathbf{x}}$ *should be solved. The natural procedure of solving this problem can be split into the following two steps:*

(i) Choose a decreasing function $g : [0,\infty[\to [0,\infty[$ *and calculate* $g\Big(P\big(\mathbf{x}, A_i(\mathbf{x})\big)\Big)$ *for all* $i \in \{1,\dots,k\}$. *The results can be considered as non-normalized weights which are ordered in an opposite way as the corresponding penalties.*

(ii) Choose a normalization function $s : [0,\infty[^k \to [0,1]^k$, *i.e., a function which satisfies* $\sum_{i=1}^{k} \big(s(\mathbf{z})\big)_i = 1$ *for all* $\mathbf{z} = (z_1,\dots,z_k) \in [0,\infty[^k$, *and* $\big(s(\mathbf{z})\big)_i \leq \big(s(\mathbf{z})\big)_j$ *whenever* $z_i \leq z_j$. *Apply the function* s *to the result of step (i) and calculate the weighting vector* $\mathbf{w}^{\mathbf{x}}$*:*

$$\mathbf{w}^{\mathbf{x}} = s\Big(g\Big(P\big(\mathbf{x}, A_1(\mathbf{x})\big)\Big),\dots,g\Big(P\big(\mathbf{x}, A_k(\mathbf{x})\big)\Big)\Big). \tag{3}$$

So we obtained a normalized weighting vector $\mathbf{w}^{\mathbf{x}}$ *whose coordinates are ordered in an opposite way as the corresponding penalties. Now, the* wPA*-function can be calculated:*

$$f(\mathbf{x}) = \sum_{i=1}^{k} \left(s\Big(g\Big(P\big(\mathbf{x}, A_1(\mathbf{x})\big)\Big),\dots,g\Big(P\big(\mathbf{x}, A_k(\mathbf{x})\big)\Big)\Big)\right)_i \cdot A_i(\mathbf{x}) \tag{4}$$

which can be simplified by denoting $p_i = P\big(\mathbf{x}, A_i(\mathbf{x})\big)$, *for all* $i \in \{1,\dots,k\}$, *as follows:*

$$f(\mathbf{x}) = \sum_{i=1}^{k} \left(s\Big(g\big(p_1\big),\dots,g\big(p_k\big)\Big)\right)_i \cdot A_i(\mathbf{x}). \tag{5}$$

4 Experimental Validation

In this section we will present an illustrative example of the suitability of wPA-functions in the context of feature downsampling for a Convolutional Neural Networks (CNN).

CNNs are a family of Deep Learning models which automate the process of extracting the most relevant spatial features of data where local information is relevant [9]. In particular, they have seen lots of popularity in the Computer Vision field as tools which can solve problems such as image classification [7] or image segmentation [12].

CNNs work in a sequential manner, outputting more complex representations of the data the deeper they become. The convolution operator which gives name to this family of neural networks is used to generate feature maps from the received input image at different points, in convolutional layers. At each layer, a series of 2D linear "filters" are convolved over all positions of the input image. Therefore, each convolution layer outputs several feature maps and, as a consequence, increases the dimensionality of the data.

In order to generate compact and informed representations of these features, CNNs need to downsample the generated feature maps. This process is usually taken care of in "pooling layers", which apply simple fusion functions such as

the arithmetic mean or the maximum to the values of disjoint submatrices of the feature image. In this work, we show that this operation can benefit from combining different fusion functions, and that wPA-functions are a direct tool to be applied to this end.

4.1 Dataset

We exemplify the suitability of the proposed method to an image classification task. The chosen dataset is the CIFAR-10 dataset [8] composed of 60000 small RGB images of the real world, representing animals or vehicles. There are 10 different possible classes each image can belong to, and the dataset is split into a train partition of 50000 samples, and a test partition with the remaining 10000 samples. Both partitions are balanced in terms of class proportion.

4.2 Deep Learning model

We have chosen to apply wPA-functions as the pooling operator of a RESNet model. RESNets are a particular example of CNN network which have gained lots of popularity thanks to their design, which allows to increase the number of layers of the model (and therefore be able to model more complex functions) without degrading their performance [7].

RESNets consists of several blocks of convolution layers (as well as additional layers which help the algorithm learning process), which are separated by downsampling operations. Thus, posterior blocks of the network work with smaller feature maps than previous ones. Although traditionally RESNets take care of these downsampling task through additional convolution layers, pooling layers can be used instead with no impact to the performance of the model.

We have trained several variants of a small RESNet20 model (i. e. a RESNet model with a total sum of 20 layers) in which we test different pooling functions and analize their different behaviours.

4.3 Pooling functions proposed

Several aggregation functions were applied as pooling operator in our model. In particular, the commonly used arithmetic mean and maximum have been tested. We have added the Choquet integral, the Sugeno integral and the $\mathbf{D}_\nu$ Sugeno-like integral.

Additionally, we combine the outputs of couples of these functions which include the arithmetic mean, using different PA and wPA-functions. We have tested the following combination of functions in order to gain a more in depth vision of the problem:

- **Penalty function**. We have tested the penalty function $P_1 : [0,1]^n \to [0,1]$ given by:
  - $P_1(\mathbf{x}, y) = \max_{i=1}^{n} |x_i - y|$

– **Weighting functions**. The following functions $g : [0,1] \to [0,1]$ assign a higher value to smaller penalty results:
  - $g_1(x) = 1 - x^{\lambda_g}$, with $\lambda_g \in \{1,2\}$
  - $g_2(x) = (1-x)^{\frac{1}{\lambda_g}}$, with $\lambda_g \in \{1,2\}$
– **Normalization functions**. The following functions $s : \mathbb{R}^k \to [0,1]^k$ guarantee that the resulting coefficients add up to 1, and therefore the resulting function is a convex combination of the chosen aggregation functions:
  - $\big(s_1(\mathbf{x})\big)_i = \frac{x_i^{\lambda_s}}{\sum_{l=1}^{k} x_l^{\lambda_s}}$, with $\lambda_s \in \{1,2\}$
  - $\big(s_2(\mathbf{x})\big)_i = \frac{\lambda_s^{x_i}}{\sum_{l=1}^{k} \lambda_s^{x_l}}$, with $\lambda_s \in \{1,2\}$

An alternative method for combining different aggregation functions in the pooling process was presented in [11]. In this paper, a linear combination of k aggregation functions $A_1, \ldots, A_k$ is used, in which the k coefficients of the combination are learnt through the optimization algorithm of the network. We compare our proposal to this method and normalize the resulting learnt coefficients in order to guarantee that the resulting combination is also a convex combination.

4.4 Experimental results

In this section we present the results of our experiments. Each of the variants of the model has been trained 5 times with different random initializations, leading to 5 different models. For each variant, we report the mean accuracy (ratio of correctly classified samples) of the 5 final models.

Results obtained with the individual aggregation functions are presented in Table 1. The Choquet integral is the function which performs better by itself, closely followed by the arithmetic mean. Despite its usual popularity, the maximum offers poor results for this model and dataset combination.

Table 2 summarizes results obtained with CombPool layers as well as PA-functions. It becomes clear that the combination of aggregation functions outperforms individual methods in most cases. Unsurprisingly, combining the arithmetic mean and the Choquet integral obtains good results, while combinations of the arithmetic mean with other fuzzy integrals seem to be equally promising.

Finally, the results obtained constructing wPA-functions through different combinations of parameters are shown in Table 3. Once again, combinations tend to improve upon individual functions, which proves the suitability of the method. In terms of comparability with the other combination techniques, there are cases for which wPA-functions outperform the remaining methods, while offering all around good results for some specific wPA-functions. In particular, the combination of g_1 weighting function with $\lambda_g = 1$ offers good results for all sets of aggregation functions.

Although similar in performance with CombPool layers, PA-functions and wPA-functions have the advantage of identifying the importance of each aggregation function without the need to learn additional parameters. This makes the method suitable in contexts where supervised learning is not an option, unlike plain CombPool layers.

Table 1. Mean results obtained for models which use individual aggregation functions

Individual functions				
AM	Max	C_ν	S_ν	D_ν
0.8581 ± 0.0010	0.8476 ± 0.0041	$\mathbf{0.8592 \pm 0.0019}$	0.8505 ± 0.0016	0.8567 ± 0.0027

Table 2. Mean results obtained for models which use PA-functions, as well as CombPool layers. CombPool layers learn the coefficients of the convex combination directly, using the optimization algorithm of the model. However, they are only applicable in contexts where training data is available, unlike PA-functions and wPA-functions.

P_1			
$AM + Max$	$AM + C_\nu$	$AM + S_\nu$	$AM + D_\nu$
0.8598 ± 0.0015	$\mathbf{0.8617 \pm 0.0032}$	0.8610 ± 0.0013	0.8602 ± 0.0042
CombPool layers			
$AM + Max$	$AM + C_\nu$	$AM + S_\nu$	$AM + D_\nu$
0.8559 ± 0.0010	0.8607 ± 0.0009	$\mathbf{0.8624 \pm 0.0004}$	0.8607 ± 0.0026

Table 3. Mean results obtained for models which use different combinations of aggregation functions, combined using different wPA-functions. The combination of penalty function P, weighting function g and normalization function s determines different valid wPA-functions. Results marked with an *slash* mean that that particular combination of wPA-function and aggregation functions leads to difficulties in the optimization method of the model and should be avoided.

wPA-function				Aggregation functions			
Penalty function	Weighting function	Normalization function		$AM + Max$	$AM + C_\nu$	$AM + S_\nu$	$AM + D_\nu$
P_1	g_1 $\lambda_g = 1$	s_1	$\lambda_s = 1$	0.8552 ± 0.0016	0.8618 ± 0.0009	0.8548 ± 0.0032	0.8599 ± 0.0058
			$\lambda_s = 2$	0.8560 ± 0.0036	$\mathbf{0.8634} \pm 0.0015$	0.8571 ± 0.0025	0.8604 ± 0.0042
		s_2	$\lambda_s = 1$	$\mathbf{0.8572 \pm 0.0038}$	0.8603 ± 0.0031	$\mathbf{0.8599 \pm 0.0023}$	0.8588 ± 0.0011
			$\lambda_s = 2$	0.8562 ± 0.0009	0.8602 ± 0.0009	0.8567 ± 0.0014	0.8608 ± 0.0027
	g_1 $\lambda_g = 2$	s_1	$\lambda_s = 1$	0.8549 ± 0.0007	0.8595 ± 0.0056	0.8557 ± 0.0013	0.8588 ± 0.0021
			$\lambda_s = 2$	0.8571 ± 0.0034	0.8609 ± 0.0062	0.8556 ± 0.0017	0.8571 ± 0.0023
		s_2	$\lambda_s = 1$	0.8549 ± 0.0051	0.8594 ± 0.0029	0.8579 ± 0.0021	0.8594 ± 0.0032
			$\lambda_s = 2$	0.8539 ± 0.0021	0.8601 ± 0.0039	0.8570 ± 0.0050	0.8607 ± 0.0019
	g_2 $\lambda_g = 1$	s_1	$\lambda_s = 1$	0.8569 ± 0.0063	0.8624 ± 0.0054	0.8563 ± 0.0015	0.8583 ± 0.0018
			$\lambda_s = 2$	0.8559 ± 0.0010	0.8591 ± 0.0020	0.8557 ± 0.0016	0.8583 ± 0.0055
		s_2	$\lambda_s = 1$	0.8556 ± 0.0034	0.8624 ± 0.0001	0.8584 ± 0.0032	0.8601 ± 0.0013
			$\lambda_s = 2$	0.8555 ± 0.0015	0.8605 ± 0.0011	0.8553 ± 0.0028	0.8597 ± 0.0013
	g_2 $\lambda_g = 2$	s_1	$\lambda_s = 1$	–	0.8605 ± 0.0071	0.8567 ± 0.0054	0.8592 ± 0.0059
			$\lambda_s = 2$	–	0.8605 ± 0.0010	0.8587 ± 0.0027	$\mathbf{0.8617 \pm 0.0016}$
		s_2	$\lambda_s = 1$	–	0.8577 ± 0.0025	0.8582 ± 0.0031	0.8565 ± 0.0006
			$\lambda_s = 2$	–	0.8615 ± 0.0008	0.8557 ± 0.0029	0.8575 ± 0.0027

5 Conclusion

In this work we have presented an extension of the concept of penalty-based function in the form of PA-functions and wPA-functions. We have shown that they can be useful when combining the reductions performed by several fusion and aggregation functions, and that the properties of the resulting functions depend on the sets of fusion functions selected.

