# Preference Learning: A Tutorial Introduction

**Johannes Fürnkranz**
Knowledge Engineering
Dept. of Computer Science
Technical University Darmstadt, Germany

**Eyke Hüllermeier**
Knowledge Engineering & Bioinformatics Lab
Dept. of Mathematics and Computer Science
Marburg University, Germany

# What is Preference Learning ?

- Preference learning is an emerging **subfield of machine learning**
- Roughly speaking, it deals with the **learning of (predictive) preference models** from observed (or extracted) preference information

MACHINE LEARNING ⟷ **Preference Learning** ⟷ PREFERENCE MODELING and DECISION ANALYSIS

computer science
artificial intelligence

operations research
social sciences (voting and choice theory)
economics and decision theory

## Workshops and Related Events

- NIPS–01: New Methods for Preference Elicitation
- NIPS–02: Beyond Classification and Regression: Learning Rankings, Preferences, Equality Predicates, and Other Structures
- KI–03: Preference Learning: Models, Methods, Applications
- NIPS–04: Learning With Structured Outputs
- NIPS–05: Workshop on Learning to Rank
- IJCAI–05: Advances in Preference Handling
- SIGIR 07–10: Workshop on Learning to Rank for Information Retrieval
- **ECML/PDKK 08–10: Workshop on Preference Learning**
- NIPS–09: Workshop on Advances in Ranking
- American Institute of Mathematics Workshop in Summer 2010: The Mathematics of Ranking
- NIPS-11: Workshop on Choice Models and Preference Learning

# Preferences in Artificial Intelligence

More generally, „preferences“ is a key topic in current AI research

**User preferences** play a key role in various fields of application:

- recommender systems,
- adaptive user interfaces,
- adaptive retrieval systems,
- autonomous agents (electronic commerce),
- games, …

Preferences in **AI research**:

- **preference representation** (CP nets, GAU networks, logical representations, fuzzy constraints, …)
- **reasoning** with preferences (decision theory, constraint satisfaction, non-monotonic reasoning, …)
- **preference acquisition** (preference elicitation, **preference learning**, ...)

## AGENDA

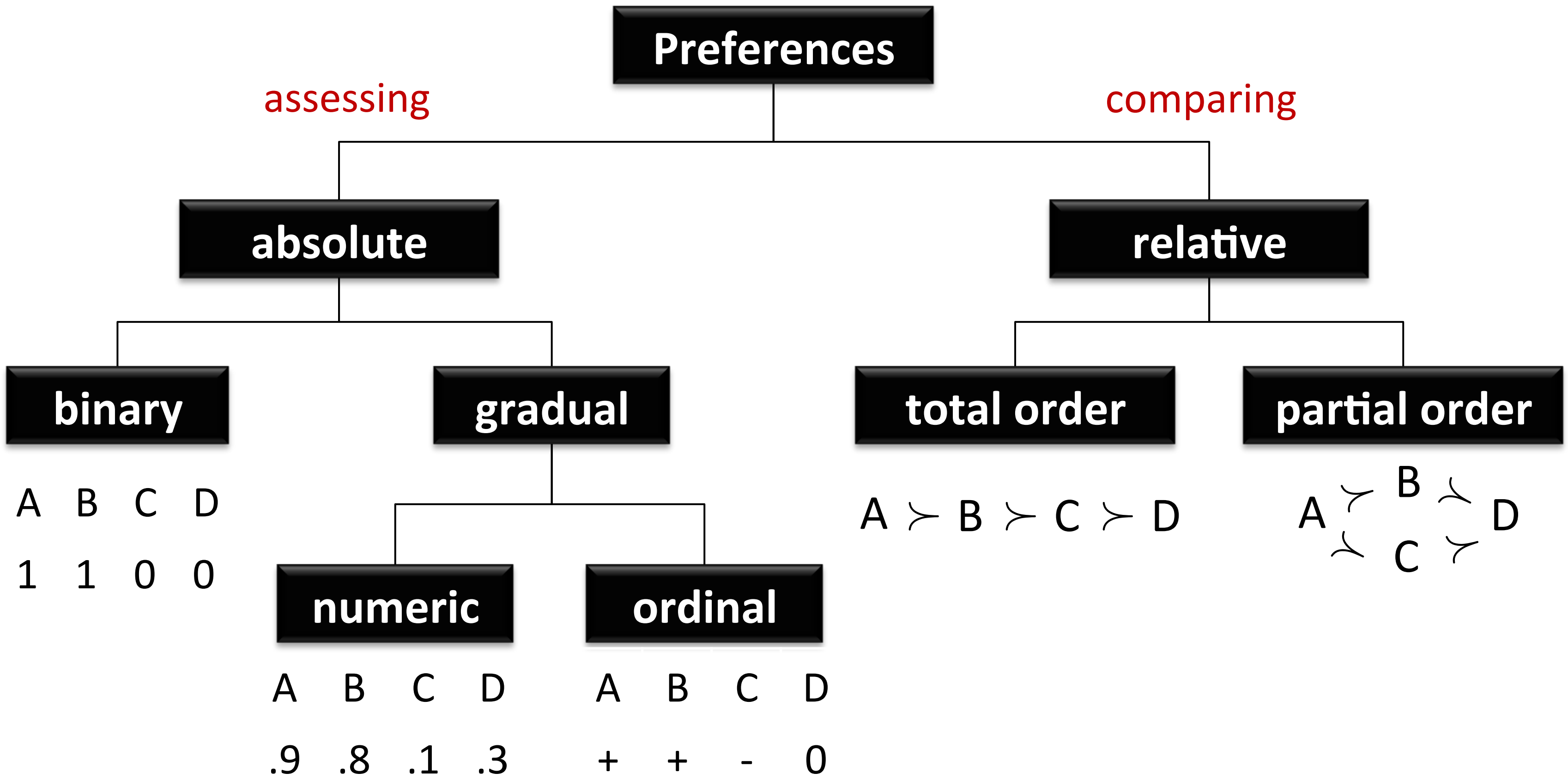
1. Preference Learning Tasks
2. Performance Assessment and Loss Functions
3. Preference Learning Techniques
4. Conclusions

## Preference Learning

Preference learning problems can be distinguished along several **problem dimensions**, including

- **representation of preferences, type of preference model**:
  - utility function (ordinal, numeric),
  - preference relation (partial order, ranking, …),
  - logical representation, …
- **description of individuals/users and alternatives/items**:
  - identifier, feature vector, structured object, …
- **type of training input**:
  - direct or indirect feedback,
  - complete or incomplete relations,
  - utilities, …
- …

# Preference Learning




→ (ordinal) regression

→ classification/ranking

## Structure of this Overview

(1) Preference learning as an extension of **conventional supervised learning**: Learn a mapping

$$\mathcal{X} \rightarrow \mathfrak{P}$$

that maps instances to preference models (→ structured/complex output prediction).

(2) Other settings (object ranking, instance ranking, CF, …)

## Structure of this Overview

(1) Preference learning as an extension of **conventional supervised learning**: Learn a mapping

$$\mathcal{X} \rightarrow \mathfrak{P}$$

that maps instances to preference models (→ structured/complex output prediction).

Instances are typically (though not necessarily) characterized in terms of a feature vector.

The output space consists of preference models over a fixed set of alternatives (classes, labels, …) represented in terms of an identifier
→ *extensions of multi-class classification*

# **Multilabel Classification** [Tsoumakas & Katakis 2007]

## Training

| X1 | X2 | X3 | X4 | A | B | C | D |
|---|---|---|---|---|---|---|---|
| 0.34 | 0 | 10 | 174 | 0 | **1** | **1** | 0 |
| 1.45 | 0 | 32 | 277 | 0 | **1** | 0 | **1** |
| 1.22 | 1 | 46 | 421 | 0 | 0 | 0 | 1 |
| 0.74 | 1 | 25 | 165 | 0 | **1** | **1** | **1** |
| 0.95 | 1 | 72 | 273 | **1** | 0 | **1** | 0 |
| 1.04 | 0 | 33 | 158 | **1** | **1** | **1** | 0 |

Binary preferences on a fixed set of items: liked or disliked

## Prediction

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0.92 | 1 | 81 | 382 | **0** | **1** | **0** | **1** |

LOSS

## Ground truth

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0.92 | 1 | 81 | 382 | **1** | **1** | **0** | **1** |

# Multilabel Ranking

### Training

| X1 | X2 | X3 | X4 | A | B | C | D |
|---|---|---|---|---|---|---|---|
| 0.34 | 0 | 10 | 174 | 0 | **1** | **1** | 0 |
| 1.45 | 0 | 32 | 277 | 0 | **1** | 0 | **1** |
| 1.22 | 1 | 46 | 421 | 0 | 0 | 0 | 1 |
| 0.74 | 1 | 25 | 165 | 0 | **1** | **1** | **1** |
| 0.95 | 1 | 72 | 273 | **1** | 0 | **1** | 0 |
| 1.04 | 0 | 33 | 158 | **1** | **1** | **1** | 0 |

Binary preferences on a fixed set of items: liked or disliked

### Prediction

$B \succ D \succ C \succ A$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **0.92** | **1** | **81** | **382** | **4** | **1** | **3** | **2** |

A ranking of all items

LOSS

### Ground truth

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **0.92** | **1** | **81** | **382** | **1** | **1** | **0** | **1** |

# Graded Multilabel Classification [Cheng et al. 2010]

### Training

| X1 | X2 | X3 | X4 | A | B | C | D |
|---|---|---|---|---|---|---|---|
| 0.34 | 0 | 10 | 174 | -- | + | ++ | 0 |
| 1.45 | 0 | 32 | 277 | 0 | ++ | -- | + |
| 1.22 | 1 | 46 | 421 | -- | -- | 0 | + |
| 0.74 | 1 | 25 | 165 | 0 | + | + | ++ |
| 0.95 | 1 | 72 | 273 | + | 0 | ++ | -- |
| 1.04 | 0 | 33 | 158 | + | + | ++ | -- |

Ordinal preferences on a fixed set of items: liked, disliked, or something in-between

### Prediction

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **0.92** | **1** | **81** | **382** | **--** | **+** | **0** | **++** |

LOSS

### Ground truth

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **0.92** | **1** | **81** | **382** | **0** | **++** | **--** | **+** |

# Graded Multilabel Ranking

### Training

| X1 | X2 | X3 | X4 | A | B | C | D |
|---|---|---|---|---|---|---|---|
| 0.34 | 0 | 10 | 174 | -- | + | ++ | 0 |
| 1.45 | 0 | 32 | 277 | 0 | ++ | -- | + |
| 1.22 | 1 | 46 | 421 | -- | -- | 0 | + |
| 0.74 | 1 | 25 | 165 | 0 | + | + | ++ |
| 0.95 | 1 | 72 | 273 | + | 0 | ++ | -- |
| 1.04 | 0 | 33 | 158 | + | + | ++ | -- |

Ordinal preferences on a fixed set of items: liked, disliked, or something in-between

### Prediction

$B \succ D \succ C \succ A$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **0.92** | **1** | **81** | **382** | **4** | **1** | **3** | **2** |

A ranking of all items

LOSS

### Ground truth

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **0.92** | **1** | **81** | **382** | **0** | **++** | **--** | **+** |

# Label Ranking [Hüllermeier et al. 2008]

## Training

| X1 | X2 | X3 | X4 | Preferences |
|---|---|---|---|---|
| 0.34 | 0 | 10 | 174 | $A \succ B$, $B \succ C$, $C \succ D$ |
| 1.45 | 0 | 32 | 277 | $B \succ C$ |
| 1.22 | 1 | 46 | 421 | $B \succ D$, $A \succ D$, $C \succ D$, $A \succ C$ |
| 0.74 | 1 | 25 | 165 | $C \succ A$, $C \succ D$, $A \succ B$ |
| 0.95 | 1 | 72 | 273 | $B \succ D$, $A \succ D$ |
| 1.04 | 0 | 33 | 158 | $D \succ A$, $A \succ B$, $C \succ B$, $A \succ C$ |

Instances are associated with pairwise preferences between labels.

