Monotonicity, a deep property in data science

Bernard DE BAETS, Ghent University, Belgium

SFC2019

Nancy, France, 03/09/2019
The narrator

- **Training**: mathematician – computer scientist – knowledge engineer
- **Profession**: senior full professor in applied mathematics
- **Affiliation**: Faculty of Bioscience Engineering at Ghent University
- **Multi- and interdisciplinary research** in three interlaced threads: knowledge-based, predictive and spatio-temporal modelling
- **Ultimate aim**: innovative applications in the bio-engineering sciences
Today’s main character in a three-act play: Monotonicity

A function \( f : P \to P' \) between two partially ordered sets (posets) \((P, \leq)\) and \((P', \leq')\) is called

- **increasing** if \( x \leq y \) implies \( f(x) \leq' f(y) \)
- **decreasing** if \( x \leq y \) implies \( f(y) \leq' f(x) \)
Act I

DECEPTION
Example: Soil erosion

**Phenomenon:** loss of soil by erosion increases with increasing slope angle and decreasing soil coverage with vegetation

(geoderma, Mitra et al., 1998)

<table>
<thead>
<tr>
<th>slope angle class</th>
<th>very large</th>
<th>large</th>
<th>medium</th>
<th>small</th>
<th>very small</th>
</tr>
</thead>
<tbody>
<tr>
<td>forest</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>pasture</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>land use class</td>
<td>moderately high</td>
<td>moderately low</td>
<td>low</td>
<td>low</td>
<td>low</td>
</tr>
</tbody>
</table>

Increasing, non-smooth rule base
Monotone fuzzy models?

Starting observations

- In many non-control applications (such as classification), fuzzy rule-based models are used for **one-shot decisions**
- At the level of linguistic terms, the underlying fuzzy rule base usually has some **flavor of monotonicity**
- However, is the resulting input-output function **effectively monotone**?
Fuzzy rule-based model

MISO model characteristics:

- $m$ input variables $X_\ell$ and a single output variable $Y$
- rules of the form
  
  $$R_s: \text{IF } X_1 \text{ IS } B_{j_1, s}^1 \text{ AND } \ldots \text{ AND } X_m \text{ IS } B_{j_m, s}^m \text{ THEN } Y \text{ IS } A_{i_s}$$
- linguistic values $B_{j_\ell, s}^\ell$ of $X_\ell$: trapezoidal; Ruspini partition
- linguistic values $A_{i_s}$: trapezoidal; Ruspini partition (bounded domain)
- natural ordering on the linguistic values of each variable
Ruspini partition
Mamdani–Assilian fuzzy models

Observation

Mamdani–Assilian fuzzy models with a monotone rule base do not necessarily result in a **monotone input-output mapping**

Monotone input-output behaviour under restrictive conditions only

If the original rule base is **complete** and **increasing**, then the input-output mapping can only be **increasing** in the following cases:

1. **Center-of-Gravity defuzzification:**
   - one input variable: basic t-norms $T_M$, $T_P$ and $T_L$
   - two or three input variables: $T_P$ and a **smooth** rule base

2. **Mean-of-Maxima defuzzification:**
   - one input variable: basic t-norms $T_M$, $T_P$ and $T_L$
   - two or **more** input variables: $T_M$ or $T_P$, and a **smooth** rule base
Alternative approach

Trivial, yet crucial observations
Consider an increasing function $f : \mathbb{R} \rightarrow \mathbb{R}$ such that $f(0) = 0$, then:

- if $x \geq 0$, then $f(x) \geq 0$
- if $x \leq 0$, then $f(x) \leq 0$

Consequences for a fuzzy rule in an increasing rule base
Consider a fuzzy rule “IF $X$ IS $C$ THEN $Y$ IS $D$”, then

- IF $X$ IS “at least” $C$ THEN $Y$ IS “at least” $D$
- IF $X$ IS “at most” $C$ THEN $Y$ IS “at most” $D
Cumulative modifiers of fuzzy sets