In addition, we have exemplified the suitability of these functions in the feature downsampling process of a CNN network. We have proven that combining the reductions of different aggregation functions improves upon the classical pooling operators, and that PA-functions and wPA-functions correctly identify the most important values in a completely unsupervised way.

In the future we would like to explore additional construction methods for PA-functions and wPA-functions which may combine the output of different fusion functions through generic aggregation functions rather than weighted means. We would also like to test the suitability of wPA-functions empirically, learning the most optimal hyperparameters for their construction with supervised techniques, which we think could improve their performance even further.

References

[1] Bustince, H., Sanz, J., Lucca, G., Dimuro, G., Bedregal, B., Mesiar, R., Kolesárová, A., Ochoa, G.: Pre-aggregation functions: Definition, properties and construction methods. pp. 294–300 (2016). https://doi.org/10.1109/FUZZ-IEEE.2016.7737700

[2] Calvo, T., Beliakov, G.: Aggregation functions based on penalties. Fuzzy Sets and Systems **161**(10), 1420–1436 (2010). https://doi.org/10.1016/j.fss.2009.05.012

[3] Elkano, M., Galar, M., Sanz, J.A., Schiavo, P.F., Pereira Jr, S., Dimuro, G.P., Borges, E.N., Bustince, H.: Consensus via penalty functions for decision making in ensembles in fuzzy rule-based classification systems. Applied Soft Computing **67**, 728–740 (2018)

[4] Ferrero-Jaurrieta, M., Horanská, L., Lafuente, J., Mesiar, R., Dimuro, G.P., Takáč, Z., Gómez, M., Fernández, J., Bustince, H.: Degree of totalness: How to choose the best admissible permutation for vector fuzzy integration. Fuzzy Sets and Systems (2023)

[5] Fumanal-Idocin, J., Cordón, O., Bustince, H.: The krypteia ensemble: Designing classifier ensembles using an ancient spartan military tradition. Information Fusion **90**, 283–297 (2023)

[6] Grabisch, M., Marichal, J.L., Mesiar, R., Pap, E.: Aggregation Functions. Encyclopedia of Mathematics and its Applications, Cambridge University Press (2009)

[7] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[8] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

[9] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)

[10] Paternain, D., Jurio, A., Beliakov, G.: Color image reduction by minimizing penalty functions (2012). https://doi.org/10.1109/FUZZ-IEEE.2012.6250794

[11] Rodriguez-Martinez, I., Lafuente, J., Santiago, R.H., Dimuro, G.P., Herrera, F., Bustince, H.: Replacing pooling functions in convolutional neural networks by linear combinations of increasing functions. Neural Networks **152**, 380–393 (2022)

[12] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)

[13] Torra, V., Narukawa, Y., Sugeno, M.: Non-Additive Measures: Theory and Applications. Springer Publishing Company, Incorporated (2013)

Patients and clinicians preferences together in the loop for ADHD treatment recommendations

Oscar Raya[1][0000−0002−0420−3038], Xavier Castells[2][0000−0002−2619−7273], David Ramírez[2][0000−0003−0002−1189], and Beatriz López[1][0000−0001−9210−0073]

[1] Control engineering and intelligent systems (eXiT) research group, University of Girona, Spain {oscar.raya,beatriz.lopez}@udg.edu
http://exit.udg.edu
[2] Translab research group, Dept. of Medical Sciences, University of Girona, Spain
xavier.castells@udg.edu,ramsacdav@gmail.com

Abstract. Limitations of clinical practice guidelines regarding treatment recommendations include the lack of personalisation and user participation in the decision-making process. APPRAISE-RS has been developed to overcome such challenges by enabling users (both clinicians and patients) to express their preferences over treatments. However, APPRAISE-RS follows a heuristic that uses preferences to provide treatment information with five possible recommendations. This paper proposes the use of multiple criteria decision analysis (MCDA) as an extension of the existing APPRAISE-RS methodology, such that preferences enhance information about the best treatment by providing outcomes in an numerical interval, providing a finer ranking of final interventions. The experiments were conducted in the context of attention deficit hyperactivity disorder (ADHD).

Keywords: Clinical decision support systems · Multiple criteria decision making · Participatory medicine · ADHD.

1 Introduction

Participation, personalization, and evidence-based medicine are essential in making informed and effective treatment recommendations. Unfortunately, clinical practice guidelines (CPGs) often do not take into account patient participation neither provide personalized treatments for them, resulting in recommendations that may not fully align with the individual needs and preferences of patients. As a consequence, many patients may discontinue their prescribed treatment.

To address this issue, APPRAISE-RS [6] has been proposed as a solution that allows both patients and clinicians to express their preferences regarding treatment outcomes. APPRAISE-RS adapts the Cochrane/GRADE heuristic for formulating treatment recommendations. This heuristic follows these steps: scoring the relevance of the outcomes of interest using a 9 point-rating scale, selecting the outcomes that have been rated 7 or above, formulating a PICO (Patient, Intervention, Comparison and Outcome) treatment question, meta-analyzing those

studies that answer said question, assessing the risk-benefit relationship of the studied interventions and generating treatment recommendations accordingly which can be of 5 types: "strong in favour", "weak in favour", "weak in against", "strong against", and "no recommendation" for each studied treatment [1]. The limitations of this heuristic are that when answering the PICO question, only those outcomes or preferences that are deemed "critical" (score of 7 or above) are considered, and their weight within the heuristic is the same regardless of their score. Furthermore, this implementation of the GRADE heuristic gives equal weight to clinician and patient preferences, which might be unfair given the difference in clinical knowledge between them. Therefore, the objective of this work is to utilize multiple criteria decision analysis (MCDA) [4] to manage preferences "as-needed" and enhance the treatment recommendations.

MCDA allows for a more comprehensive analysis of drug recommendations compared to APPRAISE-RS. By assigning scores within the interval [0,1], we can provide a wider ranking compared to the five options provided by APPRAISE-RS. Two MCDA-based approaches are explored: a utility-based approach and a preference-based (using Borda) approach. In our first approach, we propose a strategy to separate positive and negative results to more effectively combine patient and clinician preferences. In our second approach, we use the Borda method [3] to emphasize and differentiate the contributions of the clinician's preferences with respect to the patient's preferences. These contributions improve the personalization and evidence-based nature of the recommendations.

The experiments are done in the field of attention deficit hyperactivity disorder (ADHD). The proposed MCDA-based approaches provide a solution to the limitations of APPRAISE-RS and contribute to the development of patient-centered care.

This paper is organized as follows. Section 2 provides a preliminary description on how APPRAISE-RS works. Next, in Section 3 the MCDA approaches are detailed. In Section 4, the results obtained with ADHD, including the comparison with the previous APPRAISE-RS approach, are provided. We end the paper in Section 5 with some conclusions and future work.

2 Background

This work is based on the methodology called APPRAISE-RS [6], which is a recommender system that automates, adapts, extends, and iterates the Grading of Recommendations Assessment, Development and Evaluation (GRADE) working group methodology [5,1]. The main purpose of APPRAISE-RS is to formulate automated, up-to-date, participatory, and personalized treatment recommendations based on an updated database of clinical studies. The process involves several steps, including gathering the patient's demographic and clinical data and preferences, filtering relevant studies from the database, generating scientific evidence through meta-analysis techniques, assessing the quality of evidence using a rule-based system, evaluating the benefit-risk relationship of each intervention with the use of a second rule-based system, and finally, generating

a clinical recommendation. APPRAISE-RS is currently being developed specifically for attention deficit hyperactivity disorder (ADHD) and requires two visits in the point-of-care scenario: The first visit provides information to the patient, while the second involves expressing preferences and obtaining treatment recommendations. The patient and the clinician then select the final intervention together. The goal of the methodology is to help, and not replace, clinical judgement and improve the validity of CPGs while reducing the rate of patients who fail to receive the most suitable care based on available evidence.

3 Methodology

This section explains the two MCDA approaches proposed in this paper to generate treatment recommendations. To use these approaches, inputs from APPRAISE-RS are required and must be used in a specific manner to ensure that the recommendations generated are useful for the MCDA approaches.

3.1 APPRAISE-RS for MCDA

Given the patient's demographic and clinical data, as well as the preferences of users (clinician and patient) preferences, APPRAISE-RS returns a single ranking of treatments.

Preferences are expressed by both patients and clinicians using a Likert scale [2] that ranges from 1 (not at all important) to 9 (very important). Preferences are related to treatment outcomes regarding efficacy and safety, including adverse events such as vomiting, somnolence, dizziness, etc. These outcomes are referred to as variables in APPRAISE-RS. Variables are used by APPRAISE-RS to select the studies in the literature that considered such outcomes, and which conclude in the convenience or not of a given treatment regarding the patient at hand. APPRAISE-RS analyze all the variables related to the preferences expressed by the users at once, providing a single ranking for all of them.

Analogously, if we provide APPRAISE-RS with a single preference, the recommendations will be related to a single variable. Since we have a total of 18 preferences per user, we need to run APPRAISE-RS 18 times, obtaining a total of 18 rankings. Each ranking include all the interventions available, each of which is labelled according to the five possible outcomes of APPRAISE-RS ("strong in favour", "weak in favour", "weak in against", "strong in against", and "no recommendation"). These 18 rankings are used as the input for the MCDA methods.

3.2 Utility Approach

The Utility Approach involves generating two rankings based on the recommendations obtained from APPRAISE-RS. Figure 1 illustrates the workflow methodology for this approach.

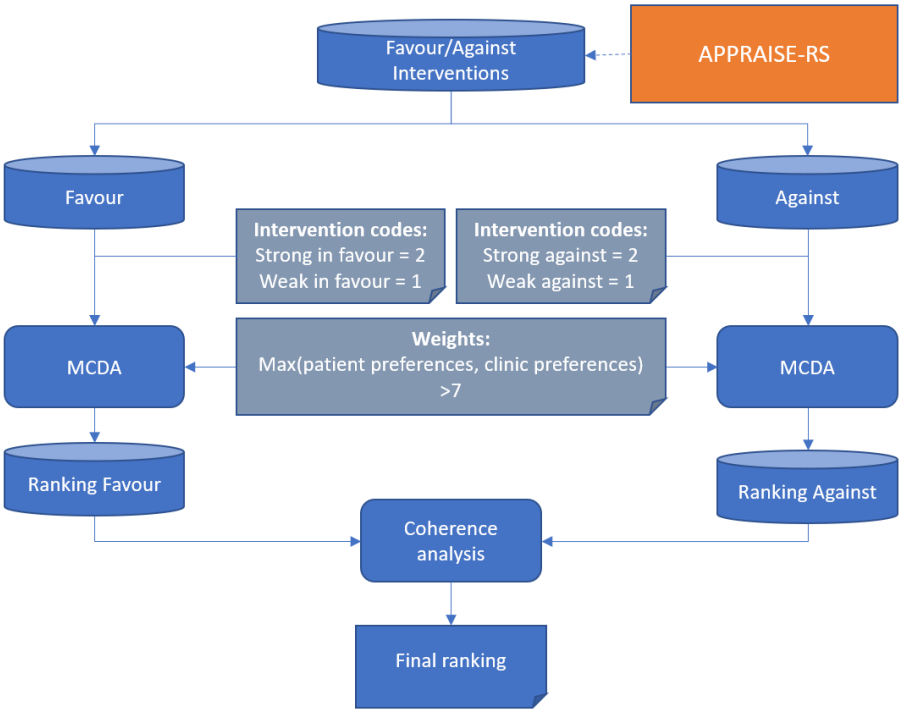


Fig. 1. Workflow methodology of the Utility Approach.