## Prediction

$B \succ D \succ C \succ A$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **0.92** | **1** | **81** | **382** | **4** | **1** | **3** | **2** |

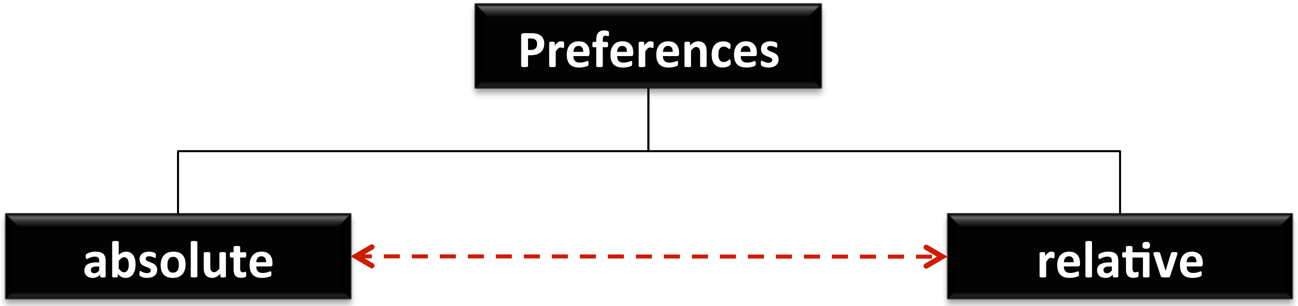
A ranking of all labels

LOSS

## Ground truth

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **0.92** | **1** | **81** | **382** | **2** | **1** | **3** | **4** |

# Calibrated Label Ranking [Fürnkranz et al. 2008]




Combining absolute and relative evaluation:

$$\underbrace{a \succ b \succ c}_{\substack{\text{relevant}\\ \text{positive}\\ \text{liked}}} \;\mid\; \underbrace{d \succ e \succ f \succ g}_{\substack{\text{irrelevant}\\ \text{negative}\\ \text{disliked}}}$$

## Structure of this Overview

(1) Preference Learning as an extension of conventional supervised learning: Learn a mapping

$$\mathcal{X} \rightarrow \mathfrak{P}$$

that maps instances to preference models (→ structured output prediction).

**(2) Other settings** (no clear distinction between input/output space)

object ranking, instance ranking, collaborative filtering

## Object Ranking [Cohen et al. 99]

### Training

$$\begin{array}{lcl} (0.74, 1, 25, 165) & \succ & (0.45, 0, 35, 155) \\ (0.47, 1, 46, 183) & \succ & (0.57, 1, 61, 177) \\ (0.25, 0, 26, 199) & \succ & (0.73, 0, 46, 185) \\ (0.95, 0, 73, 133) & \succ & (0.25, 1, 35, 153) \\ (0.68, 1, 55, 147) & \succ & (0.67, 0, 63, 182) \end{array}$$

Pairwise preferences between objects (instances).

### Prediction (ranking a new set of objects)

$$\mathcal{Q} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7, \boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}, \boldsymbol{x}_{11}, \boldsymbol{x}_{12}, \boldsymbol{x}_{13}\}$$

$$\boldsymbol{x}_{10} \succ \boldsymbol{x}_4 \succ \boldsymbol{x}_7 \succ \boldsymbol{x}_1 \succ \boldsymbol{x}_{11} \succ \boldsymbol{x}_2 \succ \boldsymbol{x}_8 \succ \boldsymbol{x}_{13} \succ \boldsymbol{x}_9 \succ \boldsymbol{x}_3 \succ \boldsymbol{x}_{12} \succ \boldsymbol{x}_5 \succ \boldsymbol{x}_6$$

### Ground truth (ranking or top-ranking or subset of relevant objects)

$$\boldsymbol{x}_{11} \succ \boldsymbol{x}_7 \succ \boldsymbol{x}_4 \succ \boldsymbol{x}_2 \succ \boldsymbol{x}_{10} \succ \boldsymbol{x}_1 \succ \boldsymbol{x}_8 \succ \boldsymbol{x}_{13} \succ \boldsymbol{x}_9 \succ \boldsymbol{x}_{12} \succ \boldsymbol{x}_3 \succ \boldsymbol{x}_5 \succ \boldsymbol{x}_6$$

$$\boldsymbol{x}_{11} \succ \boldsymbol{x}_7 \succ \boldsymbol{x}_4 \succ \boldsymbol{x}_2 \succ \boldsymbol{x}_{10}$$

$$\mathcal{P} = \{\boldsymbol{x}_{11}, \boldsymbol{x}_7, \boldsymbol{x}_4, \boldsymbol{x}_2, \boldsymbol{x}_{10}, \boldsymbol{x}_1\} \qquad \mathcal{N} = \{\boldsymbol{x}_8, \boldsymbol{x}_{13}, \boldsymbol{x}_9, \boldsymbol{x}_{12}, \boldsymbol{x}_3, \boldsymbol{x}_5, \boldsymbol{x}_6\}$$

# Instance Ranking [Fürnkranz et al. 2009]

## Training

| | X1 | X2 | X3 | X4 | class |
|---|---|---|---|---|---|
| $\boldsymbol{x}_1$ | 0.34 | 0 | 10 | 174 | -- |
| $\boldsymbol{x}_2$ | 1.45 | 0 | 32 | 277 | 0 |
| $\boldsymbol{x}_3$ | 0.74 | 1 | 25 | 165 | ++ |
| | … | … | … | … | … |
| $\boldsymbol{x}_n$ | 0.95 | 1 | 72 | 273 | + |

Absolute preferences on an ordinal scale.

## Prediction (ranking a new set of objects)

$$\mathcal{Q} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7, \boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10}, \boldsymbol{x}_{11}, \boldsymbol{x}_{12}, \boldsymbol{x}_{13}\}$$

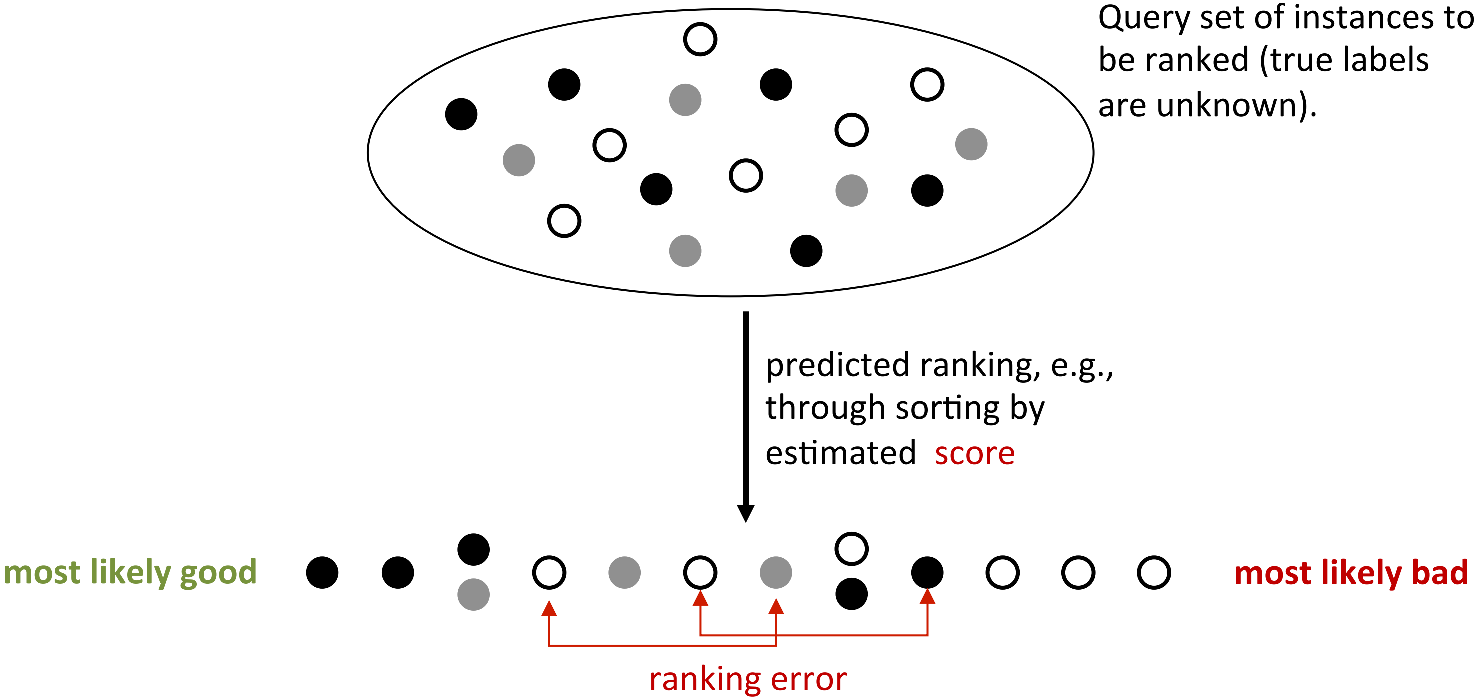
$$\boldsymbol{x}_{10} \succ \boldsymbol{x}_4 \succ \boldsymbol{x}_7 \succ \boldsymbol{x}_1 \succ \boldsymbol{x}_{11} \succ \boldsymbol{x}_2 \succ \boldsymbol{x}_8 \succ \boldsymbol{x}_{13} \succ \boldsymbol{x}_9 \succ \boldsymbol{x}_3 \succ \boldsymbol{x}_{12} \succ \boldsymbol{x}_5 \succ \boldsymbol{x}_6$$

## Ground truth (ordinal classes)

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\boldsymbol{x}_{10}$ | $\boldsymbol{x}_4$ | $\boldsymbol{x}_7$ | $\boldsymbol{x}_1$ | $\boldsymbol{x}_{11}$ | $\boldsymbol{x}_2$ | $\boldsymbol{x}_8$ | $\boldsymbol{x}_{13}$ | $\boldsymbol{x}_9$ | $\boldsymbol{x}_3$ | $\boldsymbol{x}_{12}$ | $\boldsymbol{x}_5$ | $\boldsymbol{x}_6$ |
| + | 0 | ++ | ++ | -- | + | 0 | + | -- | 0 | 0 | -- | -- |

# **Instance Ranking** [Fürnkranz et al. 2009]

Extension of AUC maximization to the polytomous case, in which instances are rated on an ordinal scale such as { **`bad, medium, good}`**



## Collaborative Filtering [Goldberg et al. 1992]

PRODUCTS

USERS

| | P1 | P2 | P3 | … | P38 | … | P88 | P89 | P90 |
|---|---|---|---|---|---|---|---|---|---|
| **U1** | 1 | | 4 | … | | … | | 3 | |
| **U2** | | 2 | 2 | … | | … | 1 | | |
| **…** | | | | … | | … | | | |
| **U46** | **?** | **2** | **?** | **…** | **?** | **…** | **?** | **?** | **4** |
| **…** | | | | … | | … | | | |
| **U98** | 5 | | | … | | … | 4 | | |
| **U99** | | | 1 | … | | … | | 2 | |

1: very bad, 2: bad, 3: fair, 4: good, 5: excellent

Inputs and outputs as identifiers, absolute preferences in terms of ordinal degrees.

# Dyadic Prediction [Menon & Elkan 2010]

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| | | | | | 10 | 14 | 45 | 32 | 52 | 61 | 16 | 33 | 53 |
| | | | | | **P1** | **P2** | **P3** | **…** | **P38** | **…** | **P88** | **P89** | **P90** |
| ? | ? | 1 | 5 | **U1** | 1 | | 4 | … | | … | | 3 | |
| ? | ? | 0 | 4 | **U2** | | 2 | 2 | … | | … | 1 | | |
| ? | ? | 0 | 6 | **…** | | | | … | | … | | | |
| ? | ? | 1 | 5 | **U46** | **?** | **2** | **?** | **…** | **?** | **…** | **?** | **?** | **4** |
| ? | ? | 1 | 7 | **…** | | | | … | | … | | | |
| ? | ? | 0 | 6 | **U98** | 5 | | | … | | … | 4 | | |
| ? | ? | 1 | 6 | **U99** | | | 1 | … | | … | | 2 | |

Additional side-information: observed features + latent features of users and items

# Preference Learning Tasks

| | representation | | type of preference information | | |
|---|---|---|---|---|---|
| **task** | **input** | **output** | **training** | **prediction** | **ground truth** |
| collaborative filtering | identifier | **identifier** | absolute ordinal | absolute ordinal | absolute ordinal |
| multilabel classification | feature | **identifier** | absolute binary | absolute binary | absolute binary |
| multilabel ranking | feature | **identifier** | absolute binary | ranking | absolute binary |
| graded multilabel classification | feature | **identifier** | absolute ordinal | absolute ordinal | absolute ordinal |
| label ranking | feature | **identifier** | relative binary | ranking | ranking |
| object ranking | **feature** | -- | relative binary | ranking | ranking or subset |
| instance ranking | **feature** | identifier | absolute ordinal | ranking | absolute ordinal |

generalized classification

ranking

Two main directions: (1) ranking and variants (2) generalizations of classification.