- **at-least modifier**: \( \text{ATL}(C)(x) = \sup\{C(t) \mid t \leq x\} \)
- **at-most modifier**: \( \text{ATM}(C)(x) = \sup\{C(t) \mid t \geq x\} \)
Implication-based fuzzy models (CRI)

**Connectives:** left-continuous t-norm and its residual implicant

**Modifying an increasing rule base**
- **ATL rule base:** applying ATL to all antecedents and consequents
- **ATM rule base:** applying ATM to all antecedents and consequents
- **ATLM rule base:** union of the above rule bases

**Increasing input-output mapping**
If the original rule base is increasing, then the input-output mapping is increasing in the following cases:

1. ATL rule base and **First-of-Maxima defuzzification**
2. ATM rule base and **Last-of-Maxima defuzzification**
3. ATLM rule base and **Mean-of-Maxima defuzzification**
Act II

OBSTRUCTION
The Age of Aggregation

New paths to growth
The Age of Aggregation

By Wayne G. Borchardt, Jill S. Dailey and Paul F. Nunes

Today’s expanded-scale businesses require ever more substantial sales volumes to create shareholder value. In order to grow, they must become market aggregators—masters at identifying commonalities in far-flung consumer segments—recognizing that the strongest growth will happen not within industries or markets but across and among them.
Data aggregation has become a very successful business model

The battle is for the **customer interface**:

- **Uber**: the world’s largest taxi company, owns no vehicles
- **Facebook**: the world’s most popular media owner, creates no content
- **Alibaba**: the most valuable retailer, has no inventory
- **Airbnb**: the world’s largest accommodation provider, owns no real estate
What processes do AGOP researchers study?

- Mathematically formalized by aggregation functions, formerly called aggregation operators (AGOPs)
- Historically, mostly confined to real numbers
- Numerous examples and parametric families: means, t-norms, t-conorms, uninorms, nullnorms, quasi-copulas, copulas, OWA operators, Sugeno integral, Choquet integral, ...
- Probably the most important spin-off of the fuzzy set community
- Monographs:
  - Aggregation Functions (2009) (Grabisch–Marichal–Mesiar–Pap)
  - etc.
1. The age of aggregation

Aggregation functions

Theory: Embarrassingly general

Consider a bounded poset \((P, \leq, 0, 1)\) and \(n \in \mathbb{N}\). A mapping \(A : P^n \to P\) is called an \(n\)-ary **aggregation function** on \((P, \leq)\) if it satisfies:

1. \(A(0, \ldots, 0) = 0\) and \(A(1, \ldots, 1) = 1\)

2. \(A\) is **increasing**: \(x \leq y \Rightarrow A(x) \leq A(y)\)

Some comments

- Practice is embarrassingly narrow
- The poset context appears dogmatic
- Does not address data types of current interest
First example: Compositional data

- \( k \)-dimensional compositional data vectors: simplex

\[ S_k = \{ x \in [0, 1]^k \mid \sum_{i=1}^{k} x_i = 1 \} \]

- Examples of application:
  - **soil science**: relative portions of sand, clay and silt in a soil sample
  - **chemistry**: compositions expressed as molar concentrations
  - **environmental science**: composition of air pollution
  - **mathematics**: weight vector of a weighted quasi-arithmetic mean
  - **fuzzy set theory**: vector of membership degrees in fuzzy \( c \)-means
  - **probability theory**: discrete probability distribution
Illustration: food composition (in %) \((k = 3)\)

Food composition (% fat, % carbonates, % protein) in barycentric coordinates
Mixing compositions

We can “aggregate” compositional data vectors componentwisely resulting in a new compositional data vector: $C : (S_k)^n \rightarrow S_k$

$$C(x^1, \ldots, x^n)_j = \frac{1}{n} \sum_{i=1}^{n} (x^i)_j$$

- The set $S_k$ is not a poset:
  - there is no natural ordering
  - there is no smallest or largest element

The function $C$ can be written as

$$C(x^1, \ldots, x^n)_j = \arg \min_y \sum_{i=1}^{n} ((x^i)_j - y)^2$$
Second example: Ranking data