First, two MCDA are performed to generate two rankings. The "in favour" MCDA takes the treatments that fall in the "in favour" category as input, while the "in against" MCDA takes the treatments that fall in the "in against" category. The treatments are the alternatives being evaluated by the MCDA methods, and the different variables (e.g., side effects, efficacy) are the criteria used by the MCDA methods to evaluate and rank the treatments. The maximum values for each preference (from the clinician and patient) are used as weights for both MCDA. Only preferences with a value greater than or equal to 7 are considered, which is done to match the GRADE methodology used on APPRAISE-RS. However, preferences below a value of 7 could be taken into consideration.

The value of each treatment/criteria is coded according to the APPRAISE-RS recommendations. Recommendations categorized as "weak in favour" and "strong in favour" are given one and two points respectively in the "in favour" ranking, while recommendations categorized as "weak in against" and "strong in against" are given one and two points respectively in the "in against" ranking. The "no recommendation" result is not used in either ranking.

Finally, the two rankings are merged using a coherence analysis. Assuming that the MCDA method returns a numerical score for each possible alternative (treatment), this step involves subtracting the score obtained on the "in against"

ranking from the score obtained on the "in favour" ranking for each treatment. The resulting scores are then used to generate the final ranking.

The Utility Approach provides separate rankings for "in favour" (positive values) and "in against" (negative value) recommendations, which can be useful in certain decision-making contexts. It allows for the use of different methods, such as the Weighted Average (WA), but also other methods that only accept positive inputs, such as the Weighted Product (WP).

However, it does not explicitly consider the relative importance of preferences from the clinician and patient, which may result in overshadowing of one agent's preferences when the other agent gives higher weights to their own. As a result, the method may not always yield an optimal ranking of treatments that considers both the clinician and patient needs and preferences. Therefore, additional methods that fully account for the relative importance of preferences from both users may need to be employed.

3.3 Preference Approach

The Preference Approach involves separating the clinician preferences and the patient preferences in order to generate separated rankings. This approach builds upon the previous method, resulting in four initial rankings: "in favour" and "in against" rankings using clinician preferences, and "in favour" and "in against" rankings using patient preferences. Figure 2 shows the workflow methodology for this approach.

In the first step, we separate the "in favour" recommendations from the "in against" recommendations and assign codes as described on the previous method.

Next, we perform two MCDA for each group using clinician preferences in one ranking and patient preferences in the other. As in the previous method, preferences below a value of 7 are not taken into account.

Then, the two rankings within each group are merged using the Borda count [3,4]. This involves assigning scores to treatments based on their position in each ranking and summing the scores across rankings.

Finally, the "in favour" and "in against" rankings are merged using the coherence analysis described earlier.

This approach may result in different scores assigned to treatments compared to the previous method, resulting in potentially different final rankings. However, unlike the Utility Approach, the Preference Approach explicitly considers the relative importance of preferences from the clinician and patient by merging the rankings. Consequently, this approach is helpful as it provides a more comprehensive decision-making process that takes into account the preferences and needs of both the clinician and patient.

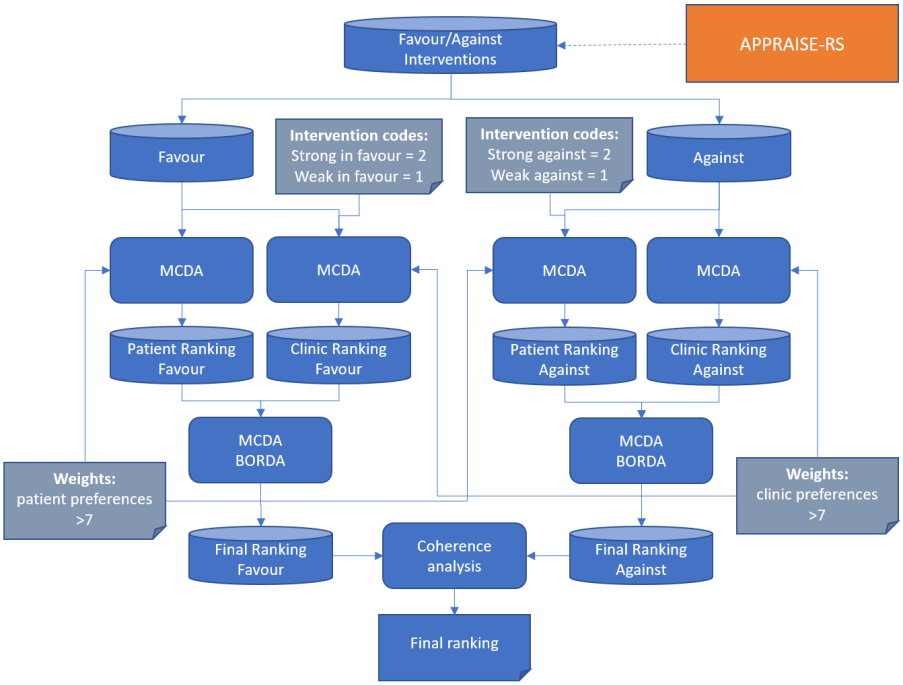


Fig. 2. Workflow methodology of the Preference Approach.

4 Results

The two MCDA approaches have been applied for ADHD intervention recommendations, and tested over the same 28 simulated patients used on APPRAISE-RS [6].

Three different scenarios has been considered for experimentation:

- Utility Approach versus Preference Approach: We use the Weighted Average (WA) as MCDA method to compare the two approaches presented in this paper, with the previous work APPRAISE-RS.
- Verification of the Utility Approach with WA: To verify that the Utility Approach, when using WA in the intermediate steps, should be equivalent to the WA in a single step (see Figure 3).
- Weighted product (WP): To show the flexibility of our approach to consider other MCDA methods that do not handle negative information.

Results are analyzed in terms of the number of treatment recommended, and the top recommendations, meaning that a more accurate ranking should provide fewer treatments at the top.

4.1 Utility Approach versus Preference Approach

Table 1 compares the two MCDA approaches, when using the Weighted Average (WA), with APPRAISE-RS based on the number of recommendations made to each patient. MCDA recommendations could be positive or negative, depending on the last step of the methods. The Table only shows the number of positive value recommendations for each patient. Analogously, only "in favour" recommendations of APPRAISE-RS are displayed in the Table. Additionally, the table presents the number of recommendations that match with those made by APPRAISE-RS and the number of interventions recommended in the first position by both approaches.

Patient	APPRAISE-RS	Utility Approach			Preference Approach		
		Number	Match	Top	Number	Match	Top
P1	7	21	6	2	23	5	1
P2	8	17	6	1	16	5	2
P3	4	21	4	1	3	1	1
P4	6	20	5	1	15	4	1
P5	1	21	0	1	11	0	1
P6	0	2	0	1	32	0	2
P7	0	3	0	2	31	0	1
P8	0	2	0	2	34	0	1
P9	0	1	0	1	35	0	1
P10	1	6	0	1	28	0	1
P11	1	9	1	1	32	1	1
P12	7	16	5	1	10	4	1
P13	0	22	0	1	27	0	2
P14	7	16	5	2	9	3	1
P15	8	23	6	1	14	3	1
P16	5	11	4	1	29	4	1
P17	1	7	0	1	32	0	1
P18	0	1	0	1	1	0	1
P19	1	1	0	1	34	1	32
P20	8	13	6	1	19	5	1
P21	5	10	3	1	24	2	1
P22	2	17	2	1	15	1	1
P23	2	20	2	1	9	0	1
P24	3	16	3	1	7	2	1
P25	2	14	0	1	30	2	2
P26	3	21	3	1	29	3	1
P27	1	8	1	1	33	1	1
P28	1	17	1	1	26	1	1
Median	2	15	1.5	1	25	1	1

Table 1. Comparison of the number of recommendations by patient of the two MCDA approaches against APPRAISE-RS.

It can be observed that the recommendations made by both MCDA approaches match those made by APPRAISE-RS for most patients. Moreover, the two MCDA approaches only provide 1 or 2 interventions as top recommendations, in contrast to APPRAISE-RS, which is assigning the same value to all interventions recommended in favour.

There is an exception for patient 19, who received only one recommendation from APPRAISE-RS and the Utility Approach, but was recommended 32 interventions in the first position by the Preference Approach.

4.2 Verification of the Utility Approach with Weighted Average (WA)

To verify the equivalence of the Utility Approach to traditional MCDA with Weighted Average (WA), we applied the latter approach using the recommendations obtained from APPRAISE-RS. As depicted in Figure 3, each recommendation from APPRAISE-RS is assigned a value ranging from -2 to 2, where positive values denote treatments recommended by APPRAISE-RS, and negative values denote those not recommended. Next, the MCDA method is applied in a similar way as in the Utility Approach.

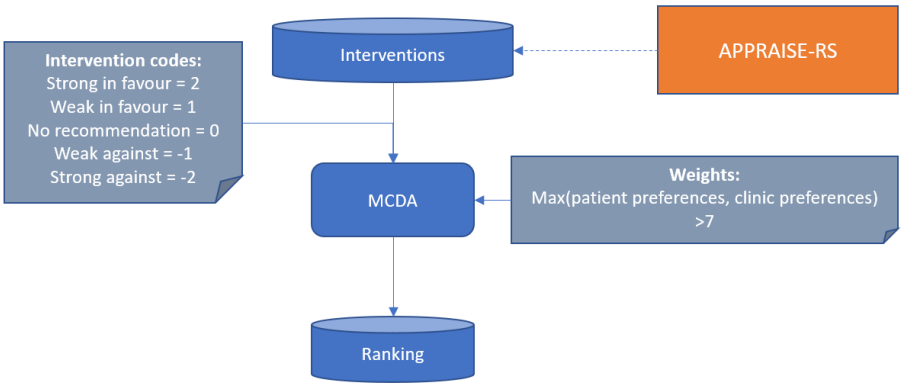


Fig. 3. Workflow methodology of the WA approach.

The results obtained from this approach were compared to those obtained using the Utility Approach when using WA on the intermediate steps, and we found that they were exactly the same. Thus, we concluded that the Utility Approach was indeed equivalent to using MCDA with a WA.

4.3 Weighted Product (WP)

To demonstrate the flexibility of our approach in accommodating other MCDA methods that are unable to handle negative information, we tested the Utility Approach using the Weighted Product (WP). The results showed that the interventions recommended using WP were identical to those recommended using WA, with only the order of treatment recommendations in the rankings varying. Consequently, the top-ranked treatment may differ between the two approaches. Table 2 displays the frequency at which each treatment intervention has been recommended as the first option. It is evident that the recommendations provided by the Utility Approach using WP are similar to those of the Utility Approach using WA, but differ from the recommendations of the other two approaches.