# Loss Functions

**Things to be compared:**

| prediction | | ground truth |
|---|---|---|
| absolute utility degree | ⟷ | absolute utility degree |
| subset of preferred items | ⟷ | subset of preferred items |
| fuzzy subset of preferred items | ⟷ | fuzzy subset of preferred items |
| subset of preferred items | ⟷ | ranking of items |
| ranking of items | ⟷ | ordered partition of items |
| ranking of items | ⟷ | ranking of items |

standard comparison of scalar predictions

non-standard comparisons

**prediction** **ground truth**

# References


- W. Cheng, K. Dembczynski and E. Hüllermeier. *Graded Multilabel Classification: The Ordinal Case*. ICML-2010, Haifa, Israel, 2010.
- W. Cheng and E. Hüllermeier. *Predicting partial orders: Ranking with abstention*. ECML/PKDD-2010, Barcelona, 2010.
- Y. Chevaleyre, F. Koriche, J. Lang, J. Mengin, B. Zanuttini. *Learning ordinal preferences on multiattribute domains: The case of CP-nets*. In: J. Fürnkranz and E. Hüllermeier (eds.) Preference Learning, Springer-Verlag, 2010.
- W.W. Cohen, R.E. Schapire and Y. Singer. *Learning to order things*. Journal of Artificial Intelligence Research, 10:243–270, 1999.
- J. Fürnkranz, E. Hüllermeier, E. Mencia, and K. Brinker. *Multilabel Classification via Calibrated Label Ranking*. Machine Learning 73(2):133-153, 2008.
- J. Fürnkranz, E. Hüllermeier and S. Vanderlooy. *Binary decomposition methods for multipartite ranking*. Proc. ECML-2009, Bled, Slovenia, 2009.
- D. Goldberg, D. Nichols, B.M. Oki and D. Terry. *Using collaborative filtering to weave and information tapestry*. Communications of the ACM, 35(12):61–70, 1992.
- E. Hüllermeier, J. Fürnkranz, W. Cheng and K. Brinker. *Label ranking by learning pairwise preferences*. Artificial Intelligence, 172:1897–1916, 2008.
- G. Tsoumakas and I. Katakis. *Multi-label classification: An overview*. Int. J. Data Warehouse and Mining, 3:1–13, 2007.
- A.K. Menon and C. Elkan. *Predicting labels for dyadic data*. Data Mining and Knowledge Discovery, 21(2), 2010

# AGENDA

1. Preference Learning Tasks
2. **Loss Functions**
   a. **Evaluation of Rankings**
   b. Weighted Measures
   c. Evaluation of Bipartite Rankings
   d. Evaluation of Partial Rankings
3. Preference Learning Techniques
4. Conclusions

# Rank Evaluation Measures

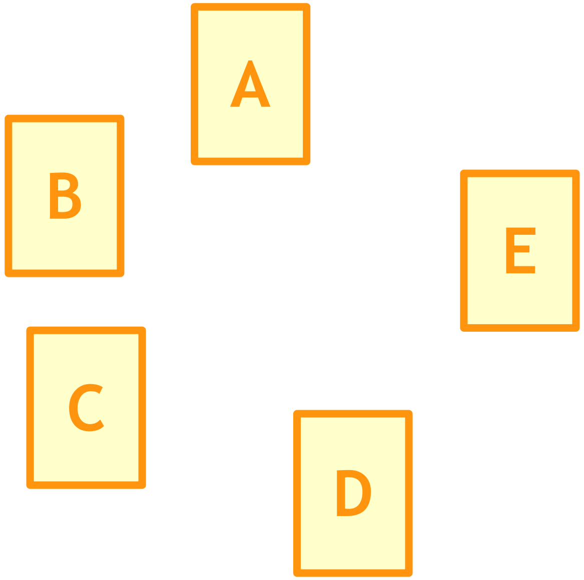
- In the following, we do not discriminate between different ranking scenarios
  - we use the term items for both, objects and labels
- All measures are applicable to both scenarii
  - sometimes have different names according to context
- Label Ranking
  - measure is applied to the ranking of the labels of each examples
  - averaged over all examples
- Object Ranking
  - measure is applied to the ranking of a set of objects
  - we may need to average over different sets of objects which have disjoint preference graphs
    - e.g. different sets of query / answer set pairs in information retrieval

# Ranking Errors

- Given:
  - a set of items $X = \{x_1, \ldots, x_c\}$ to rank
    - Example:

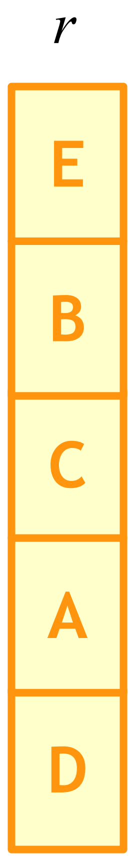
      $X = \{\text{A, B, C, D, E}\}$

items can be objects or labels



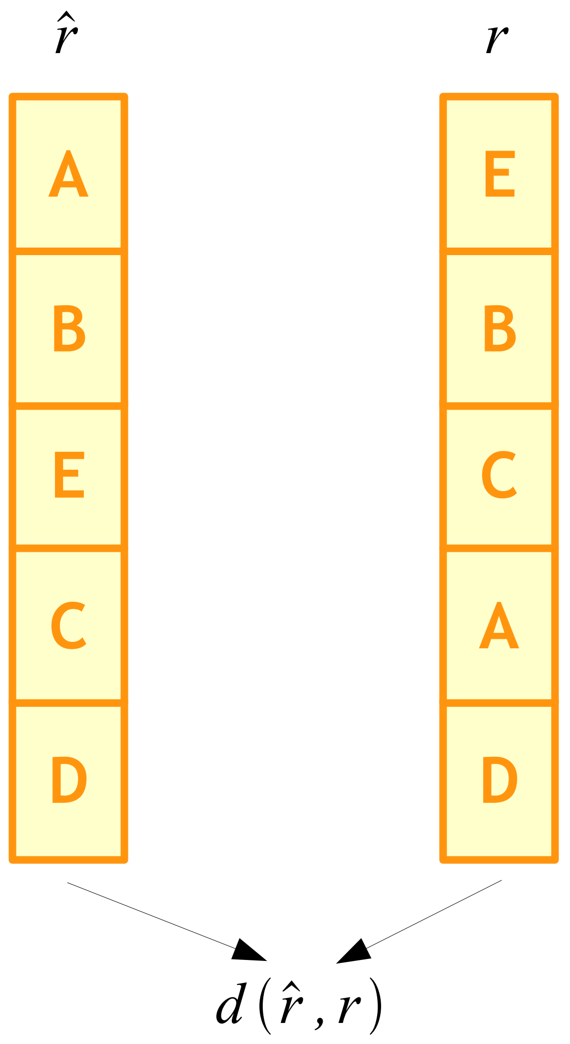
# Ranking Errors

- Given:
  - a set of items $X = \{x_1, \ldots, x_c\}$ to rank
    - *Example:*
      $X$ = {A, B, C, D, E}
  - a target ranking $r$
    - *Example*:
      E ≻ B ≻ C ≻ A ≻ D



# Ranking Errors

- Given:
  - a set of items $X = \{x_1, \ldots, x_c\}$ to rank
    - *Example:*
      $X$ = {A, B, C, D, E}
  - a target ranking $r$
    - *Example*:
      E $\succ$ B $\succ$ C $\succ$ A $\succ$ D
  - a predicted ranking $\hat{r}$
    - *Example*:
      A $\succ$ B $\succ$ E $\succ$ C $\succ$ D
- Compute:
  - a value $d(r, \hat{r})$ that measures the *distance* between the two rankings



# Notation

- $r$ and $\hat{r}$ are functions from $X \rightarrow \mathbb{N}$
  - returning the rank of an item x

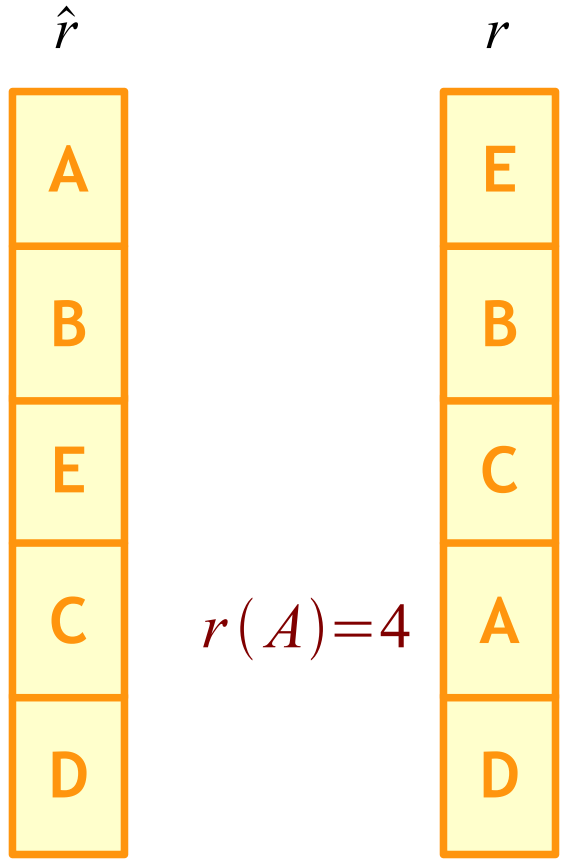
$$\hat{r}(A)=1$$

- the inverse functions $r^{-1}$: $\mathbb{N} \rightarrow X$
  - return the item at a certain position

$$\hat{r}^{-1}(1)=A \qquad r^{-1}(4)=A$$

- as a short-hand for $r \circ \hat{r}^{-1}$, we also define function $R$: $\mathbb{N} \rightarrow \mathbb{N}$
  - $R(i)$ returns the true rank of the $i$-th item in the predicted ranking

$$R(1)=r(\hat{r}^{-1}(1))=4$$



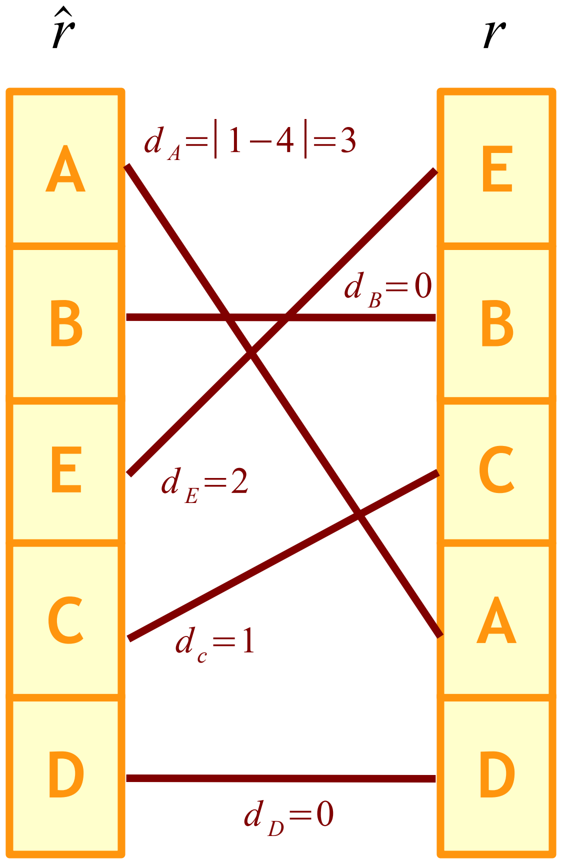

$$r(A)=4$$

# Spearman's Footrule

- Key idea:
  - Measure the sum of absolute differences between ranks

$$D_{SF}(r,\hat{r}) = \sum_{i=1}^{c} |r(x_i) - \hat{r}(x_i)| = \sum_{i=1}^{c} |i - R(i)|$$

$$= \sum_{i=1}^{c} d_{x_i}(r,\hat{r})$$



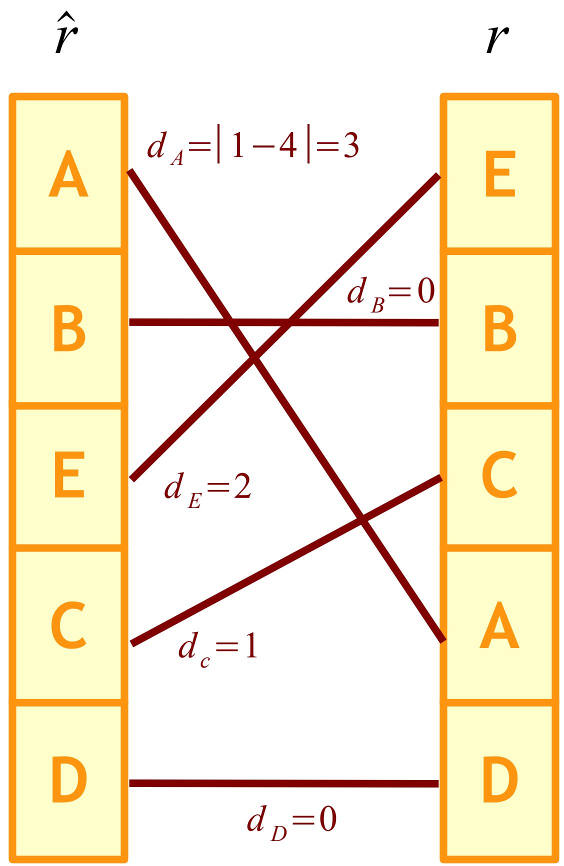

$$\sum_{x_i} d_{x_i} = 3+0+1+0+2 = 6$$

# Spearman Distance

- Key idea:
  - Measure the sum of ~~absolute~~ squared differences between ranks