- Examples of application:
  - Traditionally: voting, decision making, preference modelling
  - Nowadays: high-throughput, omics-scale, biological data, e.g. ranking of genes

- Different problem settings:
  - full rankings
  - incomplete rankings; top-$k$ lists

- The set of (full) rankings $\mathcal{L}(C)$ (briefly, $\mathcal{L}$) is **not** a poset:
  - there is **no** natural ordering
  - there is **no** smallest or largest element
2. Aggregation outside the poset framework

2.2. Ranking data

Aggregation methods for full rankings

- **Borda methods**: apply aggregation functions to the ranks (possibly leading to ties, resulting in a weak order)

- **Distance-based methods**: consider $n$ full rankings $\succ_i$

$$A(\succ_1, \ldots, \succ_n) = \arg \min_{\succ} \sum_{i=1}^{n} d(\succ_i, \succ)$$

where $d(\succ_i, \succ)$ is:

- **Kendall’s distance function** $K$
  (number of pairwise discordances)
  or

- **Spearman’s footrule distance function** $S$
  (sum of the absolute differences between the ranks)
3. Penalty-based aggregation

Penalty functions

Let $I = [a, b] \subseteq \mathbb{R}$. A function $P : I \times I^n \rightarrow \mathbb{R}$ is a **penalty function** if

1. $P(y; x) \geq 0$
2. $P(y; x) = 0$ if and only if $x = (y, \ldots, y)$
3. $P(\cdot; x)$ is quasi-convex and lower-semicontinuous

(The third condition implies that the set of minimizers of $P(\cdot; x)$ is either a singleton or an interval)
Penalty-based (aggregation) functions

Given a penalty function $P$, the corresponding **penalty-based function** is the function $f : I^n \rightarrow I$ defined by

$$f(x) = \frac{\ell(x) + r(x)}{2}$$

where $[\ell(x), r(x)]$ is the interval closure of the set of minimizers of $P(\cdot; x)$.

A penalty-based function $f$ is not necessarily increasing.
Remark

Originally, the following condition has been required for a (local) penalty function \((n = 1)\):

\[
\text{if } x' \leq x \leq y \text{ or } y \leq x \leq x', \text{ then } P(y; x) \leq P(y; x')
\]
Betweenness relations instead of order relations

Betweenness relation

A ternary relation \( B \) on \( X \) is called a betweenness relation (BR) if:

1. **Symmetry in the end points:** \((a, b, c) \in B \iff (c, b, a) \in B\)

2. **Closure:**
\[
((a, b, c) \in B \land (a, c, b) \in B) \iff b = c
\]

3. **End-point transitivity:**
\[
((o, a, b) \in B \land (o, b, c) \in B) \Rightarrow (o, a, c) \in B
\]

Product betweenness relation on \( X^n \)

The ternary relation \( B^{(n)} \) on \( X^n \) defined by

\[
(a, b, c) \in B^{(n)} \iff (\forall i \in \{1, \ldots, n\})(a_i, b_i, c_i) \in B
\]
Examples (order relation $\leq$, distance function $d$)

1. $B_0 = \{(x, y, z) \in X^3 \mid x = y \lor y = z\}$ (trivial BR)

2. $B_\leq = B_0 \cup \{(x, y, z) \in X^3 \mid (x \leq y \leq z) \lor (z \leq y \leq x)\}$

3. $B_d = \{(x, y, z) \in X^3 \mid d(x, z) = d(x, y) + d(y, z)\}$
A betweenness relation on compositional data

A natural betweenness relation: \( B_{S_k} := (B_{[0,1]})^{(k)} \cap (S_k)^3 \)

\( (x, y, z) \in (B_{[0,1]})^{(k)} \iff (\forall i \in \{1, \ldots, k\}) (\min(x_i, z_i) \leq y_i \leq \max(x_i, z_i)) \)
A betweenness relation on rankings