4.4 Discussion

Regarding the results, none of the three methods leaves any patient without a recommendation, which sets them apart from APPRAISE-RS (rule-based), which in many cases does not identify any treatment. As a notable context, clinicians also do not prescribe treatment in some patients, but clinical practice guidelines and the methods proposed in this paper always recommend some treatment.

It is interesting to note that the MCDA methods provide a more concise set of recommendations than APPRAISE-RS, as they recommend fewer treatments in the first position (Table 1).

However, the Utility Approach and Preference Approach demonstrate different characteristics in terms of the diversity of the recommended treatments. While the Utility Approach, when using WA or WP, provides a similar range of recommended treatments as APPRAISE-RS, the Preference Approach recommends a more diverse set of treatments in the top position, which enhances the personalization of recommendations (Table 2).

It is important to note that the Preference Approach is limited by the assumption that the preferences of the clinician and patient are contributing as if both users have the same expertise, which may not always be the case. Therefore, further development of the method may be needed to take into account the expertise of the user when managing the preferences from both the clinician and patient. In that regard [7] could be an interesting starting point.

5 Conclusions

This work presents two MCDA approaches for selecting medical treatments to overcome some limitations of current treatment recommendation methods that do not fully consider patient and clinician preferences. The experiments were carried out in ADHD and the results were compared with the previous approach APPRAISE-RS.

Treatment intervention	APPRAISE-RS	Utility Approach (WA)	Utility Approach (WP)	Preference Approach
atomoxetine high	12	5	5	3
atomoxetine low	10			
bupropion high				1
clomipramine				1
clonidine high				1
clonidine low				1
desipramine high				1
desipramine low				1
dexamphetamine high				1
dexamphetamine high and paroxetine				1
dexamphetamine low				1
dexmethylphenidate high		1	1	2
dexmethylphenidate low				1
guanfacine high	10	1	1	2
guanfacine low				1
lisdexamfetamine high	12			1
lisdexamfetamine low	6			1
memantine				1
methylphenidate high	14	2	2	2
methylphenidate high and clonidine low				1
methylphenidate low	11	16	17	15
methylphenidate low and clonidine high				1
methylphenidate low and nicotine				1
mixed amfetamine salts high		7	4	10
mixed amfetamine salts low				1
modafinil high	9			1
modafinil low				1
nicotine				1
paroxetine				1
pindolol				1
reboxetine				1
selegiline				1
serdexmethylphenidate low				1
viloxazine high				2
viloxazine low				1
Total	84	32	30	63

Table 2. Comparison of the number of times each treatment intervention is recommended as the first choice using four different methods: APPRAISE-RS, Utility Approach with Weighted Average (WA), Utility Approach with Weighted Product (WP), and Preference Approach.

Both approaches recommended a median of one treatment in the first position. There was a high degree of similarity in the treatments recommended by both MCDA approaches, with psycho-stimulants such as methylphenidate and amphetamine derivatives being the most recommended medications. Atomoxetine was recommended less frequently than in the previous work APPRAISE-RS. Importantly, both methods were able to provide recommendations for all patients, which is an improvement over APPRAISE-RS.

Based on our results, we suggest that the MCDA methods are useful tools for selecting ADHD treatments and can complement CPGs. The Borda method-based approach demonstrated the highest participation of clinician and patient preferences, indicating that it may be the most effective method for incorporating multiple perspectives into treatment decision-making.

Future research could involve adding another level of MCDA to weight clinician and patient preferences according to their expertise.

Acknowledgements

This work received funding from Fundació Pascual i Prats i el CampusSalut de la Universitat de Girona (AIN2018E), and joint funding from the European Regional Development Fund (ERDF), the Spanish Ministry of the Economy, Industry and Competitiveness (MINECO) and the Carlos III Research Institute, under grant no. PI19/00375. This work received support from the Generalitat de Catalunya 2021 SGR 01125.

We would like to thank Clara Martínez for his support in the initial implementation of some components of the prototype.

References

1. Alonso-Coello, P., Oxman, A., et al.: GRADE Evidence to Decision (EtD) frameworks: a systematic and transparent approach to making well informed healthcare choices. 2: Clinical practice guidelines. BMJ (Clinical research ed.) **353** (2016). https://doi.org/10.1136/BMJ.I2089
2. Derrick, B., White, P.: Comparing two samples from an individual likert question. International Journal of Mathematics and Statistics **18** (2017), `http://www.ceser.in/ceserp/index.php/ijms/article/view/4997`
3. Emerson, P.: The original borda count and partial voting. Social Choice and Welfare **40**(2), 353–358 (Feb 2013). https://doi.org/10.1007/s00355-011-0603-9
4. Greco, S., Figueira, J., Ehrgott, M.: Multiple criteria decision analysis, vol. 37. Springer (2016). https://doi.org/10.1007/978-1-4939-3094-4
5. Guyatt, G., Oxman, A.D., Akl, E.A., Kunz, R., Vist, G., Brozek, J., Norris, S., Falck-Ytter, Y., Glasziou, P., deBeer, H., Jaeschke, R., Rind, D., Meerpohl, J., Dahm, P., Schünemann, H.J.: Grade guidelines: 1. introduction—grade evidence profiles and summary of findings tables. Journal of Clinical Epidemiology **64**(4), 383–394 (2011). https://doi.org/10.1016/j.jclinepi.2010.04.026, `https://www.sciencedirect.com/science/article/pii/S0895435610003306`

6. López, B., Raya, O., et al.: Appraise-rs: Automated, updated, participatory, and personalized treatment recommender systems based on grade methodology. Heliyon (January 2023). https://doi.org/10.1016/j.heliyon.2023.e13074
7. Raya, O., Castells, X., Ramirez, D., Lopez, B.: Management of patient and physician preferences and explanations for participatory evaluation of treatment with an ethical seal. 20th International Conference on Artificial Intelligence in Medicine (AIME 2022). Accepted (2023)

An application to measure customers' interest on food waste reduction using hesitant terms

Walaa Abuasaker[1][0009−0000−4630−6687], Jennifer Nguyen[2][0000−0002−7498−7536], Núria Agell[2,⋆][0000−0001−9264−2147], Mónica Sánchez[1][0000−0002−0201−345X], and Francisco J. Ruiz[1][0000−0002−4314−101X]

[1] UPC-BarcelonaTech
{walaa.abuasaker,monica.sanchez,francisco.javier.ruiz},@upc.edu
[2] Esade Business School, URL
{jenniferthanhvan.nguyen,nuria.agell}@esade.edu

Abstract. In this study, we propose the use of hesitant term sets to capture the uncertainty of people's interest to food waste. A real case example, in which users assess their interest on the use of the "Too Good to Go" platform, shows the applicability of the proposed approach in a real context related to people's interest in the reduction of food waste. We propose a methodology, using unbalanced hesitant fuzzy linguistic term set (HFLTSs) to aggregate rating values. The objective is to fuse the opinions of different profiles of users when assessing an alternative in a complex context. The proposed approach allows considering different linguistic or ordinal scales to better capture human being's cognitive models. To this end, the concept of linguistic perceptual map and projections among perceptual maps are introduced to model and aggregate users' different profiles. HFLTSs provide a useful and powerful form to represent opinions in the decision-making process. An automated methodology which aggregates opinions across users' is defined based on rating values and perceptual maps. In addition, we compute a measure to capture the degree of consensus representing the agreement among the sentiment of users'.

Keywords: Decision making under uncertainty · Linguistic modeling · Unbalanced hesitant fuzzy linguistic term sets · Rating scales.

1 Introduction

Food waste is a huge problem for the humanity. More than one third of the food produced globally is lost or wasted [1]. This amount of food waste can be reduced by considering "localness" of food systems [2]. By promoting the purchase of unsold food products from both grocery stores and restaurants through online platforms, we can effectively reduce food waste [3]. The use of on-line platforms and apps to explore shops and restaurants in local area and sale surprise bags

⋆ Corresponding author

of surplus food from going to waste at a great price is considered as a new way to reduce food waste.

In this paper we consider an ordinal scale with unbalanced terms to measure the interest and expectations on the on-line platform "Too Good to Go" from different consumers' profiles. "Too Good to Go" is a mobile app that connects customers to restaurants and food stores that have unsold food, helping to achieve a better balance between economy, environment, and society. We consider different linguistic scales to better capture human being's cognitive models ([4], [5]). To this end, the concept of linguistic perceptual map and projections among perceptual maps are introduced to model and aggregate users' different profiles. HFLTSs provide a useful and powerful form to represent opinions in the decision-making process ([6], [7]). An automated methodology which aggregates opinions across users' is defined based on rating values and perceptual maps. In addition, we compute a measure to capture the degree of consensus representing the agreement among the sentiment of users' [8].

2 Preliminaries

This section contains the definitions of and some preliminary concepts on HFLTSs and linguistic perceptual maps based on [9] that are necessary for the methodology presented.

Let S be a totally ordered set of *basic linguistic terms* (BLTs), $S=\{s_1, \ldots, s_n\}$, with $s_1 < \ldots < s_n$ and we consider the concept of HFLTSs introduced by Rodriguez et al. in [7] to allow users to consider hesitancy in their opinions.

Definition 1. ([7]) *A hesitant fuzzy linguistic term set (HFLTS)* over S is a subset of consecutive BLTs of S, i.e., $\{x \in S \,|\, s_i \leq x \leq s_j\}$, for some $i, j \in \{1, \ldots, n\}$ with $i \leq j$. The HFLTS S is called the *full HFLTS*. Moreover, the empty set $\{\} = \emptyset$ is also considered as a HFLTS and it is called the *empty HFLTS*.

The non-empty HFLTS $\{x \in S \,|\, s_i \leq x \leq s_j\}$ denoted by $[s_i, s_j]$. If $i = j$, $[s_i, s_i]$ is the singleton $\{s_i\}$. The set of all non-empty HFLTSs over S is denoted by $\mathcal{H}_S$, that is, $\mathcal{H}_S = \{[s_i, s_j] : i, j \in \{1, \ldots, n\}, i \leq j\}$. In this way, the set of all HFLTSs over S is $\mathcal{H}_S \cup \{\emptyset\}$.

In $\mathcal{H}_S$, the *set inclusion* relation ($\subseteq$) provides a partial order. The connected union of two HFLTSs is defined as the least element of $\mathcal{H}_S \cup \{\emptyset\}$, based on the subset inclusion relation $\subseteq$, that contains both HFLTSs. The connected union together with the intersection provide to the set of HFLTSs, $\mathcal{H}_S \cup \{\emptyset\}$, a lattice structure, as proven in [5].

Considering that in a specific context different users can consider differently the meaning of linguistic labels [10], the concept of *linguistic perceptual map* is introduced in [9] as a normalized measure of HFLTSs.

Definition 2. A *basic linguistic perceptual map* is a pair (S, μ) where $S = \{s_1, s_2, ..., s_n\}$ is a set of BLTs, and μ is a normalized measure over S, i.e., $\mu : S \to \mathbb{R}^+$such that $\sum_{i=1}^{n} \mu(s_i) = 1$.

Hereafter, for any $s_i \in S$, we call $\mu(s_i) \equiv \mu_i$ the *width* of the basic label s_i. The following definition, on the one hand, extends to $\mathcal{H}_S$ this width provided by the normalized measure μ over S and, on the other, introduces the concept of linguistic perceptual map.