$$D_S(r,\hat{r})=\sum_{i=1}^{c}\left(r(x_i)-\hat{r}(x_i)\right)^2=\sum_{i=1}^{c}\left(i-R(i)\right)^2$$

$$=\sum_{i=1}^{c}d_{x_i}(r,\hat{r})^2$$




$$\sum_{x_i} d_{x_i}^2=3^2+0+1^2+0+2^2=14$$

- Value range:

$$\min D_S(r,\hat{r})=0$$

$$\max D_S(r,\hat{r})=\sum_{i=1}^{c}((c-i)-i)^2=\frac{c\cdot(c^2-1)}{3}$$

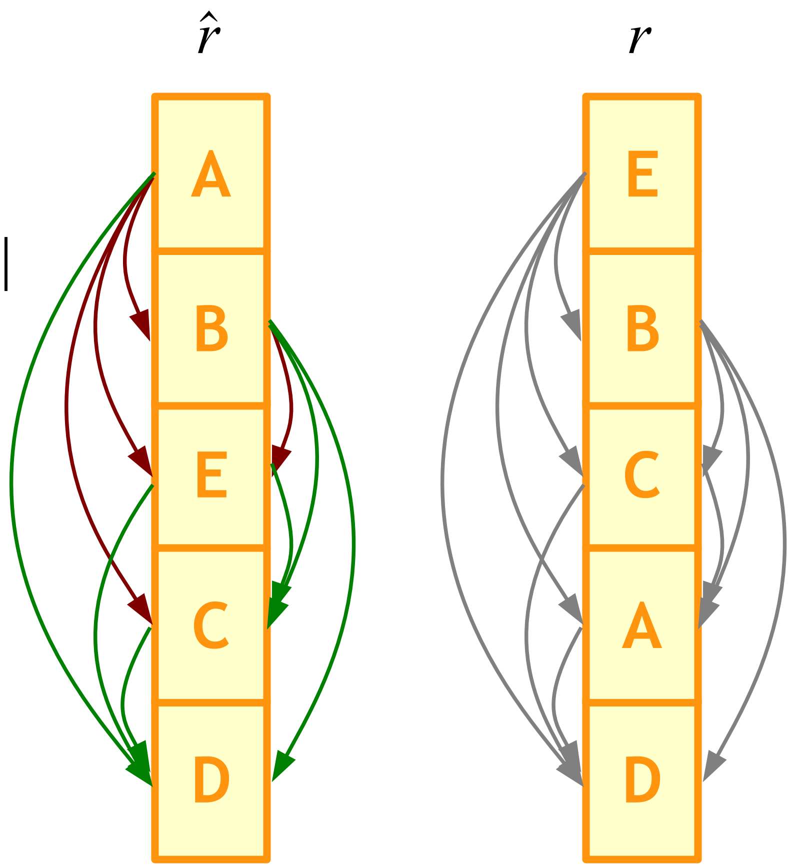
→ Spearman Rank Correlation Coefficient

$$1-\frac{6\cdot D_S(r,\hat{r})}{c\cdot(c^2-1)}\in[-1,+1]$$

# Kendall's Distance

- Key idea:
  - number of item pairs that are inverted in the predicted ranking

$$D_\tau(r, \hat{r}) = \left| \left\{ (i,j) \mid r(x_i) < r(x_j) \wedge \hat{r}(x_i) > \hat{r}(x_j) \right\} \right|$$




$D_\tau(r, \hat{r}) = 4$

- Value range:

$$\min D_\tau(r, \hat{r}) = 0$$

$$\max D_\tau(r, \hat{r}) = \frac{c \cdot (c-1)}{2}$$

$\rightarrow$ Kendall's tau

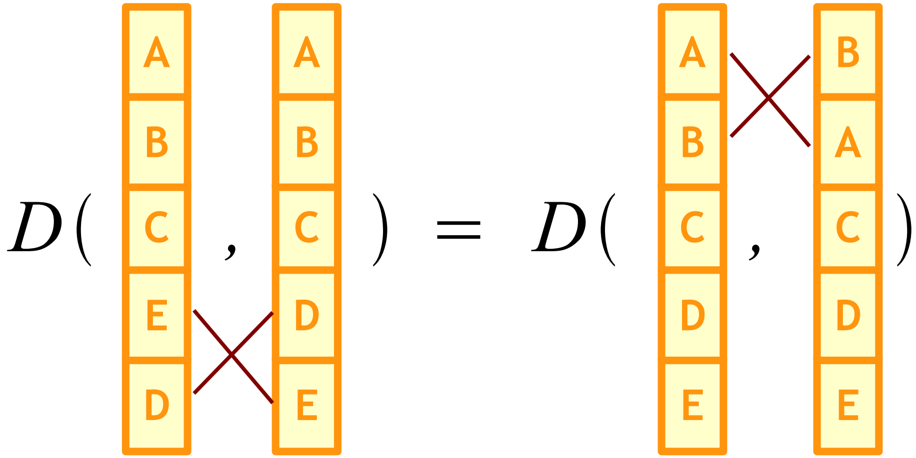
$$1 - \frac{4 \cdot D_\tau(r, \hat{r})}{c \cdot (c-1)} \in [-1, +1]$$

# AGENDA

1. Preference Learning Tasks
2. **Loss Functions**
   a. Evaluation of Rankings
   **b. Weighted Measures**
   c. Evaluation of Bipartite Rankings
   d. Evaluation of Partial Rankings
3. Preference Learning Techniques
4. Conclusions

# Weighted Ranking Errors

- The previous ranking functions give equal weight to all ranking positions
  - i.e., differences in the first ranking positions have the same effect as differences in the last ranking positions



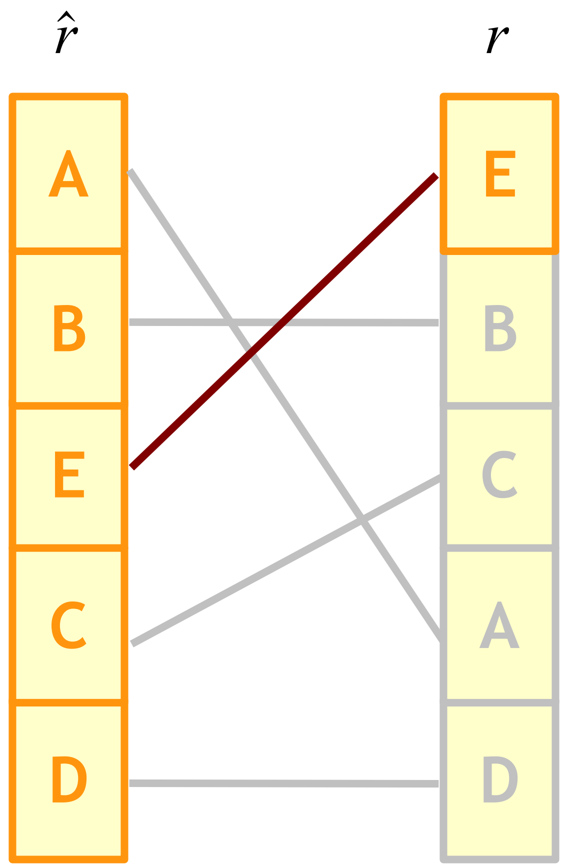

- In many applications this is not desirable
  - ranking of search results
  - ranking of product recommendations
  - ranking of labels for classification
  - ...

$\Rightarrow$ Higher ranking positions should be given more weight

# Position Error

- Key idea:
  - in many applications we are interested in providing a ranking where the target item appears a high as possible in the predicted ranking
    - e.g. ranking a set of actions for the next step in a plan
  - Error is the number of wrong items that are predicted before the target item

$$D_{PE}(r,\hat{r}) = \hat{r}(\arg\min_{x \in X} r(x)) - 1$$




$$D_{PE}(r,\hat{r}) = 2$$

- Note:
  - equivalent to Spearman's footrule with all non-target weights set to 0

$$D_{PE}(r,\hat{r}) = \sum_{i=1}^{c} w_i \cdot d_{x_i}(r,\hat{r})$$

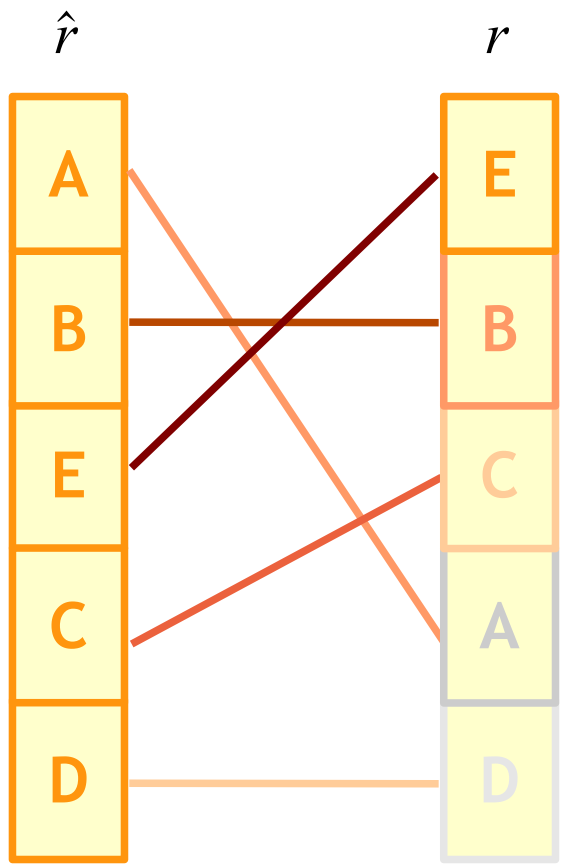
with $w_i = [\![ x_i = \arg\min_{x \in X} r(x) ]\!]$

# Discounted Error

- Higher ranks in the target position get a higher weight than lower ranks

$$D_{DR}(r,\hat{r}) = \sum_{i=1}^{c} w_i \cdot d_{x_i}(r,\hat{r})$$

with $w_i = \dfrac{1}{\log(r(x_i)+1)}$




$$D_{DR}(r,\hat{r}) = \frac{3}{\log 2} + 0 + \frac{1}{\log 4} + 0 + \frac{2}{\log 6}$$

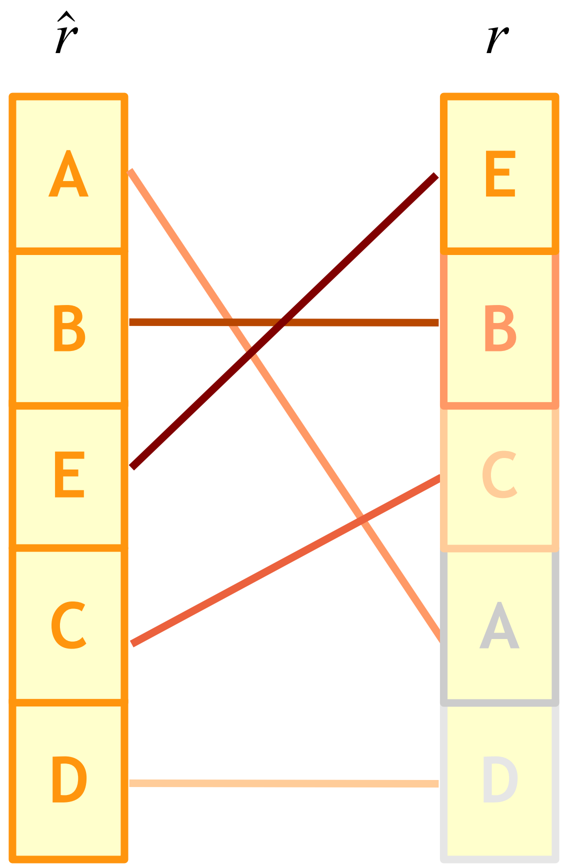
# (Normalized) Discounted Cumulative Gain

- a "positive" version of discounted error:
  Discounted Cumulative Gain (DCG)

$$DCG(r, \hat{r}) = \sum_{i=1}^{c} \frac{c - R(i)}{\log(i+1)}$$

- Maximum possible value:
  - the predicted ranking is correct, i.e. $\forall\, i : i = R(i)$
  - Ideal Discounted Cumulative Gain (IDCG)

$$IDCG = \sum_{i=1}^{c} \frac{c - i}{\log(i+1)}$$




- Normalized DCG (NDCG)

$$NDCG(r, \hat{r}) = \frac{DCG(r, \hat{r})}{IDCG}$$

$$NDCG(r, \hat{r}) = \frac{\frac{1}{\log 2} + \frac{3}{\log 3} + \frac{4}{\log 4} + \frac{2}{\log 5} + \frac{0}{\log 6}}{\frac{4}{\log 2} + \frac{3}{\log 3} + \frac{2}{\log 4} + \frac{1}{\log 5} + \frac{0}{\log 6}}$$

