Fuzzy Preference Learning

Dariush Hasanpoor

Isfahan University Of Technology

May 24 2015

Outline

1. Introduction
   - What is preference?
   - Notations
   - Types of Ranking
   - Special Cases of preference learning

2. Fuzzy Preference Learning
   - Choquet Integral
   - Fuzzy Preference Learning Using Choquet Integral
   - Results

3. References

What is preference?

Definition

Preference Learning refers to the task of learning to predict an order relation on a collection of objects (alternatives).

- Preference information plays a key role in automated decision making and appears in various guises in AI researches:
    - Qualitative decision theory
    - Non-monotonic reasoning
    - Constraint satisfaction
    - Planning

Notations

Definition: *Weak Preference*

A weak *preference* relation $\succeq$ on a set $\mathcal{A}$ is a reflexive and transitive binary relation.

Definition: *Strict Preference*

$$a \succ b \longleftrightarrow (a \succeq b) \wedge (b \not\succeq a)$$

- In agreement with preference semantics

Notation	*Interpretation*
$a \succeq b$	"alternative a is at least as prefered as alternative b."
$a \succ b$	"alternative a is prefered over alternative b."

Notations

Definition: *Weak Preference*

A weak *preference* relation $\succeq$ on a set $\mathcal{A}$ is a reflexive and transitive binary relation.

Definition: *Strict Preference*

$$a \succ b \longleftrightarrow (a \succeq b) \wedge (b \not\succeq a)$$

- In agreement with preference semantics

Notation	*Interpretation*
$a \succeq b$	"alternative a is at least as prefered as alternative b."
$a \succ b$	"alternative a is prefered over alternative b."

Notations

Definition: *Weak Preference*

A weak *preference* relation $\succeq$ on a set $\mathcal{A}$ is a reflexive and transitive binary relation.

Definition: *Strict Preference*

$$a \succ b \longleftrightarrow (a \succeq b) \land (b \not\succeq a)$$

- In agreement with preference semantics

Notation	*Interpretation*
$a \succeq b$	"alternative a is at least as prefered as alternative b."
$a \succ b$	"alternative a is prefered over alternative b."

Preference Structure

Definition: *Total Strict Order (Ranking)*

If $\mathcal{A}$ is a finit set of objects/alternatives $\{a_1, \ldots, a_m\}$ a ranking of $\mathcal{A}$ can be definied with a permutation τ of $\{1, \ldots, m\}$ which $a_i \succ a_j \leftrightarrow \tau(i) < \tau(j)$.

- $\mathcal{S}_m$ is a set of all permutation of τ.
- The task of *preference learner* is to search in $\mathcal{S}_m$ space which is *learning to rank*.

Preference Structure

Definition: *Total Strict Order (Ranking)*

If $\mathcal{A}$ is a finit set of objects/alternatives $\{a_1, \ldots, a_m\}$ a ranking of $\mathcal{A}$ can be definied with a permutation τ of $\{1, \ldots, m\}$ which $a_i \succ a_j \leftrightarrow \tau(i) < \tau(j)$.

- $\mathcal{S}_m$ is a set of all permutation of τ.
- The task of *preference learner* is to search in $\mathcal{S}_m$ space which is *learning to rank*.

Types of Ranking

- The tasks are categorized as three main problems:
  - **Label ranking**
  - **Object ranking**
  - **Instance ranking**

Task

The task of this model is to find a preference ranking among the labels for any instance.

Given

- A set of training instances $\{x_k \mid k = 1, \ldots, n\} \subseteq \mathcal{X}$.
- A set of labels $\mathcal{L} = \{\lambda_i \mid i = 1, \ldots, m\}$.
- For each training instance x_k: a set of associated pairwise preferences of the form $\lambda_i \succ_{x_k} \lambda_j$.

Find

- A ranking function in the form of an $\mathcal{X} \to \mathcal{S}_m$ mapping that assigns a ranking (permutation) $\succ_x$ of $\mathcal{L}$ to every $x \in \mathcal{X}$.

Task

The task of this model is to find a preference ranking among the labels for any instance.

Given

- A set of training instances $\{x_k \mid k = 1, \ldots, n\} \subseteq \mathcal{X}$.
- A set of labels $\mathcal{L} = \{\lambda_i \mid i = 1, \ldots, m\}$.
- For each training instance x_k: a set of associated pairwise preferences of the form $\lambda_i \succ_{x_k} \lambda_j$.

Find

- A ranking function in the form of an $\mathcal{X} \to \mathcal{S}_m$ mapping that assigns a ranking (permutation) $\succ_x$ of $\mathcal{L}$ to every $x \in \mathcal{X}$.

Task

The task of this model is to find a preference ranking among the labels for any instance.

Given

- A set of training instances $\{x_k \mid k = 1, \ldots, n\} \subseteq \mathcal{X}$.
- A set of labels $\mathcal{L} = \{\lambda_i \mid i = 1, \ldots, m\}$.
- For each training instance x_k: a set of associated pairwise preferences of the form $\lambda_i \succ_{x_k} \lambda_j$.

Find

- A ranking function in the form of an $\mathcal{X} \to \mathcal{S}_m$ mapping that assigns a ranking (permutation) $\succ_x$ of $\mathcal{L}$ to every $x \in \mathcal{X}$.

Task

The task of this model is to find a preference ranking order among instances.

Given

- A (potentially infinite) set X of objects (each object typically represented by a feature vector).
- A finite set of pairwise preferences $x_i \succ x_j$, $(x_i, x_j) \in \mathcal{X} \times \mathcal{X}$.