Definition 3. Given $H = [s_i, s_j] \in \mathcal{H}_{(S,\mu)}$, then the *width* of H is $\mu([s_i, s_j]) = \sum_{k=i}^{j} \mu_k$. The pair $(\mathcal{H}_S, \mu)$, that we also denote as $\mathcal{H}_{(S,\mu)}$, is called *linguistic perceptual map.*

Any linguistic perceptual map is uniquely associated with a partition of the interval $[0, 1]$ into n sub-intervals of lengths $\mu_1, \ldots, \mu_n$ and landmarks $\lambda_0 = 0, \lambda_1, \ldots, \lambda_(n-1), \lambda_n = 1$. The relationship between the landmarks and the width of the basic linguistic labels is $\lambda_m = \sum_{i=1}^{m} \mu_i$ and $\mu_m = \lambda_m - \lambda_{m-1}$, for any $m = 1, \ldots, n$.

To aggregate the opinion of users when they use different linguistic perceptual maps, in this paper, following the concept introduced in [9], we consider the common perceptual map that provides the appropriate unified context to work with different user profiles.

Definition 4. Let $\mathcal{H}_{(S^m,\mu^m)}, m \in \{1, \cdots, k\}$ a set of k linguistic perceptual maps. Let $\{\lambda_0^m = 0, \lambda_1^m, \cdots, \lambda_{n_m}^m = 1\}$, for $m \in \{1, \cdots, k\}$, the sets of landmarks of the k partitions associated. The *common perceptual map* is the linguistic perceptual map associated to the partition of landmarks $\bigcup_{m=1}^{k} \bigcup_{i=0}^{n_i} \{\lambda_i^k\}$. The cardinality of this partition satisfies $N \equiv \#P_U \leq \sum_{j=1}^{k} n_j - 1$.

In this way, the common perceptual map is the adequate framework to aggregate the assessments of all individuals. In addition, based on the linguistic perceptual maps lattice structure, a perceptual-based distance between HFLTSs is defined. This distance will allow us to introduce the concept of centroid and measure the agreement among a set of users' opinions.

Definition 5. Let $\mathcal{H}_{(S,\mu)}$ be a linguistic perceptual map. Given $H_1, H_2 \in \mathcal{H}_{(S,\mu)}$, the *perceptual-based distance between H_1 and H_2* is defined as:

$$D_\mu(H_1, H_2) = 2 \cdot \mu(H_1 \sqcup H_2) - \mu(H_1) - \mu(H_2) \tag{1}$$

In ([9]) it is proved that this definition is indeed a distance in $\mathcal{H}_S$.

In this article, the centroid of a set of HFLTS is considered in order to aggregate user opinion to obtain a collective opinion of different profiles on an online platform.

Definition 6. Let $\mathcal{H}_{(S,\mu)}$ be a linguistic perceptual map. Let $\{H_j = [s_{L_j}, s_{R_j}] \in \mathcal{H}_{(S,\mu)} : j \in \{1, \ldots, k\}\}$ be a set of HFLTSs, the *centroid of this set,* denoted as H^C, is defined as:

$$H^C = arg \min_{H \in \mathcal{H}_{(S,\mu)}} \sum_{j=1}^{k} D_\mu(H, H_j). \tag{2}$$

In [9], it was proved that this centroid can be obtained by:

$$H^C = \{[s_L, s_R] \in \mathcal{H}_{(S,\mu)} \mid L \in \mathbb{M}(L_1, \cdots, L_k), R \in \mathbb{M}(R_1, \cdots, R_k), L \leq R\} \quad (3)$$

where $\mathbb{M}$ is the set that contains just the median if k is an odd number or the two central values and any integer number between them if k is even.

From the centroid of the opinions of a group of users, a measure of the agreement of the group can be obtained using the following concept.

Definition 7. Let $\mathcal{H}_{(S,\mu)}$ be a linguistic perceptual map. Let Λ be an object to be assessed using HFLTS from $\mathcal{H}_{(S,\mu)}$. Given a group of users, $G = \{d_1, d_2, \ldots, d_k\}$, let $H_j = [s_{L_j}, s_{R_j}] \in \mathcal{H}_{(S,\mu)}$ be the assessment of Λ made by user d_j for $j \in \{1, \ldots, k\}$ and H^C their centroid. The *degree of consensus of the group* is defined as:

$$\delta_\Lambda(G) = 1 - \frac{\sum_{j=1}^{k} D_\mu(H_j, H^C)}{\zeta} \quad (4)$$

with $\zeta = \frac{k}{2} \cdot (2 - \mu_1 - \mu_n)$ if k is even and $\zeta = \frac{k-1}{2} \cdot (2 - \mu_1 - \mu_n)$ if k is odd.

Note that ζ is a normalization factor that ensures that δ is between 0 and 1.

3 An application considering a food waste reduction platform

In this section we show the applicability of the proposed approach in a real context related to the reduction of food waste. "Too Good to Go" platform is a mobile app that connects customers to restaurants and food stores that have unsold food. These types of platforms help to achieve a better balance between economy, environment, and society. Using the concepts presented in Section 2, we consider a specific approach for this case.

3.1 Methodology

The methodology of the use case follows five steps (see Figure 1): 1) Determine groups of users, 2) Deduce groups' linguistic perceptual maps, 3) Obtain common perceptual map, 4) Compute centroid and degree of consensus in the common perceptual map.

1. *Determine Groups of Users:* We consider that different users interpret the same linguistic terms in different ways. Users from the same group use similar linguistic terms, and linguistic terms associated to different groups may be different. Each rating profile has a linguistic perceptual map specific to the group denoted by $\mathcal{H}_{(S,\mu)}$.

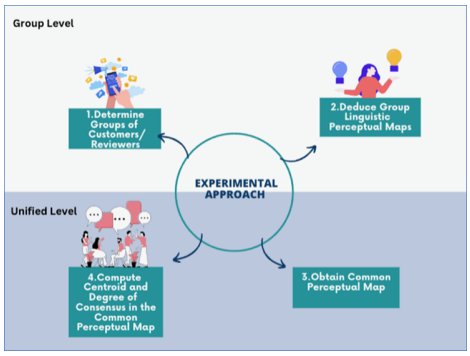


Fig. 1. Scheme of the methodology.

2. *Deduce Groups' Linguistic Perceptual Maps:* Each rating value is associated with a basic linguistic term s_i. For each group, the linguistic perceptual map $\{(\mathcal{H}_{S^j}, \mu^j) \mid j \in 1, \ldots, k\}$, is calculated based on the relative frequency with which the group assigns a particular rating. These relative frequencies are the widths of the basic labels in the group's perceptual map.
3. *Obtain Common Perceptual Map:* Once a perceptual map, $\mathcal{H}_{(S^j,\mu^j)}$, has been determined for each group, the common perceptual map, $\mathcal{H}_{(S^U,\mu^U)}$ is calculated following Definition 4. The labels in the common perceptual map are renamed $\{\lambda_1, \lambda_2, ..., \lambda_N\}$ for ease of reference and computation.
4. *Compute Centroid and Degree of Consensus in the Common Perceptual Map*: After the common perceptual map, $\mathcal{H}_{(S^U,\mu^U)}$, has been calculated, the centroid, H^C, and the degree of consensus, $\delta_\Lambda(G)$, are computed in this space for each group of users following Definitions 6 and 7, respectively.

3.2 Results

In this case, we have grouped users considering different countries. We assume that users from the same country use similar linguistic terms, and that linguistic terms associated to different countries may be different. For the sake of simplicity, we have considered just three groups of users: *Spanish*, *Danish* and *English*. Out of the 7972 users, 535 from the Spanish group , 4955 were from the Danish group and 2482 users from the English group.

For each country, we computed the relative frequency of the ratings (see Table 1) to define the landmarks in the partition associated to the linguistic perceptual map. The corresponding partitions of the unit interval, and their resulting perceptual maps, are the following:

$$\mathcal{H}_{(S^1,\mu^1)} : \{0, 0.64, 0.74, 0.8, 0.86, 1.0\};$$
$$\mathcal{H}_{(S^2,\mu^2)} : \{0, 0.09, 0.12, 0.15, 0.26, 1.0\};$$
$$\mathcal{H}_{(S^3,\mu^3)} : \{0, 0.38, 0.45, 0.51, 0.57, 1.0\}.$$

Table 1. Distribution of customers' ratings for the three countries considered.

Rating	England	Spain	Denmark
5	1055	76	3683
4	154	31	502
3	146	33	181
2	174	53	144
1	953	342	445
	2482	535	4955

Next, the common perceptual map is obtained following Definition 4. The partition associated with the common perceptual map is:

$$\mathcal{H}_{(S^U,\mu^U)} : \{0, 0.09, 0.12, 0.16, 0.26, 0.38, 0.45, 0.57, 0.64, 0.74, 0, 8, 0.86\}$$

Note that the cardinals of S^1, S^2 and S^3 are $n_1 = n_2 = n_3 = 5$ in all countries, while the cardinal of S^U is $N = 13$ in this case.

Ultimately, we calculate the collective centroid and degree of agreement across all countries' ratings within the common perceptual map. The collective centroid is $[s_5^*, s_{13}^*]$ and the degree of consensus is 0.61.

Table 2 shows the centroids for each country computed in their respective linguistic perceptual maps, together with their expressions in the common perceptual map. As it can be seen, the centroids in the initial linguistic perceptual maps are basic labels. In contrast, their expressions in the common perceptual map are wider (less precise).

Table 2. Distribution of the assements centroids as computed in the common perceptual map.

Centroids for each country	England	Spain	Denmark
In the original scales	$\{s_3\}$	$\{s_1\}$	$\{s_5\}$
In the common scale	$\{s_7^*\}$	$[s_1^*, s_9^*]$	$[s_5^*, s_{13}^*]$

Finally, in Figure 2, we compare the results obtained in each country's linguistic perceptual map with the collective centroid.

4 Conclusions and future research

In this paper we present a methodology to aggregate ratings coming from an online platform to aggregate users' opinions based on multiple unbalanced linguistic scales that allow us to aggregate and compare ratings from different profiles.

Results are shown to be relevant in rating contexts, where different individuals may assign different meanings to available linguistic labels. Individuals may even use different sets of linguistic terms.

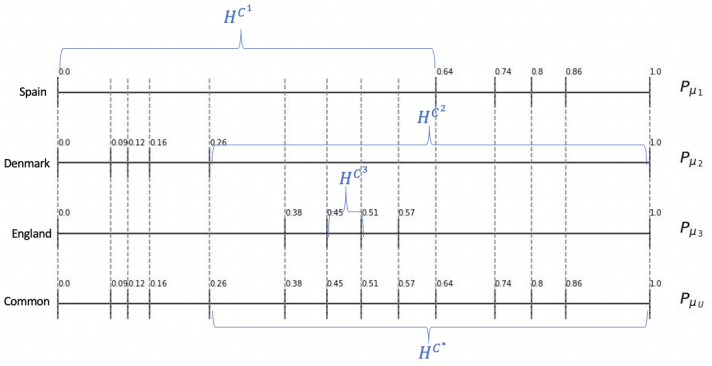


Fig. 2. Comparison of groups' centroids and collective centroid.

The presented real case, using real data from *T*rustpilot platform (https://www.trustpilot.com/) , proves the reliability of the proposed methodology. Users with a similar reviewing profile are grouped together. Three different groups of users are considered: Danish, German and Spanish profiles. Using the defined methodology, we obtain not only the aggregated opinion in each group, but also their level of consensus.

As a future work, from the theoretical point of view, we will analyze the definition of an interpretability function able to translate the results obtained in the common perceptual map to each one of the initial linguistic perceptual maps. And, from the application point of view, we plan to consider a larger variety of profiles and introduce the sentiment extracted from the written reviews to introduce it in the definition of the linguistic perceptual map for each profile.

Acknowledgements This research has been partially supported by the PERCEPTIONS Research Project (PID2020-114247GB-I00), funded by the Spanish Ministry of Science and Information Technology.

