INFUS 2025

Data-driven identification of non-additive measures

Vicenç Torra and Zuzana Ontkovičová

August, 2025

Dept. CS, Umeå University, Sweden

Outline

1. Introduction
   - Aggregation: from arithmetic mean to fuzzy integrals
   - Measures: some basic concepts
   - Fuzzy Measures
2. Measure identification
   - Learning from (input, output) pairs: Regression models
   - Learning from associations: Metric learning
   - Distances
   - Putting the pieces together
   - Still another problem
3. Summary

Introduction

Aggregation: from arithmetic mean to fuzzy integrals

Aggegation functions

Aggregation functions:

They are functions to combine data from a set of information sources.

Aggegation functions

Aggregation (a bit more formal)

- Given values $a_1, \ldots, a_n$ assume $a_i \in \mathbb{R}$ (numerical)
- We combine them $\mathbb{C}(a_1, \ldots, a_n)$, also in $\mathbb{R}$

Examples

- Arithmetic mean $\sum_{i=1}^{n}(1/n)a_i$
- Weighted mean $\sum_{i=1}^{n} w_i a_i$
  with weights $w_1, \ldots, w_n$ s.t. $w_i \geq 0$ and $\sum w_i = 1$.

Aggregation functions

Aggregation (still more formal)

- Given values $a_1, \ldots, a_n$
- We combine them $\mathbb{C}(a_1, \ldots, a_n)$
- such that
  - if $a_i \leq a'_i$, then $\mathbb{C}(a_1, \ldots, a_n) \leq \mathbb{C}(a'_1, \ldots, a'_n)$
  - $\mathbb{C}(a, \ldots, a) = a$ for all a

    (or for some a, e.g. $a = 0$ and $a = 1$)

Aggregation functions

Aggregation

- Given values $a_1, \ldots, a_n$
- We combine them $\mathbb{C}(a_1, \ldots, a_n)$

Examples (more examples)

- Arithmetic mean $\sum_{i=1}^{n}(1/n)a_i$
- Weighted mean $\sum_{i=1}^{n} w_i a_i$
  with weights $w_1, \ldots, w_n$ s.t. $w_i \geq 0$ and $\sum w_i = 1$.
- OWA operators (linear combination of order statistics)

$$\sum_{i=1}^{n} w_i a_{\sigma(i)}$$

with σ a permutation s.t. values are sorted from largest to smallest

Aggregation functions

Aggregation (revisited, making f explicit)

- Given variables / information sources / criteria $X = \{x_1, \dots, x_n\}$
- and values $f : X \to [0, 1]$
- So, $f(x_i)$ value associated to x_i (i.e., $f(x_i) = a_i$)
- We combine them $\mathbb{C}(f(x_1), \dots, f(x_n))$

Examples

- Arithmetic mean $\sum_{i=1}^{n} (1/n) f(x_i)$
- Weighted mean $\sum_{i=1}^{n} w_i f(x_i)$
- OWA operators $\sum_{i=1}^{n} w_i f(x_{\sigma(i)})$

Aggregation functions

Aggregation (as averaging of f)

Given variables / information sources / criteria $X = \{x_1, \ldots, x_n\}$

values $f : X \to [0, 1]$, weights w_i for each x_i

We integrate f with respect to the weights
(weighted mean, expected value)

$$E_w(f) = \sum_i w_i f(x_i)$$

or equivalently

$$E_w(f) = \int f dw$$

Aggregation functions

Aggregation as averaging of f (with other integrals)

- Given variables / information sources / criteria $X = \{x_1, \ldots, x_n\}$
- values $f : X \to [0, 1]$, weights w_i for each x_i
- Then, other integrals are possible
  - Choquet and Sugeno integrals
  - Other generalizations exist as well
    - Murofushi & Sugeno fuzzy t-conorm integral, Bustince & Fernandez & Mesiar etc.