Find

- A ranking function that, given a set of objects $O \subset X$ as input, returns a permutation(ranking) of these objects.

- In the training phase, preference learning algorithms have access to examples for which the order relation is **(partially)** known.

Task

The task of this model is to find a preference ranking order among instances.

Given

- A (potentially infinite) set X of objects (each object typically represented by a feature vector).
- A finite set of pairwise preferences $x_i \succ x_j$, $(x_i, x_j) \in \mathcal{X} \times \mathcal{X}$.

Find

- A ranking function that, given a set of objects $O \subset X$ as input, returns a permutation(ranking) of these objects.

- In the training phase, preference learning algorithms have access to examples for which the order relation is **(partially)** known.

Types of Ranking

Object Ranking

Task

The task of this model is to find a preference ranking order among instances.

Given

- A (potentially infinite) set X of objects (each object typically represented by a feature vector).
- A finite set of pairwise preferences $x_i \succ x_j$, $(x_i, x_j) \in \mathcal{X} \times \mathcal{X}$.

Find

- A ranking function that, given a set of objects $O \subset X$ as input, returns a permutation(ranking) of these objects.

- In the training phase, preference learning algorithms have access to examples for which the order relation is **(partially)** known.

Types of Ranking
Object Ranking

Task
The task of this model is to find a preference ranking order among instances.

Given

- A (potentially infinite) set X of objects (each object typically represented by a feature vector).
- A finite set of pairwise preferences $x_i \succ x_j$, $(x_i, x_j) \in \mathcal{X} \times \mathcal{X}$.

Find

- A ranking function that, given a set of objects $O \subset X$ as input, returns a permutation(ranking) of these objects.

- In the training phase, preference learning algorithms have access to examples for which the order relation is **(partially)** known.

Types of Ranking

Instance Ranking

Task

The task of this model is to find a preference ranking order among instances.

Given

- A (potentially infinite) set X of objects (each object typically represented by a feature vector).
- A finite set of pairwise preferences $x_i \succ x_j$, $(x_i, x_j) \in \mathcal{X} \times \mathcal{X}$.
- An order set of labels $\mathcal{L} = \{\lambda_i \mid i = 1, \ldots, m\}$ which $y_1 \succ y_2 \succ \ldots \succ y_m$.
- A set of $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{L}$ which each instance x_k is associated with a label $\lambda_{\hat{k}}$.

Find

- A ranking function that, given a set of objects $O \subset X$ as input, returns a permutation(ranking) of these objects.

- ~~In the training phase, preference learning algorithms have access to examples for which the order relation is **(partially)** known.~~

Special Cases of preference learning

▷ Classification

A single class label λ_i is assigned to each example x_k. This is equivalent to the set of preferences $\{\lambda_i \succ_{x_k} \lambda_j \mid 1 \leq j \neq i \geq m\}$.

▷ Multi-label classification

Each training example x_k is associated with a subset $\mathcal{L}_k \subseteq \mathcal{L}$ of possible labels. This is equivalent to the set of preferences $\{\lambda_i \succ_{x_k} \lambda_j \mid \lambda_i \in \mathcal{L}_k,\ \lambda_j \in \mathcal{L} \backslash \mathcal{L}_k\}$.

Special Cases of preference learning

▷ Classification

A single class label λ_i is assigned to each example x_k. This is equivalent to the set of preferences $\{\lambda_i \succ_{x_k} \lambda_j \mid 1 \leq j \neq i \geq m\}$.

▷ Multi-label classification

Each training example x_k is associated with a subset $\mathcal{L}_k \subseteq \mathcal{L}$ of possible labels. This is equivalent to the set of preferences $\{\lambda_i \succ_{x_k} \lambda_j \mid \lambda_i \in \mathcal{L}_k,\ \lambda_j \in \mathcal{L} \backslash \mathcal{L}_k\}$.

Choquet Integral

An Example:

- Suppose a school is more scientifically than literary oriented.
- How can we compare these 3 students?

Student	*Math*	*Physics*	*Literature*
a	18	16	10
b	10	12	18
c	14	15	15

- A candidate set of weights can be $\{\frac{3}{8}, \frac{3}{8}, \frac{2}{8}\}$.
- But what if the school wants to favor well equilibrated students without weak points?
  - Then the student c should be considered better than student a and b.
  - This cannot be simply done by simple weighting sum procedure!!
- So how are we going rank them?!
  - Choquet integral can address such problems.

Choquet Integral

An Example:

- Suppose a school is more scientifically than literary oriented.
- How can we compare these 3 students?

Student	*Math*	*Physics*	*Literature*
a	18	16	10
b	10	12	18
c	14	15	15

- A candidate set of weights can be $\{\frac{3}{8}, \frac{3}{8}, \frac{2}{8}\}$.
- But what if the school wants to favor well equilibrated students without weak points?
  - Then the student c should be considered better than student a and b.
  - This cannot be simply done by simple weighting sum procedure!!
- So how are we going rank them?!
  - Choquet integral can address such problems.

Choquet Integral

An Example:

- Suppose a school is more scientifically than literary oriented.
- How can we compare these 3 students?

Student	*Math*	*Physics*	*Literature*
a	18	16	10
b	10	12	18
c	14	15	15

- A candidate set of weights can be $\{\frac{3}{8},\ \frac{3}{8},\ \frac{2}{8}\}$.
- But what if the school wants to favor well equilibrated students without weak points?
  - Then the student *c* should be considered better than student *a* and *b*.
  - This cannot be simply done by simple weighting sum procedure!!
- So how are we going rank them?!
  - Choquet integral can address such problems.