References

1. J. Zarocostas, Un says a third of food wasted, The Lancet 400 (10359) (2022) 1185.
2. P. Kinnunen, J. H. Guillaume, M. Taka, P. D'odorico, S. Siebert, M. J. Puma, M. Jalava, M. Kummu, Local food crop production can fulfil demand for less than one-third of the population, Nature Food 1 (4) (2020) 229–237.
3. M. Mattila, N. Mesiranta, A. Heikkinen, Platform-based sustainable business models: reducing food waste in food services, International Journal of Entrepreneurship and Innovation Management 24 (4-5) (2020) 249–265.
4. H. Liao, Z. Xu, X.-J. Zeng, Distance and similarity measures for hesitant fuzzy linguistic term sets and their application in multi-criteria decision making, Information Sciences 271 (2014) 125–142.

5. J. Montserrat-Adell, N. Agell, M. Sánchez, F. Prats, F. J. Ruiz, Modeling group assessments by means of hesitant fuzzy linguistic term sets, Journal of Applied Logic 23 (2017) 40–50.
6. N. Agell, M. SáNchez, F. Prats, L. Roselló, Ranking multi-attribute alternatives on the basis of linguistic labels in group decisions, Information Sciences 209 (2012) 49–60.
7. R. M. Rodriguez, L. Martinez, F. Herrera, Hesitant fuzzy linguistic term sets for decision making, IEEE Transactions on fuzzy systems 20 (1) (2012) 109–119.
8. J. Nguyen, J. Montserrat-Adell, N. Agell, M. Sánchez, F. J. Ruiz, Fusing hotel ratings and reviews with hesitant terms and consensus measures, Neural Computing and Applications 32 (2020) 15301–15311.
9. O. Porro, N. Agell, M. Sánchez, F. J. Ruiz, A multi-attribute group decision model based on unbalanced and multi-granular linguistic information: An application to assess entrepreneurial competencies in secondary schools, Applied Soft Computing 111 (2021) 107662.
10. C.-C. Li, Y. Dong, F. Herrera, E. Herrera-Viedma, L. Martínez, Personalized individual semantics in computing with words for supporting linguistic group decision making. an application on consensus reaching, Information Fusion 33 (2017) 29–40. doi:https://doi.org/10.1016/j.inffus.2016.04.005.
URL https://www.sciencedirect.com/science/article/pii/S1566253516300227

On the number of counterfeits and deletions to enforce m-eligibility in continuous data publishing

Adrián Tobar Nicolau[1 [0000−0003−0198−2475]],
Javier Parra Arnau[1 [0000−0002−1772−1088]], Jordi Forné[1 [0000−0002−8401−3292]], and
Esteve Pallarès[1 [0000−0001−6966−8349]]

Universitat Politécnica de Catalunya
{adrian.tobar,javier.parra,jordi.forne,esteve.pallares}@upc.edu

Abstract. The enforcement of privacy notions over datasets is the main procedure to guarantee syntactic privacy to the individuals contributing their data. Continuous data publishing consists in the republication of updating microdata. The most relevant syntactic notions in continuous data publishing are based on m-invariance. To achieve m-invariance, the existing methods must first alter the dataset to satisfy a property called m-eligibility. Essentially, a dataset can be made m-invariant if and only if it satisfies the m-eligibility constraint. Although guaranteeing the m-eligibility property is a crucial step, no theoretical study of the best strategies to achieve it has been conducted. This work performs such a study by giving strategies and demonstrating their optimality under two approaches: insertion of counterfeit tuples and partial publication.

Keywords: m-invariance · syntactic privacy · m-eligibility.

1 Introduction

During the last decades an increasing number of privacy mechanisms have been proposed in the literature for microdata that is, information to the users level. Among the classical mechanisms, there exist a distinction between syntactic methods, such as k-anonymity [10], l-diversity [9] and t-closeness [8] which base their protection in some enforcement of structure on the microtada and semantic methods headed by (ϵ, δ)-differential privacy [5, 6] which perturb the real values of microdata with random noise. In general each approach has its own strengths and weaknesses but syntactic approaches are preferred to retain utility on the data at the expense of a rigid definition of the attacker.

With the increased necessity to access other forms of data, different dynamic publishing scenarios for microdata have emerged such as Multiple data release [15], Sequential data release [12, 11] and Continuous data publishing [4]. The latter being a framework where a dataset that is being updated is published in several releases. In continuous data publishing, the main syntactic mechanisms are based on the m-invariance notion [14] and their variations [2, 3, 16, 13, 1, 7]. This notion is deeply related to the m-eligibility property [9], that is, that the dataset has no more than $\frac{1}{m}$ fraction of tuples with the same sensitive value.

The problem of imposing m-eligibility is of relevance since it is a necessary condition to achieve recursive $l-$diversity [9] and m-invariance. Essentially, a dataset can be made m-invariant if and only if it satisfies the m-eligibility constraint. This relation was already considered in [14]. To solve this limitation the main approach in the literature has been adding artificial tuples (counterfeits) to the dataset to make it m-eligible. However, no study on how to achieve m-eligibility efficiently has been developed.

The aim of this paper is to provide effective methods to obtain m-eligible datasets with minimal perturbation with respect to the original microdata. Due to the profound connection between m-eligibility and m-invariance we focus on this notion.

The paper is structured as follows. First Section 2 introduces the basic definitions of m-eligibility, the m-invariance problem and preliminary results.Then Section 3 presents the main contributions of this paper for the different approaches to obtain m-eligible datasets, providing proofs for upper bounds, correct execution and optimality of the algorithms. After that, an evaluation of behaviour is done in Section 4 where the different methods are compared. The paper ends with Section 5, which summarizes the conclusions and possible future work.

2 Preliminaries

M-invariance [14] was the first method to allow the republication of microdata, after being modified with insertions and deletions. It consists in enforcing on each publication that each class, i.e., subset of tuples with common quasi identifiers, has: at least m tuples; no two tuples with the same sensitive value and that if a tuple appears in two releases, both classes where it appears share the same set of sensitive values (signature). The motivation behind this definition is to avoid intersection attacks. An intersection attack is based on partially identifying a user to a reduced set of tuples of each publication. The sensitive value must appear in the intersection of the signatures of each candidates set. Since m-invariance enforces that two classes with a common tuple share the same signature, this attack may not reduce such intersection to less than m sensitive values. See Figure 1 for an example.

Most algorithms that implement m-invariance or their variations have the same core structure; on the first publication, it enforces that the input dataset is m-eligible, after that, the m-invariant structure is established and the dataset published.

For the consecutive releases a distinction between never published tuples (new) and published ones (old) is made. The old tuples are structured on the same class of their last publication. If a class is missing a tuple due to a deletion, a new tuple is put in as a replacement. If none exists, a counterfeit tuple is inserted. To the remaining new tuples that are not in a class the m-invariant structure is enforced. Finally the whole dataset is published.

Id	AGE	S.D.
1	[18-20]	HIV
2	[18-20]	FLU

(a) First 2-diverse publication.

Id	AGE	S.D.
1	[18-19]	HIV
3	[18-19]	ACNE
2	[20-21]	FLU
4	[20-21]	COUCH

(b) Second not 2-invariant publication.

Id	AGE	S.D.
1	[18-20]	HIV
2	[18-20]	FLU
3	[19-21]	ACNE
4	[19-21]	COUCH

(c) Second 2-invariant publication.

Fig. 1: Example of intersection attack. If an attacker is searching information of a participant with $age = 18$ then from the Table 1a deduces that it has sensitive value HIV or FLU and from the Table 1b that it has HIV or ACNE. Intersecting both cases, the attacker deduces that the attacked tuple has HIV. Such attacks are avoidable using m-invariance, in this case, publishing Table 1c instead of Table 1b.

Methods that rely on this procedure lack a detailed explanation on how to impose m-eligibility over the datasets. Most publications argue that they can achieve it with the insertion of few counterfeits but without a deeper insight on how they accomplish it. Our results show that an optimal strategy exists.

2.1 M-invariance and m-eligibility

Let T be a microdata table (dataset) of n individuals and d attributes, i.e., a matrix $A \in \mathbb{R}^{n \times d}$. The matrix A has the form $(QI|SD)$ where $QI \in \mathbb{R}^{n \times d-1}$ and $SD \in \mathbb{N}^{n \times 1}$. We denote the row $a_{i1}, ..., a_{id-1}$ as the quasi identifiers of tuple i and the value a_{id} as the sensitive value of tuple i. We define the m-invariant problem as follows:

Definition 1 (m-invariant problem). *Given a dataset T with l distinct sensitive values and a number $m \in [2, l]$, the m-invariant problem is partitioning T into subsets of tuples (clusters) of at least size m satisfying that no two tuples in the same subset have the same sensitive value.*

In general, this problem can have no feasible solutions, for instance when a sensitive value is much more frequent than the rest (see Proposition 1).

Definition 2. *Let $T \in \mathbb{R}^{n \times d}$ be a dataset with l distinct sensitive values.*

- *We denote by $|T|$ the number of tuples, i.e., the number of rows in T.*
- *We denote by $\{c_1, ..., c_l\}$ the counts of each sensitive value (there are c_1 tuples with sensitive value sd_1 and so on).*

Now we state the m-eligibility condition.

Definition 3. *A dataset $T \in \mathbb{R}^{n \times d}$ is m-eligible if no more than $\frac{|T|}{m}$ tuples have the same sensitive value in the dataset.*

With the previous definitions we are now able to present the main relation between m-eligibility and the m-invariance problem

3 m-eligibility with minimum counterfeits and deletions

This Section contains the main results of this paper. We start with the necessary definitions and results, and then introduce the m-invariant problem with counterfeits and with partial publication. Each subsection provides different properties that are used to prove a strategy to obtain m-eligible datasets minimizing the counterfeits and deletions necessary respectively. Additionally, for each problem, an upper bound to the minimal number of counterfeits/deletions necessary to obtain m-eligibility is provided. To conclude, the hybrid problem is presented and a fast strategy to compute a solution discussed.

Proposition 1 *A dataset $T \in \mathbb{R}^{n \times d}$ has a feasible solution for the m-invariant problem if and only if T is m-eligible.*

Proposition 1 implies that in order to guarantee a solution for non m-eligible datasets some form of relaxation to the combinatorial problem must be made. Two possible variations exist: the counterfeit method and the partial publication (Cach) [7].

Definition 4. *A dataset $T' \in \mathbb{R}^{p \times d}$ is:*

- *A subset of dataset $T \in \mathbb{R}^{n \times d}$ if it is a submatrix of p rows of T. We indicate it as $T' \subseteq T$.*
- *A superset of dataset $T \in \mathbb{R}^{n \times d}$ if $T \subseteq T'$.*
- *A maximal m-eligible subset of T (if it is an m-eligible subset and) if $|T'| \geq |T''|$ holds for any m-eligible subset T'' of T.*
- *A minimal m-eligible superset of T (if it is an m-eligible superset and) if $|T'| \leq |T''|$ holds for any m-eligible superset T'' of T.*

3.1 m-invariant problem with counterfeits

Since the first publication on m-invariance [14] the necessity to enforce m-eligibility has been tackled with the addition of counterfeit tuples to the dataset [2, 3, 13]. Despite that, no study on how to minimize the number of counterfeit tuples has been carried on. This Section gives tight results, showing the minimal number of counterfeit tuples needed to enforce m-eligibility and an algorithm that achieves that optimal bound.

Definition 5 (m-invariant problem with counterfeits). *Given a dataset T with l distinct sensitive values and a number $m \in [2, l]$, the m-invariant with counterfeits problem is partitioning $T' \supseteq T$ into subsets of tuples (clusters) of at least size m satisfying that no two tuples in the same subset have the same sensitive value, where T' is a minimal m-eligible superset of T.*

Proposition 2 determines the minimum number of tuples needed to obtain a minimal m-eligible superset.