Aggregation functions

Aggregation as averaging of f

- Given variables / information sources / criteria $X = \{x_1, \ldots, x_n\}$
- values $f : X \to [0, 1]$, weights w_i for each x_i
- Then, other integrals are possible
  - Choquet and Sugeno integrals
  - but we need the concept of fuzzy measure

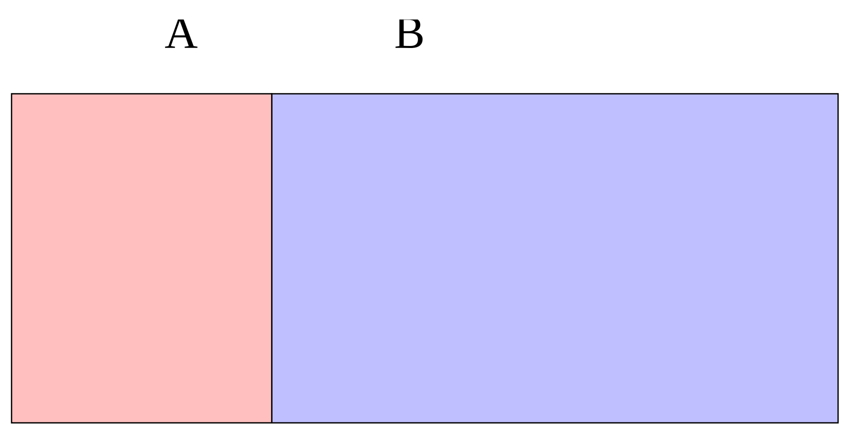
Measures: some basic concepts

Measures

Measures:

- A measure (mathematics) as a generalization of geometric measures (e.g., area)
- Used to express size, importance, and
- probabilities

Key property: additivity:



Additive measures

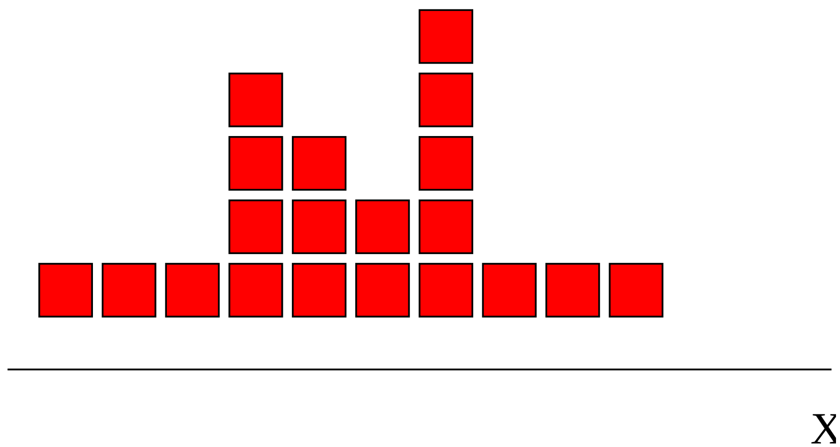
Additive measures: Formally (reference set X)

- $\mu(\emptyset) = 0$
- $\mu(S_1 \cup S_2) = \mu(S_1) + \mu(S_2)$ for disjoint S_1, S_2

Additive measures in statistics/probability theory

Measures: A typical example, probabilities!!
(on X and subsets of X, assume X finite)

- $\mu(\emptyset) = 0$
- $\mu(X) = 1$
- $\mu(S_1 \cup S_2) = \mu(S_1) + \mu(S_2)$ for disjoint S_1, S_2



Additive measures in decision making

Measures: or standard weights of sets of criteria/variables (on X and subsets of X, assume X finite)

- $\mu(\emptyset) = 0$
- $\mu(X) = 1$
- $\mu(S_1 \cup S_2) = \mu(S_1) + \mu(S_2)$ for disjoint S_1, S_2

That is,

- the importance of the set of criteria/variables (price, comfort, size) equals to
  - importance(price) + importance(comfort) + importance(size)

implicit assumption in problems using weighted means

Also, because of additivity for any disjoint S_1, S_2, C,

- if $\mu(S_1) < \mu(S_2)$ then also $\mu(S_1 \cup C) < \mu(S_2 \cup C)$.

Additive measures in decision making

Example: assessment

- Set of 10 students with marks in 5 subjects,
  assess = overall mark

student	ML	P	M	L	G	Subj. Evaluation
s_1	0.8	0.9	0.8	0.1	0.1	???
s_2	0.7	0.6	0.9	0.2	0.3	???
s_3	0.7	0.7	0.7	0.2	0.6	???
s_4	0.6	0.9	0.9	0.4	0.4	???
s_5	0.8	0.6	0.3	0.9	0.9	???
s_6	0.2	0.4	0.2	0.8	0.1	???
s_7	0.1	0.2	0.4	0.1	0.2	???
s_8	0.3	0.3	0.3	0.8	0.3	???
s_9	0.5	0.2	0.1	0.2	0.1	???
s_{10}	0.8	0.2	0.2	0.5	0.1	???