Choquet Integral

An Example:

- Suppose a school is more scientifically than literary oriented.
- How can we compare these 3 students?

Student	*Math*	*Physics*	*Literature*
a	18	16	10
b	10	12	18
c	14	15	15

- A candidate set of weights can be $\{\frac{3}{8}, \frac{3}{8}, \frac{2}{8}\}$.
- But what if the school wants to favor well equilibrated students without weak points?
  - Then the student c should be considered better than student a and b.
  - This cannot be simply done by simple weighting sum procedure!!
- So how are we going rank them?!
  - Choquet integral can address such problems.

Choquet Integral

An Example:

- Suppose a school is more scientifically than literary oriented.
- How can we compare these 3 students?

Student	*Math*	*Physics*	*Literature*
a	18	16	10
b	10	12	18
c	14	15	15

- A candidate set of weights can be $\{\frac{3}{8}, \frac{3}{8}, \frac{2}{8}\}$.
- But what if the school wants to favor well equilibrated students without weak points?
  - Then the student *c* should be considered better than student *a* and *b*.
  - This cannot be simply done by simple weighting sum procedure!!
- So how are we going rank them?!
  - Choquet integral can address such problems.

Choquet Integral

Continued...

- Choquet Integral

$$\mathcal{C}_\mu(x) = \sum_{i=1}^{n} (x_{\tau(i)} - x_{\tau(i-1)}) \mu(\{\tau(i), \ldots, \tau(n)\})$$

$x_{\tau(1)} \leq x_{\tau(2)} \leq \ldots \leq x_{\tau(n)}, \; x_{\tau(0)} = 0$

Choquet Integral

Continued...

Student	*Math(1)*	*Physics(2)*	*Literature(3)*	$\{\tau\}$	$\{x_\tau\}$
a	18	16	10	$\{0,\ 3,\ 2,\ 1\}$	$\{0,\ 10,\ 16,\ 18\}$
b	10	12	18	$\{0,\ 1,\ 2,\ 3\}$	$\{0,\ 10,\ 12,\ 18\}$
c	14	15	15	$\{0,\ 1,\ 2,\ 3\}$	$\{0,\ 14,\ 15,\ 15\}$

$$\mathcal{C}_\mu(x) = \sum_{i=1}^{n} (x_{\tau(i)} - x_{\tau(i-1)})\mu(\{\tau(i), \ldots, \tau(n)\})$$

$$\mu(\{1,\ 2,\ 3\}) = 1, \qquad \mu(\{\emptyset\}) = 0$$

$$\mu(\{1\}) = \mu(\{2\}) = 0.45, \qquad \mu(\{3\}) = 0.3$$

$$\mu(\{1,\ 3\}) = \mu(\{2,\ 3\}) = 0.9, \qquad \mu(\{1,\ 2\}) = 0.5$$

Choquet Integral

Continued...

Student	*Math(1)*	*Physics(2)*	*Literature(3)*	$\{\tau\}$	$\{x_\tau\}$
a	18	16	10	$\{0,\ 3,\ 2,\ 1\}$	$\{0,\ 10,\ 16,\ 18\}$
b	10	12	18	$\{0,\ 1,\ 2,\ 3\}$	$\{0,\ 10,\ 12,\ 18\}$
c	14	15	15	$\{0,\ 1,\ 2,\ 3\}$	$\{0,\ 14,\ 15,\ 15\}$

$$\mathcal{C}_\mu(x) = \sum_{i=1}^{n} (x_{\tau(i)} - x_{\tau(i-1)})\mu(\{\tau(i), \ldots, \tau(n)\})$$

$$\mu(\{1,\ 2,\ 3\}) = 1, \qquad \mu(\{\emptyset\}) = 0$$

$$\mu(\{1\}) = \mu(\{2\}) = 0.45, \qquad \mu(\{3\}) = 0.3$$

$$\mu(\{1,\ 3\}) = \mu(\{2,\ 3\}) = 0.9, \qquad \mu(\{1,\ 2\}) = 0.5$$

$$\mathcal{C}_\mu(a) = (10-0) \times \mu(\{3,2,1\}) + (16-10) \times \mu(\{2,1\}) + (18-16) \times \mu(\{1\}) = 13.9$$
$$\mathcal{C}_\mu(b) = (10-0) \times \mu(\{1,2,3\}) + (12-10) \times \mu(\{2,3\}) + (18-12) \times \mu(\{3\}) = 13.6$$
$$\mathcal{C}_\mu(c) = (14-0) \times \mu(\{3,2,1\}) + (15-14) \times \mu(\{2,3\}) + (15-15) \times \mu(\{3\}) = 14.9$$

Choquet Integral

Continued...

Student	*Math*	*Physics*	*Literature*	Weighted sum	Choquet integral
a	18	16	10	15.25	13.9
b	10	12	18	12.75	13.6
c	14	15	15	14.62	14.9

Non-additive Measures

- As in previous example, we had $\mu(\{1\}) = \mu(\{2\}) = 0.45,\ \mu(\{1,\ 2\}) = 0.5$.
  - We can see $\mu(\{1,\ 2\}) \neq \mu(\{1\}) + \mu(\{2\})$.

Definition: Non-additive(Capacities) Measures

Let $\mathcal{X} = \{x_1,\ \ldots,\ x_n\}$ be a finit set and μ a measure $2^{\mathcal{X}} \to [0,\ 1]$ if there are $A,\ B \subseteq \mathcal{X} \mid A \cap B = \emptyset$ such that $\mu(\{A, B\}) \neq \mu(\{A\}) + \mu(\{B\})$ then the measure μ is a non-additive or capacity fuzzy measure.