Betweenness relation based on Kendall’s d.f.:

\[(\succ_1, \succ_2, \succ_3) \in B_K \iff K(\succ_1, \succ_3) = K(\succ_1, \succ_2) + K(\succ_2, \succ_3)\]
Generalized penalty functions

Definition

Consider $n \in \mathbb{N}$, a set $X$ and a BR $B$ on $X^n$. A function $P : X \times X^n \to \mathbb{R}$ is called a **penalty function** (compatible with $B$) if

(P1) $P(y; x) \geq 0$

(P2) $P(y; x) = 0$ if and only if $x = (y, \ldots, y)$

(P3) The set of minimizers of $P(\cdot; x)$ is always non-empty

(P4) $P(y; x) \leq P(y; x')$, whenever $((y, \ldots, y), x, x') \in B$
### Generalized penalty functions

#### Optional conditions for fixed $x$

**(P5)** For any minimizer $z \in X$ of $P(\cdot; x)$ such that

$$((z, \ldots, z), (y, \ldots, y), (y', \ldots, y')) \in B$$

it holds that

$$P(y; x) \leq P(y'; x)$$

***(P6)*** For any two minimizers $z, z' \in X$ of $P(\cdot; x)$ such that

$$((z, \ldots, z), (y, \ldots, y), (z', \ldots, z')) \in B$$

it holds that

$$P(y; x) = P(z; x)$$

---

#### Penalty-based function

Given a penalty function $P$, the corresponding **penalty-based function** is the function $f : X^n \rightarrow \mathcal{P}(X)$ such that $f(x)$ is the set of minimizers of $P(\cdot; x)$.
How to create penalty functions?

**Monometric**

A mapping $M : X^2 \to \mathbb{R}$ is called a **monometric** w.r.t. a betweenness relation $B$ on $X$ if it satisfies

1. **Non-negativity**: $M(x, y) \geq 0$
2. **Coincidence**: $M(x, y) = 0 \iff x = y$
3. **Compatibility**: if $(x, y, z) \in B$, then $M(x, y) \leq M(x, z)$

**Proposition**

A distance function $d : X^2 \to \mathbb{R}$ is a monometric w.r.t.

$$B_d = \{(x, y, z) \in X^3 \mid d(x, z) = d(x, y) + d(y, z)\}$$
## Monometric-based penalty functions

**Monometric \( M \) on \( X \) w.r.t. \( B \)**

The function \( P : X^{n+1} \to \mathbb{R}^+ \) defined by

\[
P(y; x) = A(M(y, x_1), \ldots, M(y, x_n)),
\]

is a penalty function (compatible with the betweenness relation \( B^{(n)} \) on \( X^n \)) if \( A \) is an \( n \)-ary **increasing** function such that \( A(x_1, \ldots, x_n) = 0 \) iff \( x_i = 0 \) for \( i = 1, \ldots, n \).

**Particular cases: addition and maximum**

- \( P(y; x) = \sum_{i=1}^{n} M(y, x_i) \)
- \( P(y; x) = \max_{i=1}^{n} M(y, x_i) \)
5. Key examples

Key examples of penalty-based aggregation

- **Averaging compositional data:**
  - satisfies (P5) and (P6)

- **Method of Kemeny for rankings:**
  - satisfies (P5)
  - satisfies (P6) for those profiles of rankings for which there exists a Condorcet ranking

- **The center string procedure:**

\[
C(S_1, \ldots, S_n) = \arg \min_S \max_{i=1}^n d_H(S_i, S)
\]

where \(d_H\) is the Hamming distance between strings of the same length
  - neither satisfies (P5) nor (P6)
Trends in a related area: Machine Learning

- Originally: shared interest with statistics in classification and regression problems (focus on generalization abilities rather than inference)

- Currently: focus on a broad range of problem settings involving more and more complex data (at the input as well as the output side)
  - classification (multi-label, hierarchical, extreme)
  - regression (ordinal, monotone)
  - structured prediction or structured (output) learning
  - preference learning (label ranking, instance ranking)
  - pairwise learning
  - relational learning
  - multi-task learning
  - and so on