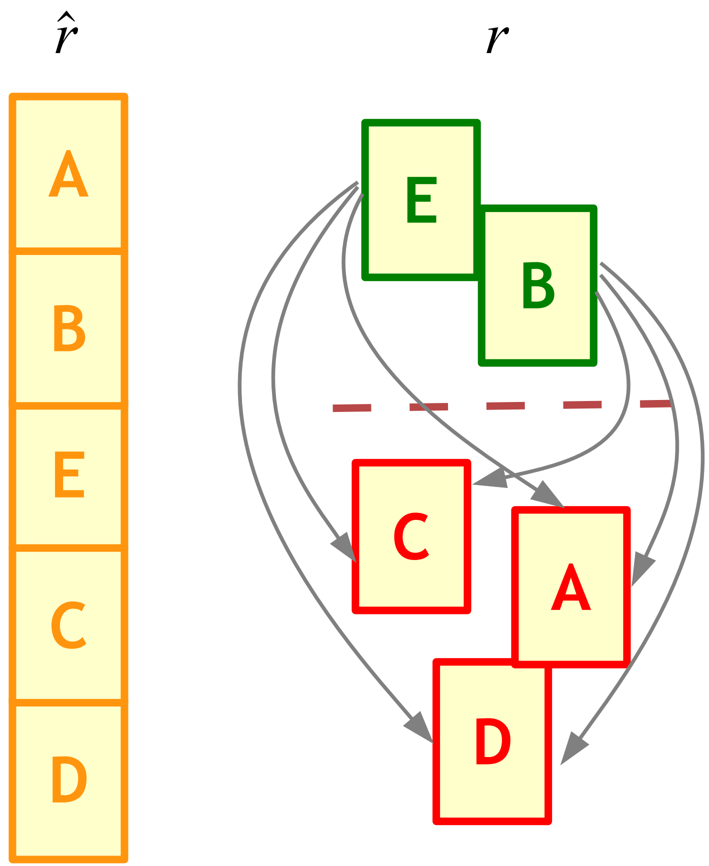
# AGENDA

1. Preference Learning Tasks
2. **Loss Functions**
    a. Evaluation of Rankings
    b. Weighted Measures
    **c. Evaluation of Bipartite Rankings**
    d. Evaluation of Partial Rankings
3. Preference Learning Techniques
4. Conclusions

# Bipartite Rankings

## Bipartite Rankings

- The target ranking is not totally ordered but a *bipartite graph*
- The two partitions may be viewed as preference levels $L = \{0, 1\}$
  - all $c_1$ items of level 1 are preferred over all $c_0$ items of level 0
- We now have fewer preferences
  - for a total order: $\frac{c}{2} \cdot (c-1)$
  - for a bipartite graph: $c_1 \cdot (c - c_1)$

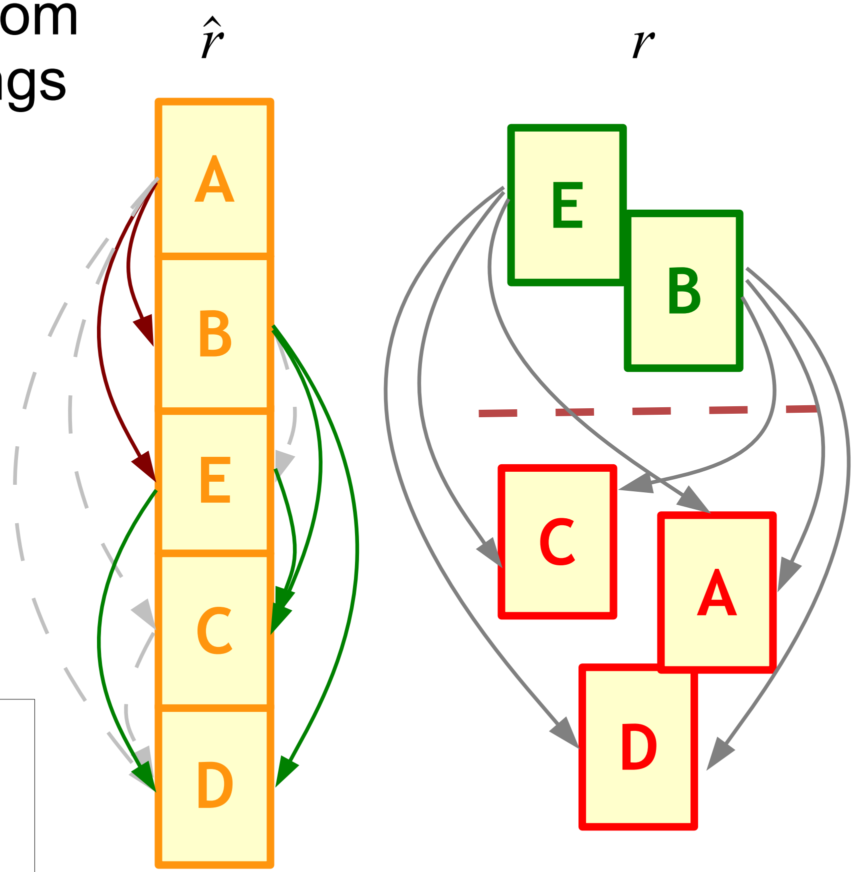


# Evaluating Partial Target Rankings

- Many Measures can be directly adapted from total target rankings to partial target rankings
- Recall: Kendall's distance
  - number of item pairs that are inverted in the target ranking

$$D_\tau(r,\hat{r}) = \left| \left\{ (i,j) \mid r(x_i) < r(x_j) \wedge \hat{r}(x_i) > \hat{r}(x_j) \right\} \right|$$

  - can be directly used
  - in case of normalization, we have to consider that fewer items satisfy $r(x_i) < r(x_j)$
- Area under the ROC curve (AUC)
  - the AUC is the fraction of pairs of $(p,n)$ for which the predicted score $s(p) > s(n)$
    - Mann Whitney statistic is the absolute number
  - This is $1$ - normalized Kendall's distance for a bipartite preference graph with $L = \{p,n\}$



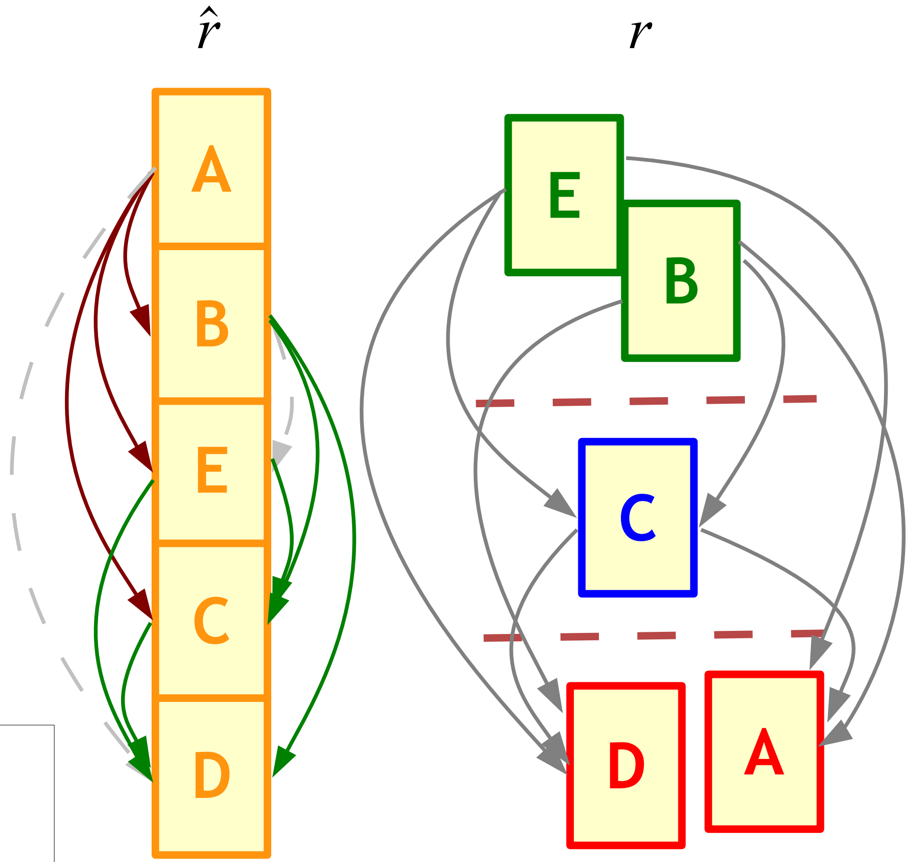

$$D_\tau(r,\hat{r}) = 2$$

$$AUC(r,\hat{r}) = \frac{4}{6}$$

# Evaluating Multipartite Rankings

- Multipartite rankings:
  - like Bipartite rankings
  - but the target ranking $r$ consists of *multiple* relevance levels $L = \{1 \ldots l\}$, where $l < c$
  - total ranking is a special case where each level has exactly one item
- # of preferences $= \sum_{(i,j)} c_i \cdot c_j \leq \frac{c^2}{2} \cdot (1 - \frac{1}{l})$
  - $c_i$ is the number of items in level $I$
- C-Index [Gnen & Heller, 2005]
  - straight-forward generalization of AUC
  - fraction of pairs $(x_i, x_j)$ for which

$$l(i) > l(j) \wedge \hat{r}(x_i) < \hat{r}(x_j)$$




$$D_\tau(r, \hat{r}) = 3$$

$$\text{C-Index}(r, \hat{r}) = \frac{5}{8}$$

# Evaluating Multipartite Rankings

## C-Index

- the C-index can be rewritten as a weighted sum of pairwise AUCs:

$$\text{C-Index}(r, \hat{r}) = \frac{1}{\sum_{i,j>i} c_i \cdot c_j} \sum_{i,j<i} c_i \cdot c_j \cdot \text{AUC}(r_{i,j}, \hat{r}_{i,j})$$

where $r_{i,j}$ and $\hat{r}_{i,j}$ are the rankings $r$ and $\hat{r}$ restricted to levels $i$ and $j$.

## Jonckheere-Terpstra statistic

- is an *unweighted* sum of pairwise AUCs:

$$\text{m-AUC} = \frac{2}{l \cdot (l-1)} \sum_{i,j>i} \text{AUC}(r_{i,j}, \hat{r}_{i,j})$$

- equivalent to well-known multi-class extension of AUC
  [Hand & Till, MLJ 2001]

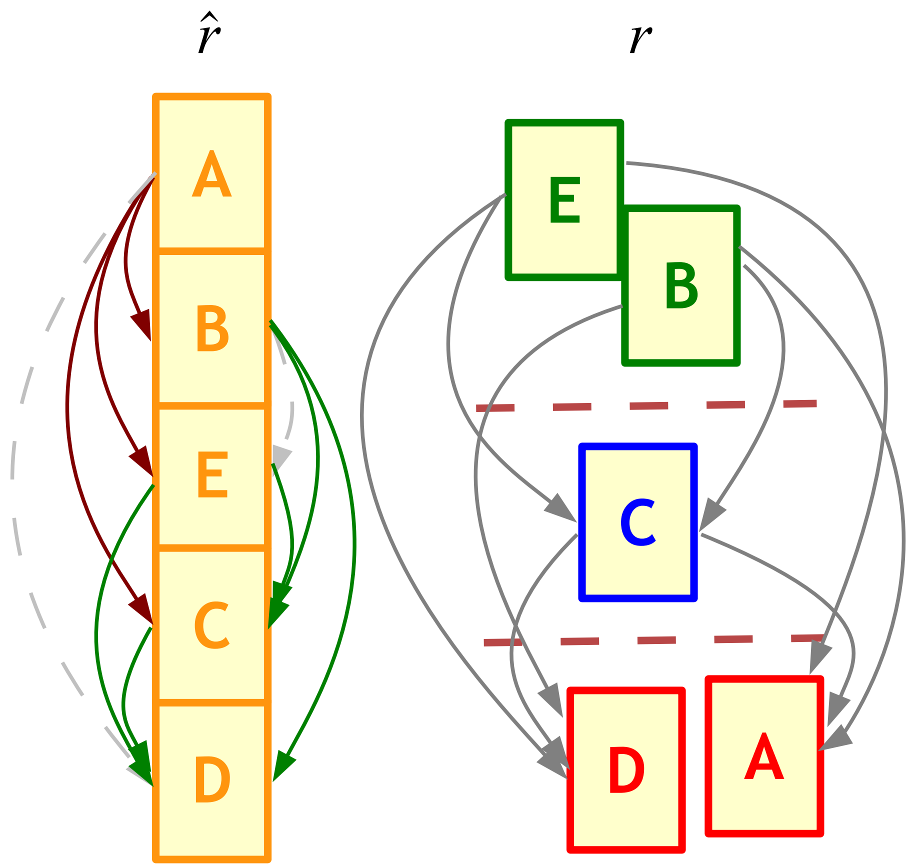
**Note:**
C-Index and m-AUC can be optimized by optimization of pairwise AUCs

# Normalized Discounted Cumulative Gain
[Jarvelin & Kekalainen, 2002]

- The original formulation of (normalized) discounted cumulative gain refers to this setting

$$DCG(r,\hat{r}) = \sum_{i=1}^{c} \frac{l(i)}{\log(i+1)}$$

  - the sum of the true (relevance) levels of the items
  - each item weighted by its rank in the predicted ranking
- Examples:
  - retrieval of relevant or irrelevant pages
    - 2 relevance levels
  - movie recommendation
    - 5 relevance levels



# AGENDA

1. Preference Learning Tasks
2. **Loss Functions**
   a. Evaluation of Rankings
   b. Weighted Measures
   c. Evaluation of Bipartite Rankings
   **d. Evaluation of Partial Rankings**
3. Preference Learning Techniques
4. Conclusions

# Evaluating Partial Structures in the Predicted Ranking

- For fixed types of partial structures, we have conventional measures
  - bipartite graphs → binary classification
    - accuracy, recall, precision, F1, etc.
    - can also be used when the items are labels!
      - e.g., accuracy on the set of labels for multilabel classification
  - multipartite graphs → ordinal classification
    - multiclass classification measures (accuracy, error, etc.)
    - regression measures (sum of squared errors, etc.)