- How do we assess? Select weighted mean + weights, (model of decision)
  w(ML)=0.25, w(P)=0.25, w(M)=0.25, w(L)=0.15, w(G)=0.1
  $WM(s_1, w) = 0.65$

Fuzzy Measures

Fuzzy measures

Non-additive measures:

- Replace the additivity condition by a monotonicity condition
  $$S_1 \subseteq S_2 \text{ then } \mu(S_1) \leq \mu(S_2)$$
- This allows for interactions:
  - $\mu(S_1 \cup S_2) > \mu(S_1) + \mu(S_2)$
  - $\mu(S_1 \cup S_2) < \mu(S_1) + \mu(S_2)$

Fuzzy measures

Non-additive measures:

- Replace the additivity condition by a monotonicity condition

  $S_1 \subseteq S_2$ then $\mu(S_1) \leq \mu(S_2)$
- This allows for interactions:
  - $\mu(S_1 \cup S_2) > \mu(S_1) + \mu(S_2)$
  - $\mu(S_1 \cup S_2) < \mu(S_1) + \mu(S_2)$
- positive/negative interactions !
  - the importance of the set of criteria/variables (price, comfort, size) does not need to equal
  - importance(price) + importance(comfort) + importance(size)
- This allows for inversing inequalities for any disjoint S_1, S_2, C, it is possible
  - $\mu(S_1) < \mu(S_2)$ but also $\mu(S_1 \cup C) > \mu(S_2 \cup C)$.

Fuzzy measures in decision making

Example: assessment

- Set of 10 students with marks in 5 subjects,
  assess = overall mark

student	ML	P	M	L	G	Subj. Evaluation
s_1	0.8	0.9	0.8	0.1	0.1	???
s_2	0.7	0.6	0.9	0.2	0.3	???
s_3	0.7	0.7	0.7	0.2	0.6	???
s_4	0.6	0.9	0.9	0.4	0.4	???
s_5	0.8	0.6	0.3	0.9	0.9	???
s_6	0.2	0.4	0.2	0.8	0.1	???
s_7	0.1	0.2	0.4	0.1	0.2	???
s_8	0.3	0.3	0.3	0.8	0.3	???
s_9	0.5	0.2	0.1	0.2	0.1	???
s_{10}	0.8	0.2	0.2	0.5	0.1	???

- Now, to assess. Fuzzy measures to represent interactions
  $\mu(\{M\}) = 0.25$, $\mu(\{L\}) = 0.15$, but $\mu(\{M, L\}) = 0.5$
- $CI(s_1, \mu)$

Measure identification

Measure identification

- Measure identification: inverse problem
  - Given data, find the measure

Measure identification

- Measure identification: inverse problem
  - Given data, find the measure
- but what is data?
  - Data is (input, data) pairs as usual in machine learning
    $\Rightarrow$ model learning for regression
    (i.e., learning a ML model based on a fuzzy measure)

Measure identification

- Measure identification: inverse problem
  - Given data, find the measure
- but what is data?
  - Data is (input, data) pairs as usual in machine learning
    $\Rightarrow$ model learning for regression
    (i.e., learning a ML model based on a fuzzy measure)
  - Data is correct association between elements,
    $\Rightarrow$ metric learning
    (i.e., learning a distance based on a fuzzy measure)

Learning from (input, output) pairs

Regression models

Measure identification

- Regression model: An example
- Set of 10 students/marks + subjective evaluation

student	ML	P	M	L	G	Subj. Evaluation
s_1	0.8	0.9	0.8	0.1	0.1	0.7
s_2	0.7	0.6	0.9	0.2	0.3	0.6
s_3	0.7	0.7	0.7	0.2	0.6	0.6
s_4	0.6	0.9	0.9	0.4	0.4	0.8
s_5	0.8	0.6	0.3	0.9	0.9	0.8
s_6	0.2	0.4	0.2	0.8	0.1	0.3
s_7	0.1	0.2	0.4	0.1	0.2	0.1
s_8	0.3	0.3	0.3	0.8	0.3	0.4
s_9	0.5	0.2	0.1	0.2	0.1	0.3
s_{10}	0.8	0.2	0.2	0.5	0.1	0.5

- Model:
  - Weighted mean or linear regression (find weights w_i)

Measure identification

- Regression model: Formalization
- Optimization problem
  - Objective function: minimize error
  - Subject to constraints
    - ▷ Constraints on the parameters. E.g.,
      Weighted mean: $\sum w_i = 1$, $w_i \geq 0$
      Choquet integral: μ is a measure (i.e., monotonicity, ...)