- One commonality: all models (i.e. functions) are the result of solving a mathematical optimization problem
Act III

SHORT-SIGHTEDNESS
Toy example

Classification problem:

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>class label</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$-$</td>
<td>$-$</td>
<td>$+$</td>
<td>A</td>
</tr>
<tr>
<td>$a_2$</td>
<td>$+$</td>
<td>$-$</td>
<td>$-$</td>
<td>B</td>
</tr>
<tr>
<td>$a_3$</td>
<td>$-$</td>
<td>$+$</td>
<td>$+$</td>
<td>C</td>
</tr>
<tr>
<td>$a_4$</td>
<td>$+$</td>
<td>$+$</td>
<td>$-$</td>
<td>B</td>
</tr>
</tbody>
</table>
Monotone classification problem:

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$-$</td>
<td>$-$</td>
<td>$+$</td>
<td>Bad</td>
</tr>
<tr>
<td>$a_2$</td>
<td>$+$</td>
<td>$-$</td>
<td>$-$</td>
<td>Moderate</td>
</tr>
<tr>
<td>$a_3$</td>
<td>$-$</td>
<td>$+$</td>
<td>$+$</td>
<td>Good</td>
</tr>
<tr>
<td>$a_4$</td>
<td>$+$</td>
<td>$+$</td>
<td>$-$</td>
<td>Moderate</td>
</tr>
</tbody>
</table>

Bernard De Baets (KERMIT) Monotonicity Nancy, France, 03/09/2019
# Monotone classification

## Toy example

Monotone classification problem:

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>Bad</td>
</tr>
<tr>
<td>$a_2$</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>Moderate</td>
</tr>
<tr>
<td>$a_3$</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>Good</td>
</tr>
<tr>
<td>$a_4$</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>Moderate</td>
</tr>
<tr>
<td>$a_5$</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>Good</td>
</tr>
<tr>
<td>$a_6$</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>Moderate</td>
</tr>
</tbody>
</table>

If monotonicity applies, any violation of it is simply unacceptable.

How to produce guaranteed **monotone** classification results, even when the set of learning examples is **not monotone**?
Multi-class classification

- Problem: to assign labels from a finite set $L$ to the elements of some set of objects $\Omega$
- Each object $a \in \Omega$ is represented by a feature vector
  $$a = (c_1(a), c_2(a), \ldots, c_n(a))$$
  in the feature space $X$
- Collection of learning examples: multiset
  $$(S, d) \equiv \{\langle a, d(a) \rangle | a \in S\}$$
  where:
  - $S \subseteq \Omega$ is a given set of objects
  - $d : S \rightarrow L$ is the associated decision function
  - multiset: the same entry can occur more than once, usually giving this entry more importance: we do not write $\langle a, d(a) \rangle$
  - notation: $S_X = \{a | a \in S\}$
Multi-class classification

- **Goal** of supervised classification algorithms:
  - extend the function $d$ to $\Omega$ in the most reasonable way
  - concentrate on finding a function $\lambda: \mathcal{X} \rightarrow \mathcal{L}$ that **minimizes the expected loss** on an independent set of test examples

- Different approaches:
  - **instance-based**, such as nearest neighbour methods
  - **model-based**, such as classification trees

- **Distribution classifiers**: output is a PMF over $\mathcal{L}$
  - mathematically: $\tilde{\lambda}: \mathcal{X} \rightarrow \mathcal{F}(\mathcal{L})$
  - selecting a single label: Bayesian decision
    (label with the highest probability is returned)
Multi-criteria evaluation

- In many cases, $\mathcal{L}$ exhibits a natural ordering and could be treated as an ordinal scale (chain): **ordinal classification/regression**

- Often, objects are described by (true) criteria $(c_i, \leq c_i)$ (chains)

- The **product ordering** turns $\mathcal{X}$ into a **partially ordered set** $(\mathcal{X}, \leq \mathcal{X})$ (poset)

- Multi-criteria evaluation: quality assessment, environmental data, social surveys, etc.