- For general partial structures
  - some measures can be directly used on the reduced set of target preferences
    - Kendall's distance, Gamma coefficient
  - we can also use set measures on the set of binary preferences
    - both, the source and the target ranking consist of a set of binary preferences
    - e.g. Jaccard Coefficient
      - size of interesection over size of union of the binary preferences in both sets

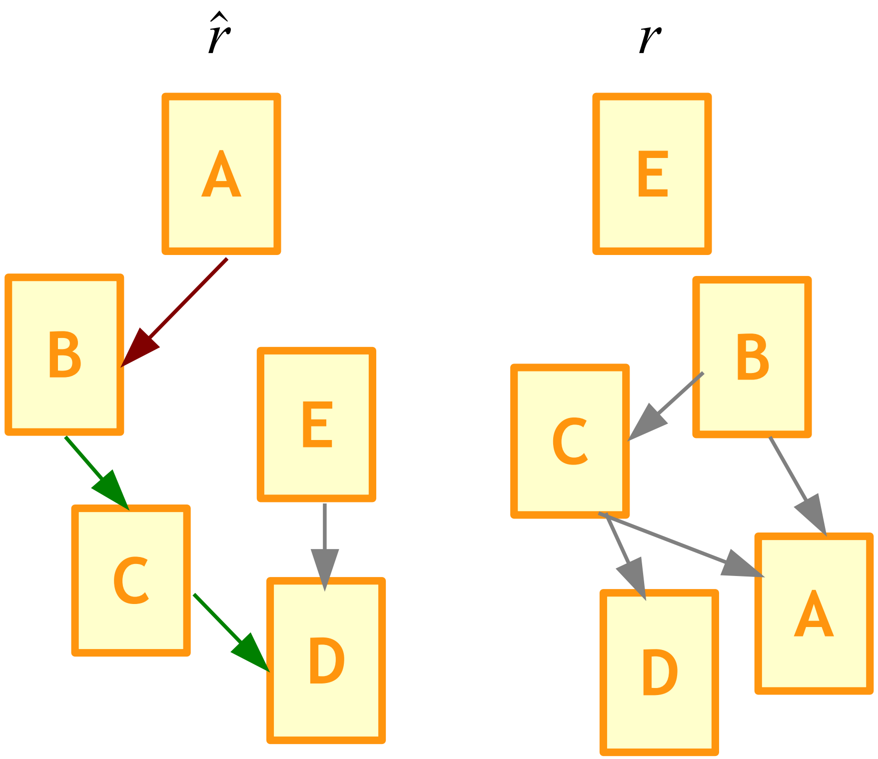
# Gamma Coefficient

- Key idea: normalized difference between
  - number of correctly ranked pairs (Kendall's distance)

    $$d = D_\tau(r, \hat{r})$$

  - number of incorrectly ranked pairs

    $$\bar{d} = \left| \left\{ (i,j) \mid r(x_i) < r(x_j) \wedge \hat{r}(x_i) < \hat{r}(x_j) \right\} \right|$$




- Gamma Coefficient
  [Goodman & Kruskal, 1979]

  $$\gamma(r, \hat{r}) = \frac{d - \bar{d}}{d + \bar{d}} \in [-1, +1]$$

  - Identical to Kendall's tau if both rankings are total
    - i.e., if $d + \bar{d} = \frac{c \cdot (c-1)}{2}$

$$\gamma(r, \hat{r}) = \frac{2-1}{2+1} = \frac{1}{3}$$

# References


- Cheng W., Rademaker M., De Baets B., Hüllermeier E.: *Predicting Partial Orders: Ranking with Abstention*. Proceedings ECML/PKDD-10(1): 215-230 (2010)
- Fürnkranz J., Hüllermeier E., Vanderlooy S.: *Binary Decomposition Methods for Multipartite Ranking*. Proceedings ECML/PKDD-09(1): 359-374 (2009)
- Gnen M., Heller G.: *Concordance probability and discriminatory power in proportional hazards regression*. Biometrika **92**(4):965–970 (2005)
- Goodman, L., Kruskal, W.: *Measures of Association for Cross Classifications*. Springer-Verlag, New York (1979)
- Hand D.J., Till R.J.: *A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems*. Machine Learning **45**(2):171-186 (2001)
- Jarvelin K., Kekalainen J.: *Cumulated gain-based evaluation of IR techniques*. ACM Transactions on Information Systems **20**(4): 422–446 (2002)
- Jonckheere, A. R.: *A distribution-free k-sample test against ordered alternatives*. Biometrika: 133–145 (1954)
- Kendall, M. *A New Measure of Rank Correlation*. Biometrika 30 (1-2): 81–89 (1938)
- Mann H. B.,Whitney D. R. *On a test of whether one of two random variables is stochastically larger than the other.* Annals of Mathematical Statistics, **18**:50–60 (1947)
- Spearman C. *The proof and measurement of association between two things.* American Journal of Psychology, **15**:72–101 (1904)

# AGENDA

1. Preference Learning Tasks
2. Performance Assessment and Loss Functions
3. **Preference Learning Techniques**
   a. Learning Utility Functions
   b. Learning Preference Relations
   c. Structured Output Prediction
   d. Model-Based Preference Learning
   e. Local Preference Aggregation
4. Conclusions

# Two Ways of Representing Preferences

- **Utility-based approach:** Evaluating single alternatives

$$U : \mathcal{A} \longrightarrow \mathbb{R}$$

- **Relational approach:** Comparing pairs of alternatives

| | | | |
|---|---|---|---|
| $a \succeq b$ | $\Leftrightarrow$ | $a$ is not worse than $b$ | weak preference |
| $a \succ b$ | $\Leftrightarrow$ | $(a \succeq b) \wedge (b \not\succeq a)$ | strict preference |
| $a \sim b$ | $\Leftrightarrow$ | $(a \succeq b) \wedge (b \succeq a)$ | indifference |
| $a \perp b$ | $\Leftrightarrow$ | $(a \not\succeq b) \wedge (b \not\succeq a)$ | incomparability |

## Utility Functions

- A **utility function** assigns a utility degree (typically a real number or an ordinal degree) to each alternative.
- Learning such a function essentially comes down to solving an (ordinal) **regression problem.**
- Often **additional conditions**, e.g., due to bounded utility ranges or monotonicity properties (→ *learning monotone models*)

- A **utility function induces a ranking** (total order), but not the other way around!
- But it can not represent more general relations, e.g., a **partial order**!
- The **feedback** can be **direct** (exemplary utility degrees given) or **indirect** (inequality induced by order relation):

$$(\boldsymbol{x}, u) \;\Rightarrow\; U(\boldsymbol{x}) \approx u, \qquad \boldsymbol{x} \succ \boldsymbol{y} \;\Leftrightarrow\; U(\boldsymbol{x}) > U(\boldsymbol{y})$$

absolute feedback relative feedback

# Predicting Utilities on Ordinal Scales

(Graded) multilabel classification

| X1 | X2 | X3 | X4 | A | B | C | D |
|---|---|---|---|---|---|---|---|
| 0.34 | 0 | 10 | 174 | -- | + | ++ | 0 |
| 1.45 | 0 | 32 | 277 | 0 | ++ | -- | + |
| 1.22 | 1 | 46 | 421 | -- | -- | 0 | + |
| 0.74 | 1 | 25 | 165 | 0 | + | + | ++ |
| 0.95 | 1 | 72 | 273 | + | 0 | ++ | -- |
| 1.04 | 0 | 33 | 158 | + | + | ++ | -- |

Collaborative filtering

| | P1 | P2 | P3 | ... | P38 | ... | P88 | P89 | P90 |
|---|---|---|---|---|---|---|---|---|---|
| U1 | 1 | | 4 | ... | | ... | | 3 | |
| U2 | | 2 | 2 | ... | | ... | 1 | | |
| ... | | | | ... | | ... | | | |
| U46 | ? | 2 | ? | ... | ? | ... | ? | ? | 4 |
| ... | | | | ... | | ... | | | |
| U98 | 5 | | | ... | | ... | 4 | | |
| U99 | | | 1 | ... | | ... | | 2 | |

Exploiting dependencies (correlations) between items (labels, products, …)

→ see work in MLC and RecSys communities

## Learning Utility Functions from Indirect Feedback

- A (latent) utility function can also be used to solve ranking problems, such as instance, object or label ranking
  → **ranking by (estimated) utility degrees (scores)**

**Object ranking**

$$(0.74, 1, 25, 165) \succ (0.45, 0, 35, 155)$$
$$(0.47, 1, 46, 183) \succ (0.57, 1, 61, 177)$$
$$(0.25, 0, 26, 199) \succ (0.73, 0, 46, 185)$$
$$(0.95, 0, 73, 133) \succ (0.25, 1, 35, 153)$$
$$(0.68, 1, 55, 147) \succ (0.67, 0, 63, 182)$$

→

Find a utility function that agrees as much as possible with the preference information in the sense that, for most examples,

$$\boxed{\boldsymbol{x}_i \succ \boldsymbol{y}_i \quad \Leftrightarrow \quad U(\boldsymbol{x}_i) > U(\boldsymbol{y}_i)}$$

**Instance ranking**

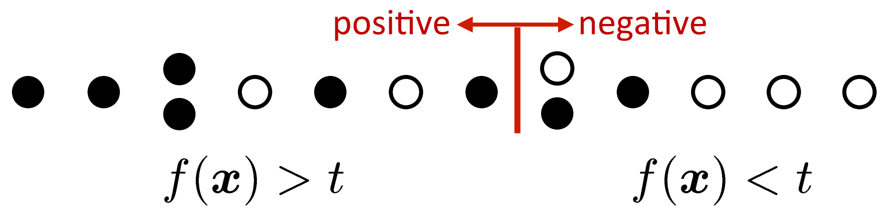
| X1 | X2 | X3 | X4 | class |
|---|---|---|---|---|
| 0.34 | 0 | 10 | 174 | -- |
| 1.45 | 0 | 32 | 277 | 0 |
| 1.22 | 1 | 46 | 421 | -- |
| 0.74 | 1 | 25 | 165 | ++ |
| 0.95 | 1 | 72 | 273 | + |

→

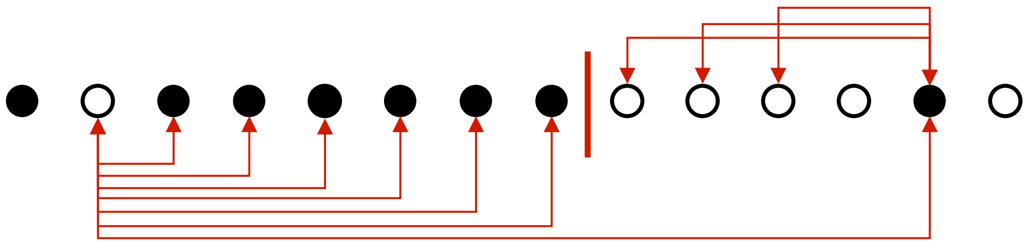
Absolute preferences given, so in principle an ordinal regression problem. However, the goal is to maximize ranking instead of classification performance.

## Ranking versus Classification

A ranker can be turned into a classifier via thresholding:




A good classifier is not necessarily a good ranker:



2 classification but
10 ranking errors

→ *learning **AUC-optimizing** scoring classifiers* !