- Non-additive measures are normalized and monotone.

  $\mu(\emptyset) = 0,\ \mu(\mathcal{X}) = 1$ and

  $\mu(A) \leq \mu(B) \quad \forall\, A \subseteq B \subseteq \mathcal{X}$

- Choquet Integral combines non-additive measures in a desirable way.

Non-additive Measures

- As in previous example, we had $\mu(\{1\}) = \mu(\{2\}) = 0.45,\ \mu(\{1,\ 2\}) = 0.5$.
  - We can see $\mu(\{1,\ 2\}) \neq \mu(\{1\}) + \mu(\{2\})$.

Definition: Non-additive(Capacities) Measures

Let $\mathcal{X} = \{x_1,\ \ldots,\ x_n\}$ be a finit set and μ a measure $2^{\mathcal{X}} \to [0,\ 1]$ if there are $A,\ B \subseteq \mathcal{X} \mid A \cap B = \emptyset$ such that $\mu(\{A, B\}) \neq \mu(\{A\}) + \mu(\{B\})$ then the measure μ is a non-additive or capacity fuzzy measure.

- Non-additive measures are normalized and monotone.

  $\mu(\emptyset) = 0,\ \mu(\mathcal{X}) = 1$ and

  $\mu(A) \leq \mu(B) \quad \forall\, A \subseteq B \subseteq \mathcal{X}$

- Choquet Integral combines non-additive measures in a desirable way.

Non-additive Measures

- As in previous example, we had $\mu(\{1\}) = \mu(\{2\}) = 0.45, \ \mu(\{1, \ 2\}) = 0.5$.
  - We can see $\mu(\{1, \ 2\}) \neq \mu(\{1\}) + \mu(\{2\})$.

Definition: Non-additive(Capacities) Measures

Let $\mathcal{X} = \{x_1, \ \ldots, \ x_n\}$ be a finit set and μ a measure $2^{\mathcal{X}} \rightarrow [0, \ 1]$ if there are $A, \ B \subseteq \mathcal{X} \mid A \cap B = \emptyset$ such that $\mu(\{A, B\}) \neq \mu(\{A\}) + \mu(\{B\})$ then the measure μ is a non-additive or capacity fuzzy measure.

- Non-additive measures are normalized and monotone.

  $\mu(\emptyset) = 0, \ \mu(\mathcal{X}) = 1$ and

  $\mu(A) \leq \mu(B) \quad \forall \, A \subseteq B \subseteq \mathcal{X}$

- Choquet Integral combines non-additive measures in a desirable way.

Definition: Möbius transform

$$\mu(B) = \sum_{A \subseteq B} m(A)$$

$$m_{\hat{\mu}}(A) = \sum_{v \subseteq A} (-1)^{|A|-|v|} \hat{\mu}(v)$$

Definition: Möbius transform

$$\mu(B) = \sum_{A \subseteq B} m(A)$$
$$m_{\hat{\mu}}(A) = \sum_{v \subseteq A} (-1)^{|A|-|v|} \hat{\mu}(v)$$

Example

$$m_{\hat{\mu}}(\{\emptyset\}) = \hat{\mu}(\emptyset)$$
$$m_{\hat{\mu}}(\{1\}) = \hat{\mu}(\{1\}) - \hat{\mu}(\emptyset)$$
$$m_{\hat{\mu}}(\{1,2\}) = \hat{\mu}(\{1,2\}) - \hat{\mu}(\{1\}) - \hat{\mu}(\{2\}) + \hat{\mu}(\emptyset)$$

Example

$$
\begin{aligned}
m_{\hat{\mu}}(\{\emptyset\}) &= \hat{\mu}(\emptyset) \\
m_{\hat{\mu}}(\{1\}) &= \hat{\mu}(\{1\}) - \hat{\mu}(\emptyset) \\
m_{\hat{\mu}}(\{1,2\}) &= \hat{\mu}(\{1,2\}) - \hat{\mu}(\{1\}) - \hat{\mu}(\{2\}) + \hat{\mu}(\emptyset)
\end{aligned}
$$

- The value $m_{\hat{\mu}}(A)$ can be interpreted as the weight that is exclusively allocated to A, instead of being indirectly connected with A through the interaction with other subsets.

Fuzzy Preference Learning Using Choquet Integral

- Suppose that $f : \mathcal{X} \to \mathbb{R}_+$ is any nonnegative function that assigns a *value* to each criterion x_i for any object.
- **Question:** How to aggregate the evaluations of individual criteria?
- **Answer:** This overall evaluation can be considered as an integral $\mathcal{C}_\mu(f)$ of the function f with respect to the measure μ.
- This article has focued on object ranking problem.

Fuzzy Preference Learning Using Choquet Integral

- Suppose that $f : \mathcal{X} \to \mathbb{R}_+$ is any nonnegative function that assigns a *value* to each criterion x_i for any object.
- **Question:** How to aggregate the evaluations of individual criteria?
- **Answer:** This overall evaluation can be considered as an integral $\mathcal{C}_\mu(f)$ of the function f with respect to the measure μ.
- This article has focued on object ranking problem.