**Natural monotonicity constraint**

An object $a$ that scores at least as good on all criteria as an object $b$ must be classified (ranked) at least as good as object $b$
1. Monotone classification

Monotone classification

Monotone classifier

Classifier + **basic monotonicity constraint**:

\[ x < x \ y \Rightarrow \lambda(x) \leq \lambda(y) \]

(supervised ranking/ordered sorting, monotone ordinal regression)

Monotone distribution classifier

Distribution classifier + **stochastic monotonicity constraint**:

\[ x < x \ y \Rightarrow \tilde{\lambda}(x) \leq_{SD} \tilde{\lambda}(y) \]

(First order) **Stochastic Dominance** (SD):

\[ f_X \leq_{SD} f_Y \iff F_X \geq F_Y \]
Stochastic dominance

\[ f_Y \quad f_X \]

\[ F_Y \quad F_X \]
1. Monotone classification

Selecting a single label

- Bayesian decision potentially breaks the desired monotonicity and is no longer acceptable in this case.

- The well-known relationship

\[ f_X \preceq_{SD} f_Y \Rightarrow E[f_X] \leq E[f_Y] \]

cannot be used as it requires the transformation of the ordinal scale into a numeric scale.

- **Set of medians** (interval) of \( f_X \):

\[ \text{med}(f_X) = \{ \ell \in \mathcal{L} \mid \mathcal{P}\{X \leq \ell\} \geq 1/2 \land \mathcal{P}\{X \geq \ell\} \geq 1/2 \} \]

  - reduces in the continuous case to the median \( m \): \( \mathcal{P}\{X \leq m\} = 1/2 \)
  - only endpoints of the interval have non-zero probability.
1. Monotone classification

Selecting a single label from the set of medians

- The set of medians reduces the PMF to an interval. Does there exist an ordering on intervals that is compatible with FSD?

\[ [k_1, \ell_1] \leq_{\mathcal{L}} [k_2, \ell_2] \iff (k_1 \leq \mathcal{L} k_2 \land \ell_1 \leq \mathcal{L} \ell_2) \]

- New relationship:

\[ f_X \leq_{\text{SD}} f_Y \Rightarrow \text{med}(f_X) \leq_{\mathcal{L}}^{[2]} \text{med}(f_Y) \]

Selecting a single label

1. **Pessimistic median** (lower)
2. **Optimistic median** (upper)
3. Midpoint (or smaller/greater of the two midpoints) [not meaningful]

...turn a monotone distribution classifier into a monotone classifier...
How to label a new point?
Minimal and maximal extensions

1. **Minimal Extension**: \( \lambda_{\text{min}} : \mathcal{X} \rightarrow \mathcal{L} \)
   - assigns best label of "objects below":
   \[
   \lambda_{\text{min}}(x) = \max\{d(s) \mid s \in S_x \land s \leq_x x\}
   \]
   - if no such object: \( \lambda_{\text{min}}(x) = \min(\mathcal{L}) \)

2. **Maximal Extension**: \( \lambda_{\text{max}} : \mathcal{X} \rightarrow \mathcal{L} \)
   - assigns worst label of "objects above":
   \[
   \lambda_{\text{max}}(x) = \min\{d(s) \mid s \in S_x \land x \leq_x s\}
   \]
   - if no such object: \( \lambda_{\text{max}}(x) = \max(\mathcal{L}) \)

---

Monotone classifiers

1. \( \lambda_{\text{min}} \) and \( \lambda_{\text{max}} \) are monotone classifiers
2. **Interpolation**: midpoint leads to a monotone classifier
Things can go dead wrong
A more realistic non-monotone data set
3. Handling noise

Noise in multi-criteria evaluation

- \((S, d)\) is called **monotone** if for all \(x\) and \(y\) in \(S\)

\[
x = y \Rightarrow d(x) = d(y)
\]

(absence of doubt/ambiguity)

and

\[
x <_X y \Rightarrow d(x) \leq_L d(y)
\]

(absence of reversed preference)

- Non-monotonicity defines a **symmetric** and **transitive** relation on \(S\)

**Monotone extensions**

If the data set is monotone, then

1. \(\lambda_{\min}\) and \(\lambda_{\max}\) are monotone **extensions** of \(d\) to \(X\)

2. any monotone extension \(\lambda\) of \(d\) to \(X\):

\[
\lambda_{\min} \leq_L \lambda \leq_L \lambda_{\max}
\]
How to handle noise?