## RankSVM and Related Methods (Bipartite Case)

- The idea is to minimize a convex upper bound on the empirical ranking error over a class of (kernelized) ranking functions:

convex upper bound on

$\mathbb{I}\left(f(\boldsymbol{x}) < f(\boldsymbol{x}')\right)$

$$f^* \in \arg\min_{f \in \mathcal{F}} \left\{ \frac{1}{|P| \cdot |N|} \underbrace{\sum_{\boldsymbol{x} \in P} \sum_{\boldsymbol{x}' \in N}}_{\text{check for all positive/negative pairs}} L(f, \boldsymbol{x}, \boldsymbol{x}') + \lambda \cdot R(f) \right\}$$

regularizer

# RankSVM and Related Methods (Bipartite Case)

- The bipartite RankSVM algorithm [Herbrich et al. 2000, Joachimes 2002]:

regularizer

$$f^* \in \arg \min_{f \in \mathcal{F}_K} \left\{ \frac{1}{|P| \cdot |N|} \sum_{\boldsymbol{x} \in P} \sum_{\boldsymbol{x}' \in N} (1 - (f(\boldsymbol{x}) - f(\boldsymbol{x}'))_+ + \frac{\lambda}{2} \cdot \|f\|_K^2 \right\}$$

hinge loss

reproducing kernel
Hilbert space (RKHS) with
kernel $\mathbf{K}$

→ learning comes down to solving a QP problem

# RankSVM and Related Methods (Bipartite Case)

- The bipartite RankBoost algorithm [Freund et al. 2003]:

$$f^* \in \arg \min_{f \in \mathcal{L}(\mathcal{F}_{base})} \left\{ \frac{1}{|P| \cdot |N|} \sum_{\boldsymbol{x} \in P} \sum_{\boldsymbol{x}' \in N} \exp\left(-(f(\boldsymbol{x}) - f(\boldsymbol{x}'))\right) \right\}$$

class of linear combinations of base functions

→ learning by means of boosting techniques

## Learning Utility Functions for Label Ranking

Label ranking is the problem of learning a function $\mathcal{X} \rightarrow \Omega$, with $\Omega$ the set of rankings (permutations) of a label set $\mathcal{Y} = \{y_1, y_2, \ldots, y_k\}$, from exemplary pairwise preferences $y_i \succ_{\boldsymbol{x}} y_j$.

Can be tackled by learning utility functions $U_1(\cdot), \ldots, U_k(\cdot)$ that are in appropriate agreement with the preferences in the training data. Given a new query $\boldsymbol{x}$, the labels are ranked according to utility degrees, i.e., a permutation $\pi$ is predicted such that

$$U_{\pi^{-1}(1)}(\boldsymbol{x}) > U_{\pi^{-1}(2)}(\boldsymbol{x}) > \ldots > U_{\pi^{-1}(k)}(\boldsymbol{x})$$

## **Label Ranking: Reduction to Binary Classification** [Har-Peled et al. 2002]

Proceeding from linear utility functions

$$U_i(\boldsymbol{x}) = \boldsymbol{w}_i \times \boldsymbol{x} = (w_{i,1}, w_{i,2}, \ldots, w_{i,m})(x_1, x_2, \ldots, x_m)^\top,$$

a binary preference $y_i \succ_{\boldsymbol{x}} y_j$ is equivalent to

$$U_i(\boldsymbol{x}) > U_j(\boldsymbol{x}) \;\Leftrightarrow\; \boldsymbol{w}_i \times \boldsymbol{x} > \boldsymbol{w}_j \times \boldsymbol{x} \;\Leftrightarrow\; (\boldsymbol{w}_i - \boldsymbol{w}_j) \times \boldsymbol{x} > 0$$

and can be modeled as a linear constraint

$$\underbrace{(\boldsymbol{w}_1, \boldsymbol{w}_2 \ldots \boldsymbol{w}_k)}_{\text{(m x k)-dimensional weight vector}} \times \underbrace{(0 \ldots 0 \; \boldsymbol{x} \; 0 \ldots 0 \;\; - \boldsymbol{x} \; 0 \ldots 0)}_{\text{positive example in the new instance space}}{}^\top > 0$$

Each **pairwise comparison** is turned into a **binary classification** example in a high-dimensional space!

# AGENDA

1. Preference Learning Tasks
2. Performance Assessment and Loss Functions
3. Preference Learning Techniques
    a. Learning Utility Functions
    b. **Learning Preference Relations**
    c. Structured Output Prediction
    d. Model-Based Preference Learning
    e. Local Preference Aggregation
4. Conclusions

## Learning Binary Preference Relations

- Learning **binary preferences** (in the form of predicates $\mathrm{P}(\mathbf{x},\mathbf{y})$) is often simpler, especially if the training information is given in this form, too.
- However, it implies an additional step, namely **extracting a ranking** from a (predicted) preference relation.
- This step is not always trivial, since a predicted preference relation may exhibit inconsistencies and may not suggest a unique ranking in an unequivocal way.

$$\text{instance } \boldsymbol{x} \longrightarrow \begin{array}{c|ccccc} f_{i,j} & y_1 & y_2 & y_3 & y_4 & y_5 \\ \hline y_1 & & 1 & 1 & 0 & 0 \\ y_2 & 0 & & 0 & \boxed{1} & 0 \\ y_3 & 0 & 1 & & 0 & 0 \\ y_4 & 1 & \boxed{0} & 1 & & 1 \\ y_5 & 1 & 1 & 1 & 0 & \end{array} \xrightarrow{\text{inference}} y_4 \succ y_5 \succ y_1 \succ y_3 \succ y_2$$

# Object Ranking: Learning to Order Things [Cohen et al. 99]

- In a first step, a **binary preference function** $\mathrm{PREF}$ is constructed; $\mathrm{PREF}(\mathbf{x},\mathbf{y}) \in [0,1]$ is a measure of the certainty that $\mathbf{x}$ should be ranked before $\mathbf{y}$, and $\mathrm{PREF}(\mathbf{x},\mathbf{y})=1- \mathrm{PREF}(\mathbf{y},\mathbf{x})$.
- This function is expressed as a linear combination of base preference functions:

$$\mathrm{PREF}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{N} w_i \cdot R_i(\boldsymbol{x},\boldsymbol{y})$$

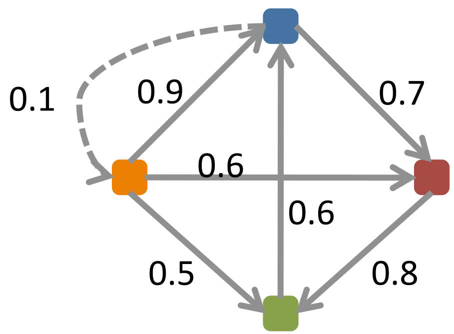
- The weights can be learned, e.g., by means of the weighted majority algorithm [Littlestone & Warmuth 94].
- In a second step, a total order is derived, which is as much as possible in agreement with the binary preference relation.

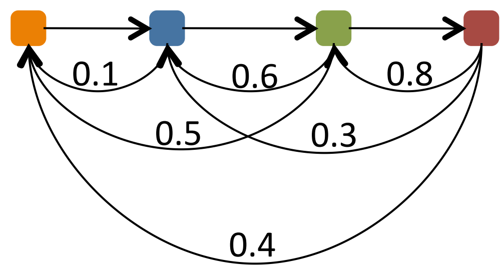
# Object Ranking: Learning to Order Things [Cohen et al. 99]

- The weighted feedback arc set problem: Find a permutation $\pi$ such that

$$\sum_{(\boldsymbol{x},\boldsymbol{y}):\pi(\boldsymbol{x})>\pi(\boldsymbol{y})} \mathrm{PREF}(\boldsymbol{x},\boldsymbol{y})$$

  becomes minimal.




cost = 0.1+0.6+0.8+0.5+0.3+0.4 = 2.7

# Object Ranking: Learning to Order Things [Cohen et al. 99]

- Since this is an NP-hard problem, it is solved heuristically.

**Input:** an instance set $X$; a preference function PREF
**Output:** an approximately optimal ordering function $\hat{\rho}$
***let*** $V = X$
***for*** each $v \in V$ ***do*** $\pi(v) = \sum_{u \in V} \mathrm{PREF}(v,u) - \sum_{u \in V} \mathrm{PREF}(u,v)$
***while*** $V$ is non-empty ***do***
    ***let*** $t = \arg\max_{u \in V} \pi(u)$
    ***let*** $\hat{\rho}(t) = |V|$
    $V = V - \{t\}$
    ***for*** each $v \in V$ ***do*** $\pi(v) = \pi(v) + \mathrm{PREF}(t,v) - \mathrm{PREF}(v,t)$
***endwhile***

- The algorithm successively chooses nodes having **maximal „net-flow"** within the remaining subgraph.
- It can be shown to provide a **2-approximation** to the optimal solution.

## Label Ranking: Learning by Pairwise Comparison (LPC) [Hüllermeier et al. 2008]

Label ranking is the problem of learning a function $\mathcal{X} \rightarrow \Omega$, with $\Omega$ the set of rankings (permutations) of a label set $\mathcal{Y} = \{y_1, y_2, \ldots, y_k\}$, from exemplary pairwise preferences $y_i \succ_{\boldsymbol{x}} y_j$.

LPC trains a model

$$\boxed{\mathcal{M}_{i,j} : \mathcal{X} \rightarrow [0,1]}$$

for all $i < j$. Given a query instance $\boldsymbol{x}$, this model is supposed to predict whether $y_i \succ y_j$ ($\mathcal{M}_{i,j}(\boldsymbol{x}) = 1$) or $y_j \succ y_i$ ($\mathcal{M}_{i,j}(\boldsymbol{x}) = 0$).

More generally, $\mathcal{M}_{i,j}(\boldsymbol{x})$ is the estimated probability that $y_i \succ y_j$.

Decomposition into $k(k-1)/2$ **binary classification problems**.

## Label Ranking: Learning by Pairwise Comparison (LPC) [Hüllermeier et al. 2008]

Training data (for the label pair A and B):

| X1 | X2 | X3 | | | class |
|---|---|---|---|---|---|
| 0.34 | 0 | 10 | | | 1 |
| ~~1.45~~ | ~~0~~ | ~~32~~ | | | |
| 1.22 | 1 | 46 | | | 0 |
| 0.74 | 1 | 25 | | | 1 |
| ~~0.95~~ | ~~1~~ | ~~72~~ | ~~273~~ | ~~B $\succ$ D, A $\succ$ D,~~ | |
| 1.04 | 0 | 33 | 158 | D $\succ$ A, **A $\succ$ B**, C $\succ$ B, A $\succ$ C | 1 |

| X1 | X2 | X3 | X4 | class |
|---|---|---|---|---|
| 0.34 | 0 | 10 | 174 | 1 |
| 1.22 | 1 | 46 | 421 | 0 |
| 0.74 | 1 | 25 | 165 | 1 |
| 1.04 | 0 | 33 | 158 | 1 |

## Label Ranking: Learning by Pairwise Comparison (LPC) [Hüllermeier et al. 2008]

At prediction time, a query instance is submitted to all models, and the predictions are combined into a binary preference relation:

predictions $\mathcal{M}_{i,j}(\boldsymbol{x})$ →

| | A | B | C | D |
|---|---|---|---|---|
| A | | 0.3 | 0.8 | 0.4 |
| B | 0.7 | | 0.7 | 0.9 |
| C | 0.2 | 0.3 | | 0.3 |
| D | 0.6 | 0.1 | 0.7 | |

## Label Ranking: Learning by Pairwise Comparison (LPC) [Hüllermeier et al. 2008]

At prediction time, a query instance is submitted to all models, and the predictions are combined into a binary preference relation:

predictions $\mathcal{M}_{i,j}(\boldsymbol{x})$ →

| | A | B | C | D | |
|---|---|---|---|---|---|
| A | | 0.3 | 0.8 | 0.4 | 1.5 |
| B | 0.7 | | 0.7 | 0.9 | 2.3 |
| C | 0.2 | 0.3 | | 0.3 | 0.8 |
| D | 0.6 | 0.1 | 0.7 | | 1.4 |

B $\succ$ A $\succ$ D $\succ$ C

From this relation, a ranking is derived by means of a **ranking procedure**. In the simplest case, this is done by sorting the labels according to their sum of **weighted votes**.

# AGENDA

1. Preference Learning Tasks
2. Performance Assessment and Loss Functions
3. Preference Learning Techniques
   a. Learning Utility Functions
   b. Learning Preference Relations
   c. **Structured Output Prediction**
   d. Model-Based Preference Learning
   e. Local Preference Aggregation
4. Conclusions

## Structured Output Prediction [Bakir et al. 2007]

- Rankings, multilabel classifications, etc. can be seen as specific types of **structured** (as opposed to scalar) **outputs**.
- Discriminative structured prediction algorithms infer a **joint scoring function on input-output pairs** and, for a given input, predict the output that maximises this scoring function.
- Joint feature map and scoring function

$$\phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d, \quad f(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{w}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}, \boldsymbol{y}) \rangle$$

- The learning problem consists of estimating the weight vector, e.g., using structural risk minimization.
- Prediction requires solving a **decoding problem**:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} f(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{w}) = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} \langle \boldsymbol{w}, \phi(\boldsymbol{x}, \boldsymbol{y}) \rangle$$

## **Structured Output Prediction** [Bakir et al. 2007]

- **Preferences** are expressed through **inequalities** on inner products:

loss function

$$\min_{\boldsymbol{w},\xi} \; |||w|||^2 + \nu \sum_{i=1}^{m} \xi_i$$

$$\text{s.t.} \quad \langle \boldsymbol{w}, \phi(\boldsymbol{x}_i, \boldsymbol{y}_i)\rangle - \langle \boldsymbol{w}, \phi(\boldsymbol{x}_i, \boldsymbol{y})\rangle \geq \Delta(\boldsymbol{y}_i, \boldsymbol{y}) - \xi_i \text{ for all } \boldsymbol{y} \in \mathcal{Y}$$

$$\xi_i \geq 0 \quad (i = 1, \ldots, m)$$

- The potentially huge **number of constraints** cannot be handled explicitly and calls for specific techniques (such as cutting plane optimization)

# AGENDA

1. Preference Learning Tasks
2. Performance Assessment and Loss Functions
3. Preference Learning Techniques
    a. Learning Utility Functions
    b. Learning Preference Relations
    c. Structured Output Prediction
    **d. Model-Based Preference Learning**
    e. Local Preference Aggregation
4. Conclusions

# Model-Based Methods for Ranking

- **Model-based approaches** to ranking proceed from specific assumptions about the possible rankings (**representation bias**) or make use of **probabilistic models** for rankings (parametrized probability distributions on the set of rankings).