Fuzzy Preference Learning Using Choquet Integral

- Suppose that $f : \mathcal{X} \to \mathbb{R}_+$ is any nonnegative function that assigns a *value* to each criterion x_i for any object.
- **Question:** How to aggregate the evaluations of individual criteria?
- **Answer:** This overall evaluation can be considered as an integral $\mathcal{C}_\mu(f)$ of the function f with respect to the measure μ.
- This article has focued on object ranking problem.

Fuzzy Preference Learning Using Choquet Integral

- Suppose that $f : \mathcal{X} \to \mathbb{R}_+$ is any nonnegative function that assigns a *value* to each criterion x_i for any object.
- **Question:** How to aggregate the evaluations of individual criteria?
- **Answer:** This overall evaluation can be considered as an integral $\mathcal{C}_\mu(f)$ of the function f with respect to the measure μ.
- This article has focued on object ranking problem.

Fuzzy Preference Learning Using Choquet Integral

Continued...

$$
\begin{aligned}
\mathcal{C}_{\hat{\mu}}(f) &= \sum_{i=1}^{n} (f(x_{(i)}) - f(x_{(i-1)})) \cdot \mu(\overbrace{\{x_{(i)},\ \ldots,\ x_{(n)}\}}^{A_{(i)}}) \\
&= \sum_{i=1}^{n} f(x_{(i)}) \cdot (\mu(A_{(i)}) - \mu(A_{(i+1)})) \\
&\overset{\text{MT}}{=} \sum_{i=1}^{n} f(x_{(i)}) \cdot \sum_{R \subseteq \mathcal{T}} m(R) \mid \mathcal{T} = \{\mathcal{G} \cup \{x_{(i)}\} \mid \mathcal{G} \subset \{x_{(i+1)},\ \ldots,\ x_{(n)}\}\} \\
&= \sum_{T \subseteq \mathcal{X}} m(T) \times \min_{x_i \in T} f(x_i) \\
&= \sum_{T \subseteq \mathcal{X}} \sum_{v \subseteq T} (-1)^{|A|-|v|} \hat{\mu}(v) \times \min_{x_i \in T} f(x_i)
\end{aligned}
$$

Fuzzy Preference Learning Using Choquet Integral

Continued...

Description of approach

- Training data to be available in the form of a set of objects o_i.
- Every object o_i is linked to a corresponding label information l_i.
- A set $\mathcal{D}$ is constructed: $(o_i,\ o_j) \in \mathcal{D}$ suggesting that $o_i \succ o_j \mid l_i > l_j$.
- The Choquet integral is uniquely identified by the underlying measure μ on the set of criteria $\mathcal{X}$, the problem comes down to defining μ.
- Finding μ measure is actually a optimization problem!

Experiments

Datebases

DATASETS AND THEIR PROPERTIES

data set	#instances	#attributes	#classes
Color (CLR) 1–7	120	3	3
Scientific Journals (SCJ)	172	5	4
CPU	209	6	2
Auto MPG	398	8	6
Employee Selection (ESL)	488	4	9
Mamographic (MMG)	830	5	2
Lecturers Evaluation (LEV)	1000	4	5
Concrete Compressive Strength (CCS)	1030	8	6
Car Evaluation (CEV)	1728	6	4

PERFORMANCE IN TERMS OF THE AVERAGE C-INDEX ± STANDARD DEVIATION

data set	WM	PL d=1	PL d=2	PL d=3	CI
CLR 1	.9663±.0148(4)	.9506±.0155(5)	.9674±.0129(3)	.9700±.0141(2)	.9828±.0090(1)
CLR 2	.8740±.0293(4)	.8601±.0294(5)	.8876±.0200(3)	.9341±.0244(2)	.9804±.0128(1)
CLR 3	.9343±.0204(4)	.9268±.0219(5)	.9375±.0156(3)	.9633±.0143(2)	.9878±.0150(1)
CLR 4	.9357±.0171(4)	.9228±.0247(5)	.9431±.0189(3)	.9659±.0166(2)	.9915±.0056(1)
CLR 5	.9518±.0194(3)	.9485±.0179(5)	.9565±.0142(2)	.9516±.0171(4)	.9682±.0140(1)
CLR 6	.9046±.0202(4)	.8923±.0205(5)	.9127±.0201(3)	.9460±.0191(2)	.9825±.0121(1)
CLR 7	.8880±.0312(4)	.8797±.0256(5)	.8892±.0219(3)	.9258±.0237(2)	.9688±.0167(1)
SCJ	.8168±.0105(4)	.8098±.0112(5)	.8270±.0241(3)	.8313±.0109(2)	.8450±.0201(1)
CPU	.9965±.0027(3)	.9950±.0093(5)	.9978±.0012(2)	.9955±.0005(4)	.9986±.0014(1)
MPG	.8887±.0176(4)	.8850±.0143(5)	.8912±.0078(3)	.8967±.0093(2)	.9060±.0111(1)
ESL	.9497±.0162(2)	.9559±.0071(1)	.9465±.0104(4)	.9491±.0126(3)	.9424±.0098(5)
MMG	.8961±.0230(2)	.8536±.0168(4)	.8714±.0181(3)	.7813±.0350(5)	.9015±.0210(1)
LEV	.8710±.0289(2)	.8620±.0320(3)	.8713±.0250(1)	.8527±.0300(5)	.8610±.0320(4)
CCS	.8650±.0068(4)	.8586±.0102(5)	.8862±.0184(3)	.8962±.0203(2)	.9050±.0038(1)
CEV	.8981±.0066(4)	.8804±.0076(5)	.9118±.0059(3)	.9585±.0090(2)	.9771±.0039(1)
average rank	3.47	4.53	2.8	2.73	1.47

Additionally, the rank of each method it shown in brackets.