1. **Data reduction**: identify the noisy objects and **delete** them

2. **Data relabelling**: identify the noisy objects and **relabel** them

3. **Non-invasive approach**: keep the data set as is
   - excludes the use of some monotone classification algorithms
   - restricts the accuracy of any monotone classifier
     (independence number)
Option 1, Data reduction: A non-monotone data set
Option 1, Data reduction: A non-monotone data set
Option 1, Data reduction: A non-monotone data set
The maximum independent set problem

The non-monotonicity relation corresponds to a **comparability graph**:

- A monotone subset corresponds to an **independent set** of this graph.
- **Maximal independent set** = independent set that is not a subset of any other independent set.
- **Maximum independent set** (MIS) = independent set of biggest cardinality (= **independence number** \( \alpha \)).
- A MIS in a comparability graph can be determined using **network flow theory** (cubic time complexity).
Option 2, Data relabelling: which MIS to select?

\[ |2 - 1| + |3 - 1| = 3 \]
Option 2, Data relabelling: which MIS to select?

\[ |1 - 2| + |3 - 2| = 2 \]
Option 2, Data relabelling: options

Universal tool: weighted MIS problems and network flow theory

1. **Optimal ordinal relabelling**: relabelling a minimum number of objects, of which all corona objects are relabelled to a minimum extent.

2. **Optimal cardinal relabelling** (identifying $\mathcal{L}$ with the first $n$ integers): minimal relabelling loss
   - zero-one loss: MIS
   - broad class of loss functions, including L1 loss and squared loss

3. **Optimal hierarchical cardinal relabelling** (single pass):
   - minimizing loss while relabelling a minimal number of objects
   - relabelling a minimal number of objects while minimizing loss
Distribution representation of a data set

- Collection of learning examples \((S, d)\)
- For each \(x \in S_x\), a CDF \(\hat{F}(x, \cdot) : \mathcal{L} \rightarrow [0, 1]\) is built from the collection of learning examples

\[
\hat{F}(x, \ell) = \frac{|\{a \in S \mid a = x \land d(a) \leq \ell \}|}{|\{a \in S \mid a = x\}|}
\]

(cumulative relative frequency distribution)

- The distribution data set \((S_x, \hat{F})\)
A distribution data set
Stochastic minimal and maximal extensions

1. **Minimal Extension**: \( F_{\text{min}} : \mathcal{X} \times \mathcal{L} \rightarrow [0, 1] \)

\[
F_{\text{min}}(x, \ell) = \min\{\hat{F}(s, \ell) \mid s \in S_x \land s \leq_x x\}
\]

- if no such object: \( f_{\text{min}}(x, \min(\mathcal{L})) = 1 \)

2. **Maximal Extension**: \( F_{\text{max}} : \mathcal{X} \times \mathcal{L} \rightarrow [0, 1] \)

\[
F_{\text{max}}(x, \ell) = \max\{\hat{F}(s, \ell) \mid s \in S_x \land x \leq_x s\}
\]

- if no such object: \( f_{\text{max}}(x, \max(\mathcal{L})) = 1 \)

Monotone distribution classifiers

1. \( F_{\text{min}} \) and \( F_{\text{max}} \) are monotone distribution classifiers

2. **Interpolation**: for any \( S \in [0, 1] \), the mapping

\[
\tilde{F} = SF_{\text{min}} + (1 - S)F_{\text{max}}
\]

is also a monotone distribution classifier
Monotone distribution data sets

- \((S\chi, \hat{F})\) is called **monotone** if for all \(x\) and \(y\) in \(S\chi\)
  \[
  x <_{\chi} y \Rightarrow \hat{F}(x, \cdot) \preceq_{SD} \hat{F}(y, \cdot)
  \]