Proposition 2 *Let $T \in \mathbb{R}^{n\times d}$ be a dataset and let $T' \in \mathbb{R}^{j\times d}$ be a minimal m-eligible superset of T then*

$$|T'| - |T| = \max(0, cm - n)$$

where c is the number of tuples with the most frequent sensitive value in T.

Proof. Observe that the m-eligibility condition $c - \frac{n}{m} \leq 0$, whenever we add a tuple with a new sensitive value changes to $c - \frac{n}{m} - \frac{1}{m}$ and, in general, for x tuples to $c - \frac{n}{m} - \frac{x}{m}$, is then straightforward that the minimal number of tuples to be added to ensure $c \leq \frac{j}{m}$ is at least $cm - n$. That can be achieved if we add $cm - n$ tuples each with a unique sensitive value not appearing in the dataset.

In general, the previous result can be of no interest since the addition of new sensitive values can be detrimental for the practical objectives of the computation. Next we present an improvement of Proposition2 since it does not need the insertion of new sensitive values.

Proposition 3 *Let $T \in \mathbb{R}^{n\times d}$ be a dataset with $l \geq m$ distinct sensitive values and let $T' \in \mathbb{R}^{j\times d}$ be a minimal m-eligible superset of T with $SD(T') \subseteq SD(T)$ then*

$$|T'| - |T| = \max(0, cm - n),$$

where c is the number of tuples with the most frequent sensitive value in T and $SD(T)$ is the set of sensitive values of T.

Proof. Assume T is not m-eligible, otherwise the proof is trivial. Consider T' the minimal m-eligible superset of the proof of Proposition 2. The counts of sensitive values of T and T' are $\{c_1, ..., c_l\}$ and $\{c_1, ..., c_l, c_{l+1} = 1, ..., c_k = 1\}$, in descending order respectively. Now observe that $c_1 \geq c_l + 1$ otherwise $c_i = c_j$ for all $i, j \in [1, l]$ which would imply that T is m-eligible. Consider now the process of changing the sensitive value of the tuple with sensitive value k to c_l, that yields a dataset T_1 with counts $\{c_1, ..., c_l + 1, 1, ..., 0\}$ which is clearly m-eligible since $c_l + 1 \leq c_1 \leq \frac{|T'|}{m}$, as previously stated. We can repeat this strategy with T_1, i.e., at each step remove a tuple with sensitive value in $[l + 1, ..., k]$ and add a new tuple with the least frequent sensitive value in $[1, l]$ and thus maintaining the m-eligibility of the dataset. After the last tuple with sensitive value in $[l+1, ..., k]$ is replaced we will have a minimal m-eligible superset of T with l distinct sensitive values.

From the proof of Proposition 3 we yield an algorithm to compute minimal m-eligible supersets.

Corollary 1. *Let $T \in \mathbb{R}^{n\times d}$ be a non m-eligible dataset with $l \geq m$ sensitive values. The following strategy computes T', a minimal m-eligible superset of T:*

1) *$T' = T \cup \{t\}$, where t is a tuple with sensitive value with least frequency in T.*

2) If T' is m-eligible, then stop. Otherwise $T = T'$ and go to 1.

Proof. Straightforward from the proof of Proposition 3.

Observe that step 1 of the algorithm of Corollary 1 could choose in-between several options showing that more than one optimal solution exists.

3.2 m-invariant problem with partial publication

Recently, the authors of [7] raised a new strategy to tackle the m-invariant problem: instead of adding counterfeits, they considered the removal of a small sample of tuples which they used in substitution of counterfeits. This subsection is devoted to the presentation of our results in relation to this problem, namely, we provide an upper bound on the minimal number of deletions, as well as an algorithm which constructs an optimal solution.

First we state the m-invariant problem with partial publication.

Definition 6 (m-invariant problem with partial publication). *Given a dataset T with l distinct sensitive values and a number $m \in [2, l]$, the m-invariant partial publication problem is partitioning $T' \subseteq T$ into subsets of tuples (clusters) of at least size m satisfying that no two tuples in the same subset have the same sensitive value, where T' is a maximal m-eligible subset of T'.*

This process demands a previous computation of T'. We present a fast strategy to find one instance.

Proposition 4 *If $T' \subseteq T$ is a maximal m-eligible subset of T and $\{c_1, ..., c_l\}$, $\{c'_1, ..., c'_l\}$ are the counts of each sensitive value in T and T' respectively (possibly 0) then for all $i \in [1, l]$ $c'_i \leq c_i - \max(0, \lceil \frac{mc_i - n}{m-1} \rceil)$.*

Proof. Observe that a dataset T is m-eligible if for all $i \in [1, l]$ holds $c_i - \frac{n}{m} \leq 0$. Notice that the function $f_i(x, y) = c_i - x - \frac{n-x-y}{m}$ returns the difference in-between the elements of the m-eligibility condition after removing from the dataset x tuples with sensitive value i and y tuples with a different sensitive value. It is straightforward to see that removing tuples with sensitive values i reduces f_i and removing tuples with sensitive value different from i increases f_i. Since we want $f_i \leq 0$, we compute the minimum number of tuples with sensitive value i that need to be removed to make $f_i \leq 0$:

$$\begin{aligned} f_i(x, y) &\leq 0 \\ c - x - \frac{n - x - y}{m} &\leq 0 \\ \frac{mc - n + y}{m - 1} &\leq x, \end{aligned}$$

which implies that at least $\lceil \frac{mc-n+y}{m-1} \rceil$ tuples must be removed. Since $y \geq 0$ we conclude the desired result.

This results gives a simple lower bound on the difference $|T| - |T'|$ and, as we see next, a method to compute T'.

Proposition 5 *Let $T \in \mathbb{R}^{n \times d}$ be a dataset, T' a subset of T and T° a maximal m-eligible subset of T' and let $\{c_1, ..., c_l\}$ and $\{c'_1, ..., c'_l\}$ be the sensitive values counts of T and T' respectively, then if for all $i \in [1, l]$ holds $c_1 - c'_i \leq \lceil \frac{mc_i - n}{m-1} \rceil$ then T° is also a maximal m-eligible subset of T.*

Proof. Suppose otherwise, that is that T° is not maximal w.r.t. T, then there exists $\bar{T}$ a maximal m-eligible subset of T such that $|T^\circ| < |\bar{T}|$. Since T° is maximal m-eligible subset of T' we know that $\bar{T} \nsubseteq T'$, in other words there is some sensitive value i such that $c'_i < \bar{c}_i$ where $\bar{c}_i$ is the count of that attribute in $\bar{T}$. However we deduce that $c_i - \bar{c}_i < c_i - c'_i \leq \lceil \frac{mc_i - n}{m-1} \rceil$ contradiction with Proposition 4.

Proposition 5 allows for a fast method to compute a maximal m-eligible subset since, as we see next, it can be used algorithmically.

Proposition 6 *Let $T \in \mathbb{R}^{n \times d}$ be a dataset with $l \geq m$ sensitive values. The following strategy computes T', a maximal m-eligible subset of T:*

1 *Compute $\{c_1, ..., c_l\}$ and $\{r_1, .., r_l\}$ where $r_i = \max(0, \lceil \frac{mc_i - n}{m-1} \rceil)$.*
2 *For each $i \in [1, l]$ remove r_i tuples from T with the ith sensitive value. Obtain T'.*
3 *If T' is m-eligible, then stop. Otherwise repeat with T'.*

Proof. First we prove that the algorithm halts and then that the output is the expected result.

With each loop we are removing tuples from dataset T and checking the m-eligibility of the result. Let us prove that if we do not remove at least one tuple from T then T' is m-eligible. If no element is removed, then $\frac{mc_i - n}{m-1} \leq 0$ which implies $c_i \leq \frac{n}{m}$ for all $i \in [1, l]$ exactly the condition of m-eligibility. Now, since each extra iteration implies the removal of at least one tuple, no more iterations than tuples can be done. We conclude that the algorithm always halts and that the output is m-eligible.

During the execution of the strategy we have created a finite list $T \supseteq T_1 \supseteq ... \supseteq T_k$ verifying the hypothesis of Proposition 5. Since T_k is a maximal m-eligible subset of itself we deduce, using Proposition 5, that T_k is a maximal m-eligible subset of $T_{k-1}, ..., T_1, T$.

This leads to a fast way to obtain an m-eligible subset, which can be then used to compute the desired solution. Notice the non unicity of solutions should be taken into account (a removed tuple is interchangeable with an existing one if they have the same sensitive value) if a utility metric is being considered in the solution as an objective to minimize (reduce the SSE,...). Such considerations are out of the scope of this paper.

3.3 Hybrid m-invariance problem

We define the hybrid m-invariance problem as allowing, simultaneously, the removal and insertion of tuples. Consider a dataset with counts $\{10, 9, 7, 1\}$ where we seek 3-invariance. Via 3 additions we obtain $\{10, 9, 7, 4\}$, a minimal 3-eligible superset. Via 3 deletions we obtain $\{8, 8, 7, 1\}$, a maximal 3-eligible subset. But the frequencies $\{9, 9, 7, 2\}$ are obtained with only one addition and one deletion strictly reducing the number of modifications in the dataset while obtaining 3-eligibility.

The hybrid approach has not been extensively tackled in the literature, only in a particular case of [7], so the following results are focused on establishing the basis for future algorithms that need a fast enforcement of m-eligibility with a reduced number of changes on the dataset over the disjoint choice of counterfeits or deletions. Since the desired output is not a subset nor a superset, we define the similarity of two datasets as follows.

Definition 7. *Let $T, T' \in \mathbb{R}^{n\times d}$ be datasets with l distinct sensitive values and respective sensitive value counts $\{c_1, ..., c_l\}$ and $\{c'_1, ..., c'_l\}$ (possibly 0). We define the distance $d(T, T')$ as*

$$d(T, T') = \sum_{i=1}^{l} |c_i - c'_i|,$$

where $|a|$ denotes the absolute value of a. That is the sum of absolute differences between the counts of each sensitive value on each dataset.

Observe that the defined distance can be conceptualized as the sum of non-redundant[1] additions and deletions performed in the dataset T to obtain T' or viceversa. Now we define the concept of closest m-eligible dataset.

Definition 8. *Let $T \in \mathbb{R}^{n\times d}$ be a dataset, we say that $T' \in \mathbb{R}^{j\times d}$ is a closest m-eligible dataset of T if it is m-eligible, $SD(T') \subseteq SD(T)$ and $d(T, T') \leq d(T, T'')$ for any T'' m-eligible dataset with $SD(T'') \subseteq SD(T)$. Where $SD(T)$ is the set of distinct sensitive values of tuples of T.*

From the results of the m-invariant problem with counterfeits and partial publication we obtain an upper bound for the hybrid problem.

Proposition 7 *Let $T \in \mathbb{R}^{n\times d}$ be a dataset and T' a closest m-eligible dataset of T then*

$$d(T, T') \leq \min(d(T, T_{super}), D(T, T_{sub})) \leq \max(0, cm - n)$$

where T_{super} is a minimal m-eligible superset of T and T_{sub} a maximal m-eligible subset of T.

[1] Redundant means that a tuple has been deleted and a counterfeit with their sensitive value has been added.

Proof. From Proposition 3 we know that there exists T_{super} a minimal m-eligible superset of T such that $|T_{super}| - |T| = d(T, T_{super}) = \max(0, cm - n)$ and $SD(T_{super}) \subseteq SD(T)$. Now since T' is closest to T we have $d(T, T') \leq d(T, T_{super}) = \max(0, cm-n)$. Similarly, since $SD(T_{sub}) \subseteq SD(T)$ holds $d(T, T') \leq d(T, T_{sub})$.