Compared the approach with kernel-based methods.

- Spider implementation of the RankSVM approach with a linear and a polynomial kernel(PL) with kernel degree of d.
- Weighted Mean(WM).

WIN STATISTICS (NUMBER OF DATASETS ON WHICH THE FIRST METHOD WAS BETTER THAN THE SECOND ONE)

	WM	PL d=1	PL d=2	PL d=3	CI
WM	–	14	2	5	2
PL d=1	1	–	1	3	2
PL d=2	13	14	–	4	2
PL d=3	10	12	11	–	1
CI	13	13	13	14	–

Compared the approach with kernel-based methods.

- Spider implementation of the RankSVM approach with a linear and a polynomial kernel(PL) with kernel degree of d.
- Weighted Mean(WM).

References

Dr. Safayani, "Fuzzy integral lecture," 2015.

A. F. Tehrani, W. Cheng, and E. Hullermeier, "Preference Learning Using the Choquet Integral: The Case of Multipartite Ranking," in *Fuzzy Systems*. IEEE Transactions, Dec. 2012, vol. 20, pp. 1102–1113.

J. Furnkranz and E. Hullermeier, *Encyclopedia of Machine Learning*. Springer, 2010, ch. Preference Learning, pp. 789–795.

Johannes Furnkranz and Eyke Hullermeier, "Preference Learning," *Kunstliche Intelligenz*, pp. 60–61, 2005.

E. Hullermeier, J. Furnkranz, W. Cheng, and K. Brinker, *Artificial Intelligence*. ScienceDirect, 2008, ch. Label ranking by learning pairwise preferences.

- A natural way to represent preferences is to evaluate the alternatives by means of a utility function.

Object Preferences Senario

- ▷ Such a function is a mapping $\mathcal{F} : \mathcal{X} \to \mathcal{U}$ that assigns a utility degree $\mathcal{F}(x)$ to each object x and, thereby, induces a linear order on $\mathcal{X}$.

Label Preferences Senario

- ▷ A utility function $\mathcal{F}_i : \mathcal{X} \to \mathcal{U}$ is needed for every label $\lambda_i, i = , \ldots, m$.
- ▷ $\mathcal{F}(x)$ is the utility assigned to alternative λ_i by instance x.
- ▷ A ranking $\succ_x$ is derived that satisfies $\lambda_i \succ_x \lambda_j \to \mathcal{F}_i(x) \geq \mathcal{F}_j(x)$.

- A natural way to represent preferences is to evaluate the alternatives by means of a utility function.

Object Preferences Senario

- ▷ Such a function is a mapping $\mathcal{F} : \mathcal{X} \to \mathcal{U}$ that assigns a utility degree $\mathcal{F}(x)$ to each object x and, thereby, induces a linear order on $\mathcal{X}$.

Label Preferences Senario

- ▷ A utility function $\mathcal{F}_i : \mathcal{X} \to \mathcal{U}$ is needed for every label $\lambda_i, i = , \ldots, m$.
- ▷ $\mathcal{F}(x)$ is the utility assigned to alternative λ_i by instance x.
- ▷ A ranking $\succ_x$ is derived that satisfies $\lambda_i \succ_x \lambda_j \to \mathcal{F}_i(x) \geq \mathcal{F}_j(x)$.

- A natural way to represent preferences is to evaluate the alternatives by means of a utility function.

Object Preferences Senario

▷ Such a function is a mapping $\mathcal{F} : \mathcal{X} \to \mathcal{U}$ that assigns a utility degree $\mathcal{F}(x)$ to each object x and, thereby, induces a linear order on $\mathcal{X}$.

Label Preferences Senario

▷ A utility function $\mathcal{F}_i : \mathcal{X} \to \mathcal{U}$ is needed for every label $\lambda_i, i = , \ldots , m$.

▷ $\mathcal{F}(x)$ is the utility assigned to alternative λ_i by instance x.

▷ A ranking $\succ_x$ is derived that satisfies $\lambda_i \succ_x \lambda_j \to \mathcal{F}_i(x) \geq \mathcal{F}_j(x)$.

- A natural way to represent preferences is to evaluate the alternatives by means of a utility function.

Object Preferences Senario

- ▷ Such a function is a mapping $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{U}$ that assigns a utility degree $\mathcal{F}(x)$ to each object x and, thereby, induces a linear order on $\mathcal{X}$.

Label Preferences Senario

- ▷ A utility function $\mathcal{F}_i : \mathcal{X} \rightarrow \mathcal{U}$ is needed for every label $\lambda_i, i = , \ldots , m$.
- ▷ $\mathcal{F}(x)$ is the utility assigned to alternative λ_i by instance x.
- ▷ A ranking $\succ_x$ is derived that satisfies $\lambda_i \succ_x \lambda_j \rightarrow \mathcal{F}_i(x) \geq \mathcal{F}_j(x)$.

- A natural way to represent preferences is to evaluate the alternatives by means of a utility function.

Object Preferences Senario

- ▷ Such a function is a mapping $\mathcal{F} : \mathcal{X} \to \mathcal{U}$ that assigns a utility degree $\mathcal{F}(x)$ to each object x and, thereby, induces a linear order on $\mathcal{X}$.

Label Preferences Senario

- ▷ A utility function $\mathcal{F}_i : \mathcal{X} \to \mathcal{U}$ is needed for every label $\lambda_i, i = , \ldots, m$.
- ▷ $\mathcal{F}(x)$ is the utility assigned to alternative λ_i by instance x.
- ▷ A ranking $\succ_x$ is derived that satisfies $\lambda_i \succ_x \lambda_j \to \mathcal{F}_i(x) \geq \mathcal{F}_j(x)$.