Measure identification

- Regression model: Examples of solution
  - Solution by our software (python): http://www.mdai.cat/ifao/
    `ciFindMoebius(data85, kAdditive=1),`
    `fromMoebius2FM(ciFindMoebius(data85),5)`
  - weighted mean, weights learnt:
    [0.424, 0.411, 0.000, 0.125, 0.040]
  - Choquet integral, FM learnt:
    [0, 0.440, 0.197, 0.629, 0, 0.580, 0.561, 0.704, 0.200, 0.570, 0.200, 0.695, 0.704, 0.704, 0.732, 0.815, 0.438, 0.608, 0.619, 0.806, 0.699, 0.748, 0.744, 0.859, 0.585, 0.706, 0.684, 0.919, 0.825, 0.825, 0.825, 1.0]

Measure identification

- Regression model: Difficulties
  - As the number of parameters of the measure is high $(2^{|X|} - 2)$ we may have overfitting
    $\Rightarrow$ Families of reduced complexity:
    k-additive, belief functions, etc.
  - Choquet integral-based model is easy to solve with optimization
    Quadratic problem with linear constraints
  - Other measures with other integrals are not so easy to learn
    $\Rightarrow$ e.g., Sugeno integral

Measure identification

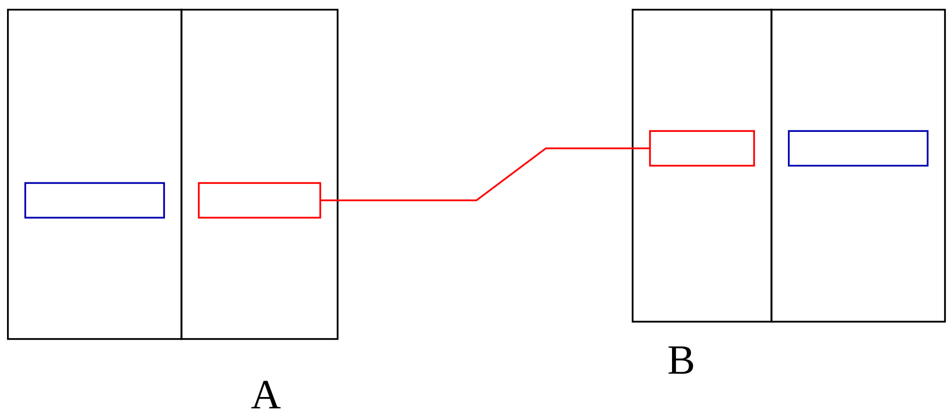
- Regression model: Sugeno integral-based
  - Optimization problem is no longer quadratic (because of the max-min in the integral)
  - Solution based on $(max, \oplus)$-transform and genetic algorithms
    - ▷ In GA, we represent possible solutions with chromosomes then, we need that most chromosomes lead to feasible measures.
    - ▷ With $(max, \oplus)$-transform this is the case.

Learning from associations

Metric learning

Context: Identity disclosure risk in data privacy

- Parametric Distance based record linkage: $d(A_i, B_i)$



- Find the *nearest* record (*nearest* in terms of a distance)
- Formally, 2 sets of vectors
  $A_i = (a_1, \ldots, a_N)$,
  (a_i protected version of b_i)
  $B_i = (b_1, \ldots, b_N)$
- $V_k(a_i)$: kth variable, ith record
- Distance $d(V_k(a_i), V_k(b_j))$ for all pairs (a_i, b_j).

- Goal: find the best distance (in terms of re-identification attacks)
  - Other families beyond Euclidean distance
    approach using fuzzy measures (non-additive measures)

Distances

From the Euclidean distance to CI-based distances

Distances: $d : [0,1]^{|X|} \times [0,1]^{|X|} \to \mathbb{R}^+$.

$$d(A = (a_1, \ldots, a_n), B = (b_1, \ldots, b_n))$$

- Euclidean distance (squared)

$$d(A = (a_1, \ldots, a_n), B = (b_1, \ldots, b_n)) = \sum (a_i - b_i)^2$$

From the Euclidean distance to CI-based distances

Distances: $d : [0,1]^{|X|} \times [0,1]^{|X|} \to \mathbb{R}^+$.

$$d(A = (a_1, \ldots, a_n), B = (b_1, \ldots, b_n))$$

- Euclidean distance (squared)

$$d(A = (a_1, \ldots, a_n), B = (b_1, \ldots, b_n)) = \sum (a_i - b_i)^2$$

- Weighted Euclidean (with weights w)

$$\begin{aligned} d_w(A, B) &= \sum w_i (a_i - b_i)^2 \\ &= WM(d(V_1(A), V_1(B)), \ldots, d(V_n(A), V_n(B))) \end{aligned}$$

where $d(V_i(A), V_i(B)) = (a_i - b_i)^2$.