- **Reversed preference:**
  \[
  x <_{\chi} y \text{ while not } \hat{F}(x, \cdot) \preceq_{SD} \hat{F}(y, \cdot)
  \]

**Monotone extensions**

If the distribution data set is monotone, then

1. \(F_{\min}\) and \(F_{\max}\) are monotone **extensions** of \(\hat{F}\) to \(\chi\)

2. any monotone extension \(F\) of \(d\) to \(\chi\):
   \[
   F_{\min}(y, \cdot) \preceq_{SD} F(y, \cdot) \preceq_{SD} F_{\max}(y, \cdot)
   \]
A non-monotone distribution data set
A non-monotone distribution data set
5. Reversed preference revisited

How to handle noise?

1. **Data reduction**: *identify* the noisy distributions and *delete* them
   - the non-monotonicity relation is **not transitive** (MIS problem is NP-complete)
   - deleting entire distributions is quite invasive
   - deleting a single instance affects the entire distribution and is hard to realize

2. **Data relabelling**: *identify* the noisy distributions and *modify* them
   - transitivity of non-monotonicity still holds at the label level
   - **L1-optimal relabelling** is possible using network flow algorithms
   - does not affect the frequency of feature vectors

3. **Non-invasive approach**: keep the data set as is
After relabelling: a monotone distribution data set
A non-invasive approach

- Aim: to build a monotone distribution classifier from a possibly non-monotone distribution data set
- Weighted sums of $F_{\min}$ and $F_{\max}$ are solutions to this problem
- Aim: to identify more general interpolation schemes, depending on both the element $x$ and the label $\ell$
- For given $x$ and $\ell$:
  - **monotone situation**: $F_{\min}(x, \ell) \geq F_{\max}(x, \ell)$
  - **reversed preference situation**: $F_{\min}(x, \ell) < F_{\max}(x, \ell)$
The main theorem

OSDL generic theorem

Given two \( \mathcal{X} \times \mathcal{L} \rightarrow [0, 1] \) mappings \( s \) and \( t \), the mapping
\[
\tilde{F} : \mathcal{X} \times \mathcal{L} \rightarrow [0, 1]
\]

\[
\tilde{F}(x, \ell) = \begin{cases} 
  s(x, \ell)F_{\min}(x, \ell) + (1 - s(x, \ell))F_{\max}(x, \ell) & \text{if } F_{\min}(x, \ell) \geq F_{\max}(x, \ell) \\
  t(x, \ell)F_{\min}(x, \ell) + (1 - t(x, \ell))F_{\max}(x, \ell) & \text{if } F_{\min}(x, \ell) < F_{\max}(x, \ell)
\end{cases}
\]

is a monotone distribution classifier if and only if

1. \( s \) is decreasing in 1st and increasing in 2nd argument
2. \( t \) is increasing in 1st and decreasing in 2nd argument
The main theorem: realizations

Several realizations

1. **OSDL**: if one does not want to distinguish between the monotone and the reversed preference situation (s and t are identical), then the simple interpolation scheme is the only one.

2. **Balanced and Double-balanced OSDL**: use as weighing functions measures of support that count:
   - the number of instances that indicate that $x$ should receive a label strictly greater than $\ell$
   - the number of instances that indicate that $x$ should receive a label at most $\ell$
Epilogue
Concluding observations

1. In many modelling problems, there exists a **monotone relationship** between some or all of the **input variables** and the **output variable** that has to be accounted for.

2. **Mamdani–Assilian fuzzy models** for one-shot decisions should be abandoned.

3. Aggregation theory needs a **reboost**.

4. **Resolution of non-monotonicity** can be translated into an optimization problem (network flow theory).

5. Loyalty to the credo of fuzzy set theory (**“First process the data, then defuzzify”**) urges us to develop new mathematics.
References


Aggregation theory


References

Machine learning


Machine learning

Relabelling:


Monotone data set generation:

Merci pour votre attention