- An alternative approach to preference learning consists of comparing pairs of objects(labels) in terms of a binary preference relation.

Object Preferences Senario

- ▷ Learning a binary preference predicate $Q(x, x')$, which predicts whether x is preferred to x' or vice versa.
- ▷ A final ordering is found in a second phase by deriving a ranking that is maximally consistent with these predictions.

Label Preferences Senario

- ▷ One can train a separate model(base learner) $\mathcal{M}_{i,j}$ for each pair of labels $(\lambda_i, \lambda_j) \in \mathcal{L},\ 1 \leq i < j \leq m$; thus, a total number of $\frac{m(m-1)}{2}$ models is needed.
- ▷ For training, a preference information of the form $\lambda_i \succ_x \lambda_j$ is turned into a (classification) example (x, y) for the learner $\mathcal{M}_{a,b}$, where $a = \min(i, j)$ and $b = \max(i, j)$.
- ▷ $\mathcal{M}_{a,b}$ is intended to learn the mapping that outputs 1 if $\lambda_a \succ_x \lambda_b$ and 0 if $\lambda_b \succ_x \lambda_a$.

- An alternative approach to preference learning consists of comparing pairs of objects(labels) in terms of a binary preference relation.

Object Preferences Senario

- Learning a binary preference predicate $\mathcal{Q}(x, x')$, which predicts whether x is preferred to x' or vice versa.
- A final ordering is found in a second phase by deriving a ranking that is maximally consistent with these predictions.

Label Preferences Senario

- One can train a separate model(base learner) $\mathcal{M}_{i,j}$ for each pair of labels $(\lambda_i, \lambda_j) \in \mathcal{L},\ 1 \leq i < j \leq m$; thus, a total number of $\frac{m(m-1)}{2}$ models is needed.
- For training, a preference information of the form $\lambda_i \succ_x \lambda_j$ is turned into a (classification) example (x, y) for the learner $\mathcal{M}_{a,b}$, where $a = \min(i, j)$ and $b = \max(i, j)$.
- $\mathcal{M}_{a,b}$ is intended to learn the mapping that outputs 1 if $\lambda_a \succ_x \lambda_b$ and 0 if $\lambda_b \succ_x \lambda_a$.

Techniques

Learning Preference Relations

- An alternative approach to preference learning consists of comparing pairs of objects(labels) in terms of a binary preference relation.

Object Preferences Senario

- ▷ Learning a binary preference predicate $\mathcal{Q}(x, x')$, which predicts whether x is preferred to x' or vice versa.
- ▷ A final ordering is found in a second phase by deriving a ranking that is maximally consistent with these predictions.

Label Preferences Senario

- ▷ One can train a separate model(base learner) $\mathcal{M}_{i,j}$ for each pair of labels $(\lambda_i, \lambda_j) \in \mathcal{L},\ 1 \leq i < j \leq m$; thus, a total number of $\frac{m(m-1)}{2}$ models is needed.
- ▷ For training, a preference information of the form $\lambda_i \succ_x \lambda_j$ is turned into a (classification) example (x, y) for the learner $\mathcal{M}_{a,b}$, where $a = \min(i, j)$ and $b = \max(i, j)$.
- ▷ $\mathcal{M}_{a,b}$ is intended to learn the mapping that outputs 1 if $\lambda_a \succ_x \lambda_b$ and 0 if $\lambda_b \succ_x \lambda_a$.

- An alternative approach to preference learning consists of comparing pairs of objects(labels) in terms of a binary preference relation.

Object Preferences Senario

- Learning a binary preference predicate $\mathcal{Q}(x, x')$, which predicts whether x is preferred to x' or vice versa.
- A final ordering is found in a second phase by deriving a ranking that is maximally consistent with these predictions.

Label Preferences Senario

- One can train a separate model(base learner) $\mathcal{M}_{i,j}$ for each pair of labels $(\lambda_i, \lambda_j) \in \mathcal{L}, \ 1 \leq i < j \leq m$; thus, a total number of $\frac{m(m-1)}{2}$ models is needed.
- For training, a preference information of the form $\lambda_i \succ_x \lambda_j$ is turned into a (classification) example (x, y) for the learner $\mathcal{M}_{a,b}$, where $a = \min(i, j)$ and $b = \max(i, j)$.
- $\mathcal{M}_{a,b}$ is intended to learn the mapping that outputs 1 if $\lambda_a \succ_x \lambda_b$ and 0 if $\lambda_b \succ_x \lambda_a$.

- An alternative approach to preference learning consists of comparing pairs of objects(labels) in terms of a binary preference relation.

Object Preferences Senario

- Learning a binary preference predicate $\mathcal{Q}(x, x')$, which predicts whether x is preferred to x' or vice versa.
- A final ordering is found in a second phase by deriving a ranking that is maximally consistent with these predictions.

Label Preferences Senario

- One can train a separate model(base learner) $\mathcal{M}_{i,j}$ for each pair of labels $(\lambda_i, \lambda_j) \in \mathcal{L},\ 1 \leq i < j \leq m$; thus, a total number of $\frac{m(m-1)}{2}$ models is needed.
- For training, a preference information of the form $\lambda_i \succ_x \lambda_j$ is turned into a (classification) example (x, y) for the learner $\mathcal{M}_{a,b}$, where $a = \min(i, j)$ and $b = \max(i, j)$.
- $\mathcal{M}_{a,b}$ is intended to learn the mapping that outputs 1 if $\lambda_a \succ_x \lambda_b$ and 0 if $\lambda_b \succ_x \lambda_a$.