From the Euclidean distance to CI-based distances

Distances: $d : [0,1]^{|X|} \times [0,1]^{|X|} \to \mathbb{R}^+$.

- Choquet integral-based

From the Euclidean distance to CI-based distances

Distances: $d : [0,1]^{|X|} \times [0,1]^{|X|} \to \mathbb{R}^+$.

- Choquet integral-based

$$d_\mu(A, B) = CI_\mu(d(V_1(A), V_1(B)), \ldots, d(V_n(A), V_n(B)))$$

where $d(V_i(A), V_i(B)) = (a_i - b_i)^2$,
and μ a fuzzy measure (satisfying fuzzy measure properties)

From the Euclidean distance to CI-based distances

Distances: $d : [0,1]^{|X|} \times [0,1]^{|X|} \to \mathbb{R}^+$.

- Choquet integral-based

$$d_\mu(A,B) = CI_\mu(d(V_1(A),V_1(B)),\ldots,d(V_n(A),V_n(B)))$$

where $d(V_i(A),V_i(B)) = (a_i - b_i)^2$,
and μ a fuzzy measure (satisfying fuzzy measure properties)

CI generalizes WM, and WM generalizes Euclidean distance,
So, appropriate μ and w make d_w and d_μ the Euclidean distance

When μ submodular, d_μ a metric (triangle inequality)

Putting the pieces together

Metric learning

- Can we do better than with the Euclidean distance?
  - Consider a supervised machine learning problem
  - We know the correct links, and look for the best distance

Metric learning

- Can we do better than with the Euclidean distance?
  - Consider a supervised machine learning problem
  - We know the correct links, and look for the best distance

- Other options (than the Euclidean distance):
  - Weighted Euclidean distance (weights w) d_w
  - Mahalanobis distance (using covariance matrix Q)

Metric learning

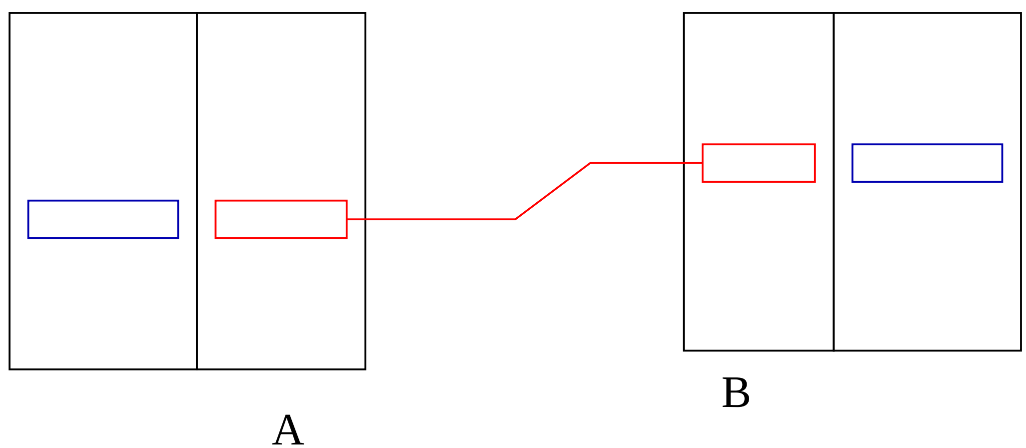
- Can we do better than with the Euclidean distance?
  - Consider a supervised machine learning problem
  - We know the correct links, and look for the best distance
- Other options (than the Euclidean distance):
  - Weighted Euclidean distance (weights w) d_w
  - Mahalanobis distance (using covariance matrix Q)
- But also
  - Choquet integral (measure μ) d_μ
  - Bilinear forms (using positive definite matrix Q) d_Q