Proposition 7 gives us a reduced search space for the optimal solution, in other words, no more than $cm - n$ modifications will be needed to obtain a closest m-eligible dataset for a non m-eligible dataset T.

Proposition 8 *Let T° be a closest m-eligible dataset of a dataset T, and let a, d be the minimal number of necessary additions and deletions, respectively, done to T to obtain T°, then the dataset $\bar{T}^\circ$ made by iteratively adding a times a tuple with minimal frequency sensitive value and iterativey removing d times a tuple with maximal frequency sensitive value, is also a closest m-eligible dataset of T.*

Proof. Let c_1° be the count of the most frequent sensitive value in T° and $\bar{c_i^\circ}$ the ith most frequent sensitive value in $\bar{T}^\circ$. From the construction of $\bar{T}^\circ$ we have $\bar{c_1^\circ} \leq c_1^\circ$. Since the same number of additions and deletions have been performed on T° and $\bar{T}^\circ$, we know that $|T^\circ| = |\bar{T}^\circ|$. We conclude that $\bar{c_i^\circ} \leq \bar{c_1^\circ} \leq c_1^\circ \leq \frac{|T^\circ|}{m} = \frac{|\bar{T}^\circ|}{m}$ which proves the m-eligibility. It is easy to see that $d(T, T^\circ) = a + b = d(T, \bar{T}^\circ)$ completing the proof.

From last Proposition we reduce the search space, at each step, choosing between adding minimal sensitive value frequency tuple or removing a maximal sensitive value frequency tuple. The following algorithm does that process greedily.

Proposition 9 *Let T be a dataset with $l \geq m$ distinct sensitive values. The following algorithm outputs a m-eligible dataset of T.*

- *While T not m-eligible:*
  - *Compute $T_{add} = T \cup \{t_{min}\}$ and $T_{del} = T \setminus \{t_{max}\}$.*
  - *Compute $R_{del} = \sum_{i=1}^{l} \max(0, c_i^{del} - \frac{|T_{del}|}{m})$.*
  - *Compute $R_{add} = \sum_{i=1}^{l} \max(0, c_i^{add} - \frac{|T_{add}|}{m})$.*
  - *If $R_{del} \leq R_{add}$ then $T = T_{del}$ else $T = T_{add}$.*
- *Return T.*

Where t_{min} is a counterfeit tuple with sensitive value with minimal frequency in T, t_{max} is a tuple of T with maximal frequency sensitive value, T_{del} and T_{add} have sensitive value counts c_i^{del} and c_i^{add} respectively for $i \in [1, l]$.

Although we do not have a formal proof of optimality for the algorithm of Proposition 9, we have observed that its results are good, outperforming in many cases the non-hybrid approaches. We expect to develop a provably optimal output algorithm as future research.

4 Evaluation

We evaluate the different strategies presented in this paper for a real dataset commonly used in data privacy known as the adult dataset.[2]

4.1 Empirical evaluation

For our experimental evaluation we implemented the algorithms of Section 3 and compute their results, setting as sensitive value the columns work-class, occupation, education, marital status and relationship.

Figure 2 was formed by, for each sensitive value with l distinct values, for each $m \in [2, l-1]$ a computation on the number of modifications needed to achieve m-eligibility. Figure 2 compares the number of modifications over the relation m/l, that is the eligibility parameter over the number of distinct sensitive values of the dataset. On Figure 2a the dashed horizontal line corresponds half the dataset size, if a value surpasses such line more than one third of dataset is formed by counterfeits. If more than half the dataset is made with counterfeits the values are not reported on the figure to maintain the scale of the figures. Figure 2d is the not cropped version of Figure 2a. On Figure 2b the dashed horizontal line is placed at half the dataset size. If a value surpasses such line then more than half the dataset has been deleted.

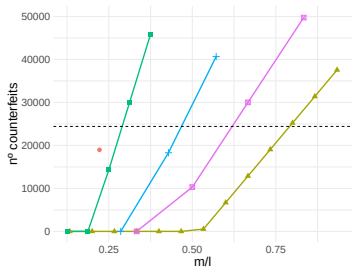
4.2 Observations

As we can see from Figure 2 the number of modifications needed to enforce m-eligibility grows as the parameter m increases, that was expected since m-eligibility is a descendent property, that is, m-eligibility implies (m-1)-eligibility. The use of counterfeits over deletions or vice-versa is not trivial since non improves on the other in all cases. The heuristic algorithm of Proposition 9 presents the best results making the hybrid method the preferred approach if the objective is reducing modifications. Although the hybrid approach is best, for small values of m the use of any method yields similar results (see Figure 2).

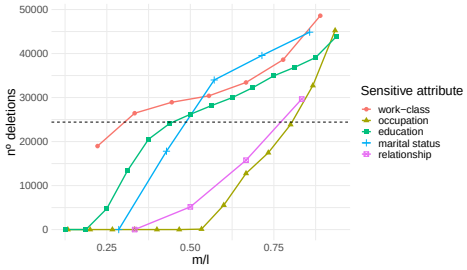
5 Conclusions and future work

This paper gives a formal approach to the problem of enforcing m-eligibility over a dataset. We present upper bounds on the number of necessary modification to achieve m-eligibility for the m-invariant problem with counterfeit and with partial publication. Effective algorithms to compute optimal m-eligible dataset are presented with proofs of their correctness. We illustrate the novel hybrid problem and give initial results for practical implementations. We end up with an empirical evaluation of our results using a classical dataset in statistical disclosure control. We expect our results to ease the comparison of future empirical

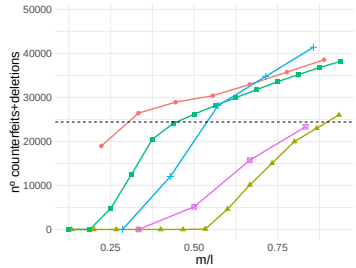
[2] `https://www.kaggle.com/uciml/adult-census-income`



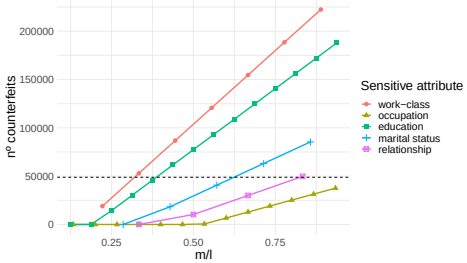
(a) Nº additions to obtain eligibility.



(b) Nº deletion to obtain eligibility.



(c) Nº modifications to obtain eligibility.



(d) Not cropped version of Figure 2a. Dashed line indicates original dataset size.

Fig. 2: Nº of modifications to obtain m-eligibility via counterfeits, deletions and the hybrid method. The x-axis is the relation m/l between m the eligibility parameter and l the number of distinct sensitive values. The dashed horizontal lines indicate when the number of modifications reaches half the size of the dataset

evaluations of novel approaches to the m-invariance problem, for example as a lower bound on the amount of modification needed to achieve m-invariance.

As future work we expect to extend our results on the hybrid m-invariant problem proving the optimality of our algorithm or of a new one that we design.

6 Acknowlegments

Javier Parra-Arnau is the recipient of a "Ramón y Cajal" fellowship (ref. RYC2021-034256-I) funded by the Spanish Ministry of Science and Innovation and the European Union.This work was also supported by the Spanish Government under the project COMPROMISE PID2020-113795RB-C31and through the project "MOBILYTICS" (TED2021-129782B-I00), funded by MCIN/AEI/10.13039/501100011033 and the European Union "NextGenerationEU"/PRTR.

References

1. Amiri, F., YAzdani, N., Shakery, A., Ho, S.S.: Bayesian-based anonymization framework against background knowledge attack in continuous data publishing. Trans. Data Priv. **12** (2019)
2. Anjum, A., Raschia, G.: Anonymizing sequential releases under arbitrary updates. In: Proc. Joint EDBT/ICDT 2013 Workshop. p. 145–154. EDBT '13, ACM, New York, NY, USA (2013). https://doi.org/10.1145/2457317.2457342
3. Anjum, A., Raschia, G., Gelgon, M., Khan, A., ur Rehman Malik, S., Ahmad, N., Ahmed, M., Suhail, S., Alam, M.M.: τ-safety: A privacy model for sequential publication with arbitrary updates. Comput. & Security **66**, 20–39 (2017). https://doi.org/10.1016/j.cose.2016.12.014
4. Byun, J.W., Sohn, Y., Bertino, E., Li, N.: Secure anonymization for incremental datasets. In: Secure Data Manage. (2006)
5. Dwork, C.: Differential privacy. In: Proc. Int. Colloq. Automata, Lang., Program. pp. 1–12. Springer-Verlag (2006)
6. Dwork, C., Naor, M., Pitassi, T., Rothblum, G.N.: Differential privacy under continual observation. In: Proc. ACM Int. Symp. Theory Comput. (STOC). pp. 715–724. ACM (2010)
7. Khan, R., Tao, X., Anjum, A., Malik, S., Yu, S., Khan, A., Rehman, W., Malik, H.: (τ,m)-slicedbucket privacy model for sequential anonymization for improving privacy and utility. Transactions on Emerging Telecommunications Technologies (06 2022). https://doi.org/10.1002/ett.4130
8. Li, N., Li, T., Venkatasubramanian, S.: t-Closeness: Privacy beyond k-anonymity and l-diversity. In: Proc. IEEE Int. Conf. Data Eng. (ICDE). pp. 106–115. Istanbul, Turkey (Apr 2007)
9. Machanavajjhala, A., Gehrke, J., Kiefer, D., Venkitasubramanian, M.: l-Diversity: Privacy beyond k-anonymity. In: Proc. IEEE Int. Conf. Data Eng. (ICDE). p. 24. Atlanta, GA (Apr 2006)
10. Samarati, P.: Protecting respondents identities in microdata release. IEEE Transactions on Knowledge and Data Engineering **13**(6), 1010–1027 (2001). https://doi.org/10.1109/69.971193
11. Shmueli, E., Tassa, T.: Privacy by diversity in sequential releases of databases. Inform. Sci. **298**, 344–372 (2015). https://doi.org/10.1016/j.ins.2014.11.005
12. Shmueli, E., Tassa, T., Wasserstein, R., Shapira, B., Rokach, L.: Limiting disclosure of sensitive data in sequential releases of databases. Inform. Sci. **191**, 98–127 (2012). https://doi.org/https://doi.org/10.1016/j.ins.2011.12.020
13. Temuujin, O., Ahn, J., Im, D.H.: Efficient l-diversity algorithm for preserving privacy of dynamically published datasets. IEEE Access **7**, 122878–122888 (2019). https://doi.org/10.1109/ACCESS.2019.2936301
14. Xiao, X., Tao, Y.: M-invariance: Towards privacy preserving re-publication of dynamic datasets. In: Proc. 2007 ACM SIGMOD Int. Conf. Manage. Data. p. 689–700. SIGMOD '07, Assoc. for Comput. Machinery, New York, NY, USA (2007). https://doi.org/10.1145/1247480.1247556
15. Yao, C., Wang, X.S., Jajodia, S.: Checking for k-anonymity violation by views. In: Proceedings of the 31st International Conference on Very Large Data Bases. p. 910–921. VLDB '05, VLDB Endowment (2005)
16. Zhu, H., Liang, H.B., Zhao, L., Peng, D.Y., Xiong, L.: τ -safe (l, k)-diversity privacy model for sequential publication with high utility. IEEE Access **7**, 687–701 (2019). https://doi.org/10.1109/ACCESS.2018.2885618