- An alternative approach to preference learning consists of comparing pairs of objects(labels) in terms of a binary preference relation.

Object Preferences Senario

- Learning a binary preference predicate $\mathcal{Q}(x, x')$, which predicts whether x is preferred to x' or vice versa.
- A final ordering is found in a second phase by deriving a ranking that is maximally consistent with these predictions.

Label Preferences Senario

- One can train a separate model(base learner) $\mathcal{M}_{i,j}$ for each pair of labels $(\lambda_i, \lambda_j) \in \mathcal{L},\ 1 \leq i < j \leq m$; thus, a total number of $\frac{m(m-1)}{2}$ models is needed.
- For training, a preference information of the form $\lambda_i \succ_x \lambda_j$ is turned into a (classification) example (x, y) for the learner $\mathcal{M}_{a,b}$, where $a = \min(i, j)$ and $b = \max(i, j)$.
- $\mathcal{M}_{a,b}$ is intended to learn the mapping that outputs 1 if $\lambda_a \succ_x \lambda_b$ and 0 if $\lambda_b \succ_x \lambda_a$.

- The idea is to represent the latent utility function $\mathcal{U}(\cdot)$ in terms of a Choquet integral.
- Assuming that objects $o \in \mathcal{O}$ are represented as feature vectors:

$$f_o = (f_o(x_1),\ \ldots,\ f_o(x_n))$$

- $f_o(x_i)$ can be thought of as the evaluation of object o on the criterion x_i, So:

$$\mathcal{U}(o) = \mathcal{C}_\mu(f_o)$$

- The goal is to minimizes the number of ranking errors($\mathcal{C}$-index) on the training data $\mathcal{D}$.

$$C(U,O) = \frac{\sum_{1 \le i < j \le k} \sum_{(o,o\prime) \in O_i \times O_j} \mathcal{S}(\mathcal{U}(o), \mathcal{U}(o\prime))}{\sum_{i<j} |O_i|.|O_j|},\ \mathcal{S}(u,v) = \begin{cases} 1 & u < v \\ 0 & \text{otherwise} \end{cases}$$

Fuzzy Preference Learning Using Choquet Integral

Continued...

- The idea is to represent the latent utility function $\mathcal{U}(\cdot)$ in terms of a Choquet integral.
- Assuming that objects $o \in \mathcal{O}$ are represented as feature vectors:

$$f_o = (f_o(x_1),\ \ldots,\ f_o(x_n))$$

- $f_o(x_i)$ can be thought of as the evaluation of object o on the criterion x_i, So:

$$\mathcal{U}(o) = \mathcal{C}_\mu(f_o)$$

- The goal is to minimizes the number of ranking errors($\mathcal{C}$-index) on the training data $\mathcal{D}$.

$$C(U,O) = \frac{\sum\limits_{1 \le i < j \le k} \ \sum\limits_{(o,o\prime) \in O_i \times O_j} \mathcal{S}(\mathcal{U}(o), \mathcal{U}(o\prime))}{\sum\limits_{i<j} |O_i|.|O_j|},\ \mathcal{S}(u,v) = \begin{cases} 1 & u < v \\ 0 & \text{otherwise} \end{cases}$$

Fuzzy Preference Learning Using Choquet Integral

Continued...

- The idea is to represent the latent utility function $\mathcal{U}(\cdot)$ in terms of a Choquet integral.
- Assuming that objects $o \in \mathcal{O}$ are represented as feature vectors:

$$f_o = (f_o(x_1),\ \ldots,\ f_o(x_n))$$

- $f_o(x_i)$ can be thought of as the evaluation of object o on the criterion x_i, So:

$$\mathcal{U}(o) = \mathcal{C}_\mu(f_o)$$

- The goal is to minimizes the number of ranking errors($\mathcal{C}$-index) on the training data $\mathcal{D}$.

$$C(U,O) = \frac{\sum\limits_{1 \le i < j \le k} \sum\limits_{(o,o\prime) \in O_i \times O_j} \mathcal{S}(\mathcal{U}(o), \mathcal{U}(o\prime))}{\sum\limits_{i<j} |O_i|.|O_j|},\ \mathcal{S}(u,v) = \begin{cases} 1 & u < v \\ 0 & \text{otherwise} \end{cases}$$

Fuzzy Preference Learning Using Choquet Integral

Continued...

- The idea is to represent the latent utility function $\mathcal{U}(\cdot)$ in terms of a Choquet integral.
- Assuming that objects $o \in \mathcal{O}$ are represented as feature vectors:

$$f_o = (f_o(x_1),\ \ldots,\ f_o(x_n))$$

- $f_o(x_i)$ can be thought of as the evaluation of object o on the criterion x_i, So:

$$\mathcal{U}(o) = \mathcal{C}_\mu(f_o)$$

- The goal is to minimizes the number of ranking errors($\mathcal{C}$-index) on the training data $\mathcal{D}$.

$$C(U,O) = \frac{\sum\limits_{1\leq i<j\leq k} \ \sum\limits_{(o,o\prime)\in O_i\times O_j} \mathcal{S}(\mathcal{U}(o),\, \mathcal{U}(o\prime))}{\sum\limits_{i<j} |O_i|.|O_j|},\ \mathcal{S}(u,v) = \begin{cases} 1 & u<v \\ 0 & \text{otherwise} \end{cases}$$