Metric learning

- Can we do better than with the Euclidean distance?
  - Consider a supervised machine learning problem
  - We know the correct links, and look for the best distance
- Other options (than the Euclidean distance):
  - Weighted Euclidean distance (weights w) d_w
  - Mahalanobis distance (using covariance matrix Q)
- But also
  - Choquet integral (measure μ) d_μ
  - Bilinear forms (using positive definite matrix Q) d_Q
- From a machine learning perspective (correct links=reidentifications)
  - correct links $d_\mu \geq$ correct links $d_w \geq$ correct links d

Metric learning

- Metric learning. How to find these parameters (μ and Q)?
  - We consider the two files A and B
  - Assume they are aligned (A_i and B_i refer to the same record)
  - Then, the distance between A_i and B_i should be smaller than the distance A_i and other B_j.



Metric learning

- Metric learning. Formalization (case WM_w): $d_w(A_i, B_i)$
  - Main constraint: for a given i, for all $j \neq i$

$$\sum_{k=1}^{N} w_k d(V_k(A_i), V_k(B_j)) > \sum_{k=1}^{N} w_k d(V_k(A_i), V_k(B_i))$$

For aligned files A and B (i.e., A_i corresponds to B_i)

- This is sometimes impossible to satisfy for all i, so, introduce K_i (integer slack variable) which means $K_i = 1$ incorrect linkage, and then

$$\sum_{k=1}^{N} w_k (d(V_k(A_i), V_k(B_j)) - d(V_k(A_i), V_k(B_i))) + CK_i > 0$$

Metric learning

- Case $\mathbb{C} = WM$:

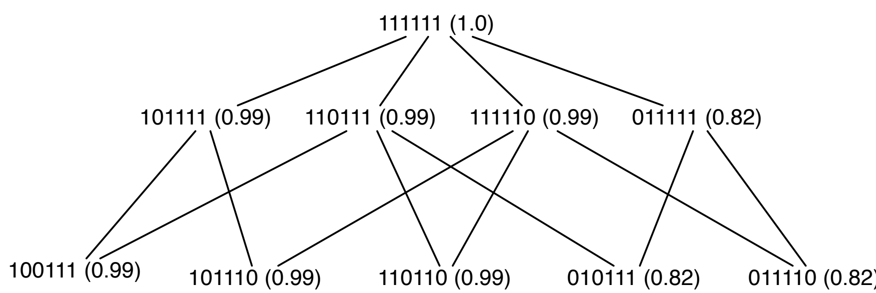
$$
\begin{aligned}
&Minimise && \sum_{i=1}^{N} K_i \\
&Subject\ to: && \\
& && \sum_{k=1}^{N} w_k(d(V_k(a_i), V_k(b_j)) - d(V_k(a_i), V_k(b_i))) + CK_i > 0 \\
& && K_i \in \{0,1\} \\
& && \sum_{k=1}^{N} w_k = 1 \\
& && w_k \geq 0
\end{aligned}
$$

- Similar with $\mathbb{C} = CI$ (Choquet integral) and μ
- Extensive work comparing different scenarios and $\mathbb{C}$.

Metric learning

- Results give:
  - number reidentifications in the worst-case scenario
  - Importance of weights (or sets of weights in fuzzy measures)

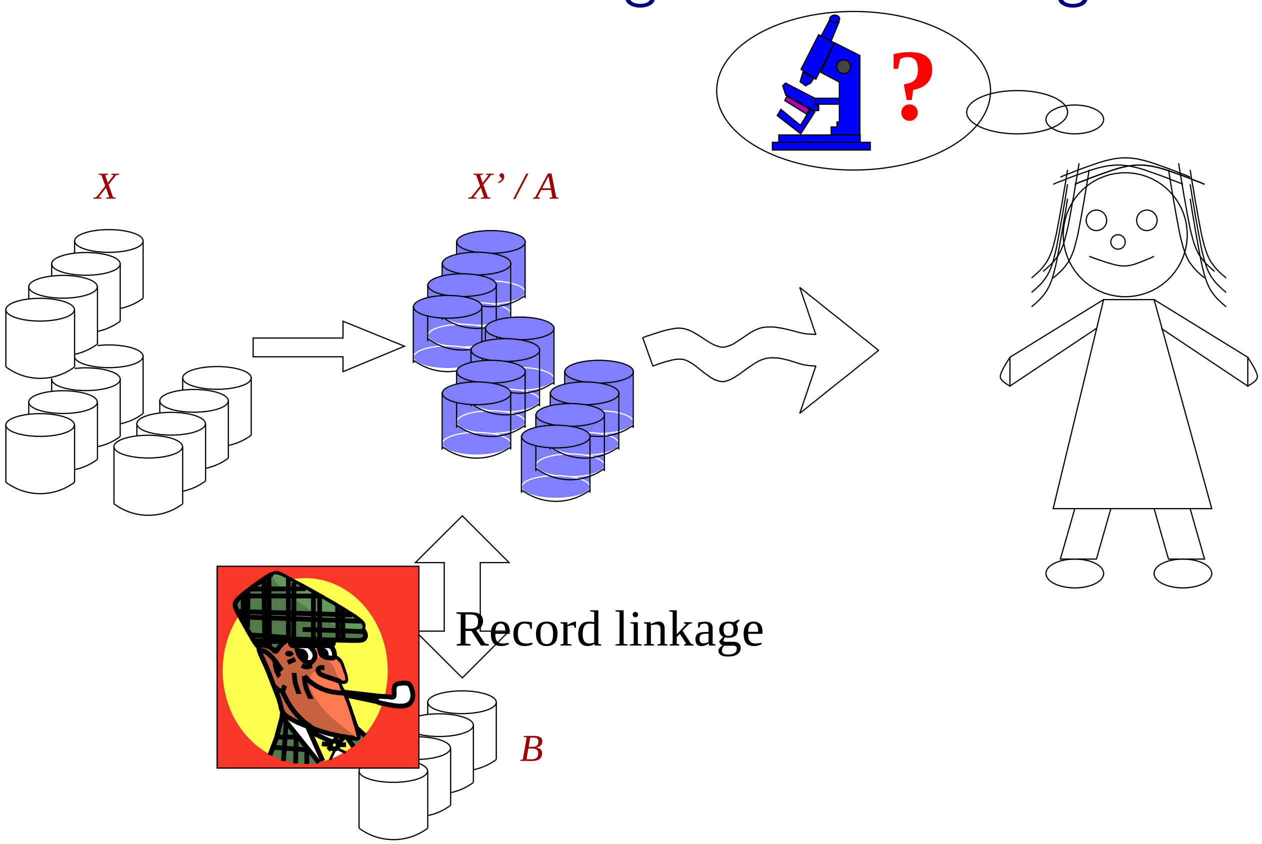
- Examples:
  - Choquet integral



  - Weighted Mean (WM):
    - ▷ V_1 0.016809573957189, V_2 0.00198841786482128, V_3 0.00452923777074791
    - ▷ V_4 0.138812880222131, V_5 0.835523953314578, V_6 0.00233593687053289

Metric learning

- Our application:
  - Metric learning for privacy-preserving machine learning
  - Model intrusion attacks in terms of attacking protected databases.
  - Assessing worst-case scenario using metric learning



Still another problem

Radon Nikodym-derivative

- Given two fuzzy measures ν and μ, find f such that

$$\nu(A) = (C) \int_A f d\mu$$

- This is the Radon-Nikodym-like derivative for additive measures useful for defining f-divergence, KL-divergence, entropy
  - difficult to solve for non-additive (fuzzy) measures
  - useful if we learn fuzzy measures, to compute distances

Summary

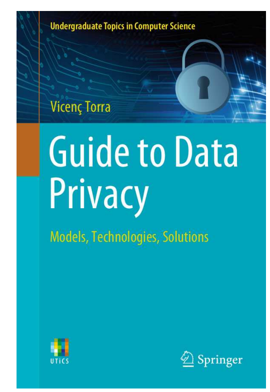
Summary

- Measure identification
  - Regression models
  - Metric learning
- Future directions
  - RN-like derivatives

References

References

- D. Abril, V. Torra, G. Navarro-Arribas (2015) Supervised learning using a symmetric bilinear form for record linkage. Inf. Fusion 26: 144-153.
- V. Torra (2022) $(Max, \oplus)$-transforms and genetic algorithms for fuzzy measure identification. Fuzzy Sets Syst. 451 253-265
- E. Türkarslan, V. Torra (2022) Measure Identification for the Choquet Integral: A Python Module. Int. J. Comput. Intell. Syst. 15:1 89
- Z. Ontkovicová, V. Torra (2024) Computation of Choquet integrals: Analytical approach for continuous functions. Inf. Sci. 679: 121105
- V. Torra (2022) Guide to Data Privacy, Springer.



Thank you