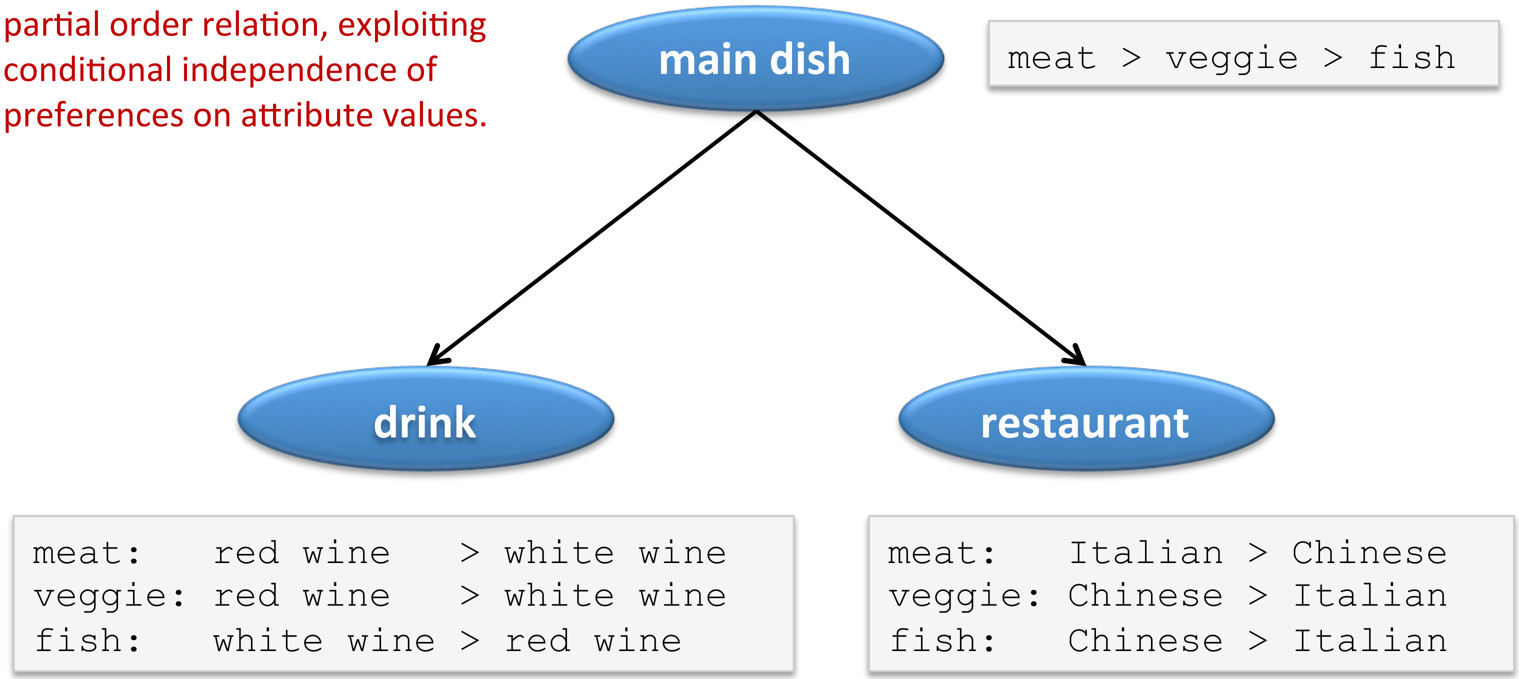
- In the following, we shall see examples of both type:
  - Restriction to lexicographic preferences
  - Conditional preference networks (CP-nets)
  - ~~Label ranking using the Plackett-Luce model~~ see our talk tomorrow

## Learning Lexicographic Preference Models [Yaman et al. 2008]

- Suppose that objects are represented as feature vectors of length $m$, and that each attribute has $k$ values.
- For $n=k^m$ objects, there are $n!$ permutations (rankings).
- A **lexicographic order** is uniquely determined by
  - a total order of the attributes
  - a total order of each attribute domain
- **Example:** Four binary attributes ($m=4$, $k=2$)
  - there are $16! \approx 2 \cdot 10^{13}$ rankings
  - but only $(2^4) \cdot 4! = 384$ of them can be expressed in terms of a lexicographic order
- [Yaman et al. 2008] present a learning algorithm that explictly maintains the version space, i.e., the attribute-orders compatible with all pairwise preferences seen so far (assuming binary attributes with 1 preferred to 0). Predictions are derived based on the „votes“ of the consistent models.

# Learning Conditional Preference (CP) Networks [Chevaleyre et al. 2010]

Compact representation of a partial order relation, exploiting conditional independence of preferences on attribute values.

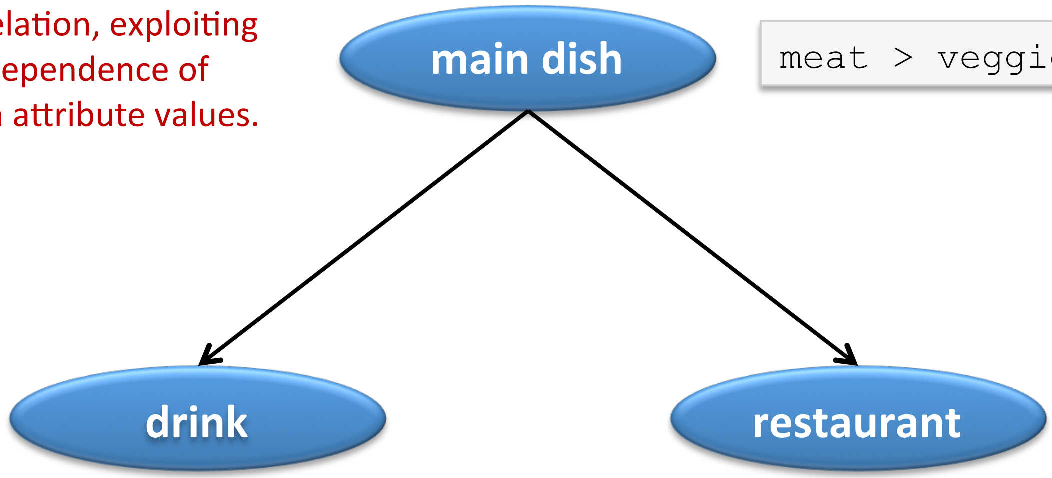



Induces partial order relation, e.g.,

```
(meat, red wine, Italian)    >  (meat, white wine, Chinese)
(fish, white wine, Chinese)  >  (fish, red wine, Chinese)
(meat, white wine, Italian)  ?  (meat, red wine, Chinese)
```

## Learning Conditional Preference (CP) Networks [Chevaleyre et al. 2010]

Compact representation of a partial order relation, exploiting conditional independence of preferences on attribute values.




```
meat > veggie > fish
```

```
meat:   red wine   > white wine
veggie: red wine   > white wine
fish:   white wine > red wine
```

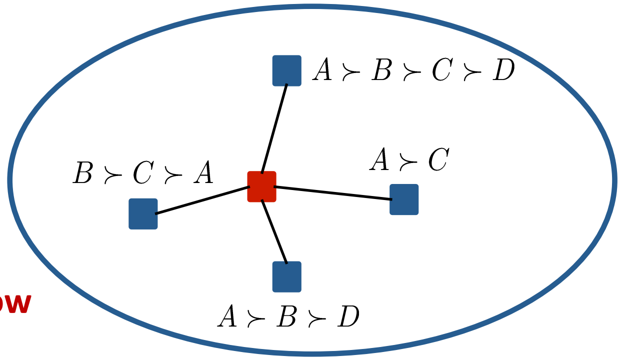
```
meat:   Italian > Chinese
veggie: Chinese > Italian
fish:   Chinese > Italian
```

**Training data** (possibly noisy):

```
(meat, red wine, Italian)          > (veggie, red wine, Italian)
(fish, whited wine, Chinease)      > (veggie, red wine, Chinease)
(veggie, whited wine, Chinease)    > (veggie, red wine, Italian)
            …                      …            …
```

# AGENDA

1. Preference Learning Tasks
2. Performance Assessment and Loss Functions
3. Preference Learning Techniques
   a. Learning Utility Functions
   b. Learning Preference Relations
   c. Structured Output Prediction
   d. Model-Based Preference Learning
   e. ~~Local Preference Aggregation~~ **see our talk tomorrow**
4. Conclusions



## Summary of Main Algorithmic Principles

- **Reduction** of ranking to (binary) classification (e.g., constraint classification, LPC)
- **Direct optimization** of (regularized) smooth approximation of ranking losses (RankSVM, RankBoost, …)
- **Structured output prediction**, learning joint scoring („matching“) function
- Learning parametrized **probabilistic ranking models** (e.g., Plackett-Luce)
- **Restricted model classes**, fitting (parametrized) deterministic models (e.g., lexicographic orders)
- **Lazy learning**, local preference aggregation (lazy learning)

# References


- G. Bakir, T. Hofmann, B. Schölkopf, A. Smola, B. Taskar and S. Vishwanathan. *Predicting structured data*. MIT Press, 2007.
- W. Cheng, K. Dembczynski and E. Hüllermeier. *Graded Multilabel Classification: The Ordinal Case*. ICML-2010, Haifa, Israel, 2010.
- W. Cheng, K. Dembczynski and E. Hüllermeier. *Label ranking using the Plackett-Luce model*. ICML-2010, Haifa, Israel, 2010.
- W. Cheng and E. Hüllermeier. *Predicting partial orders: Ranking with abstention*. ECML/PKDD-2010, Barcelona, 2010.
- W. Cheng, C. Hühn and E. Hüllermeier. *Decision tree and instance-based learning for label ranking*. ICML-2009.
- Y. Chevaleyre, F. Koriche, J. Lang, J. Mengin, B. Zanuttini. *Learning ordinal preferences on multiattribute domains: The case of CP-nets*. In: J. Fürnkranz and E. Hüllermeier (eds.) Preference Learning, Springer-Verlag, 2010.
- W.W. Cohen, R.E. Schapire and Y. Singer. *Learning to order things*. Journal of Artificial Intelligence Research, 10:243–270, 1999.
- Y. Freund, R. Iyer, R. E. Schapire and Y. Singer. *An efficient boosting algorithm for combining preferences*. Journal of Machine Learning Research, 4:933–969, 2003.
- J. Fürnkranz, E. Hüllermeier, E. Mencia, and K. Brinker. *Multilabel Classification via Calibrated Label Ranking*. Machine Learning 73(2):133-153, 2008.
- J. Fürnkranz, E. Hüllermeier and S. Vanderlooy. *Binary decomposition methods for multipartite ranking*. Proc. ECML-2009, Bled, Slovenia, 2009.
- D. Goldberg, D. Nichols, B.M. Oki and D. Terry. *Using collaborative filtering to weave and information tapestry*. Communications of the ACM, 35 (12):61–70, 1992.
- S. Har-Peled, D. Roth and D. Zimak. *Constraint classification: A new approach to multiclass classification*. Proc. ALT-2002.
- R. Herbrich, T. Graepel and K. Obermayer. *Large margin rank boundaries for ordinal regression*. Advances in Large Margin Classifiers, 2000.
- E. Hüllermeier, J. Fürnkranz, W. Cheng and K. Brinker. *Label ranking by learning pairwise preferences*. Artificial Intelligence, 172:1897–1916, 2008.
- T. Joachims. *Optimizing search engines using clickthrough data*. Proc. KDD 2002.
- N. Littlestone and M.K. Warmuth. *The weighted majority algorithm*. Information and Computation, 108(2): 212–261, 1994.
- G. Tsoumakas and I. Katakis. *Multilabel classification: An overview*. Int. J. Data Warehouse and Mining, 3:1–13, 2007.
- F. Yaman, T. Walsh, M. Littman and M. desJardins. *Democratic Approximation of Lexicographic Preference Models*. ICML-2008.

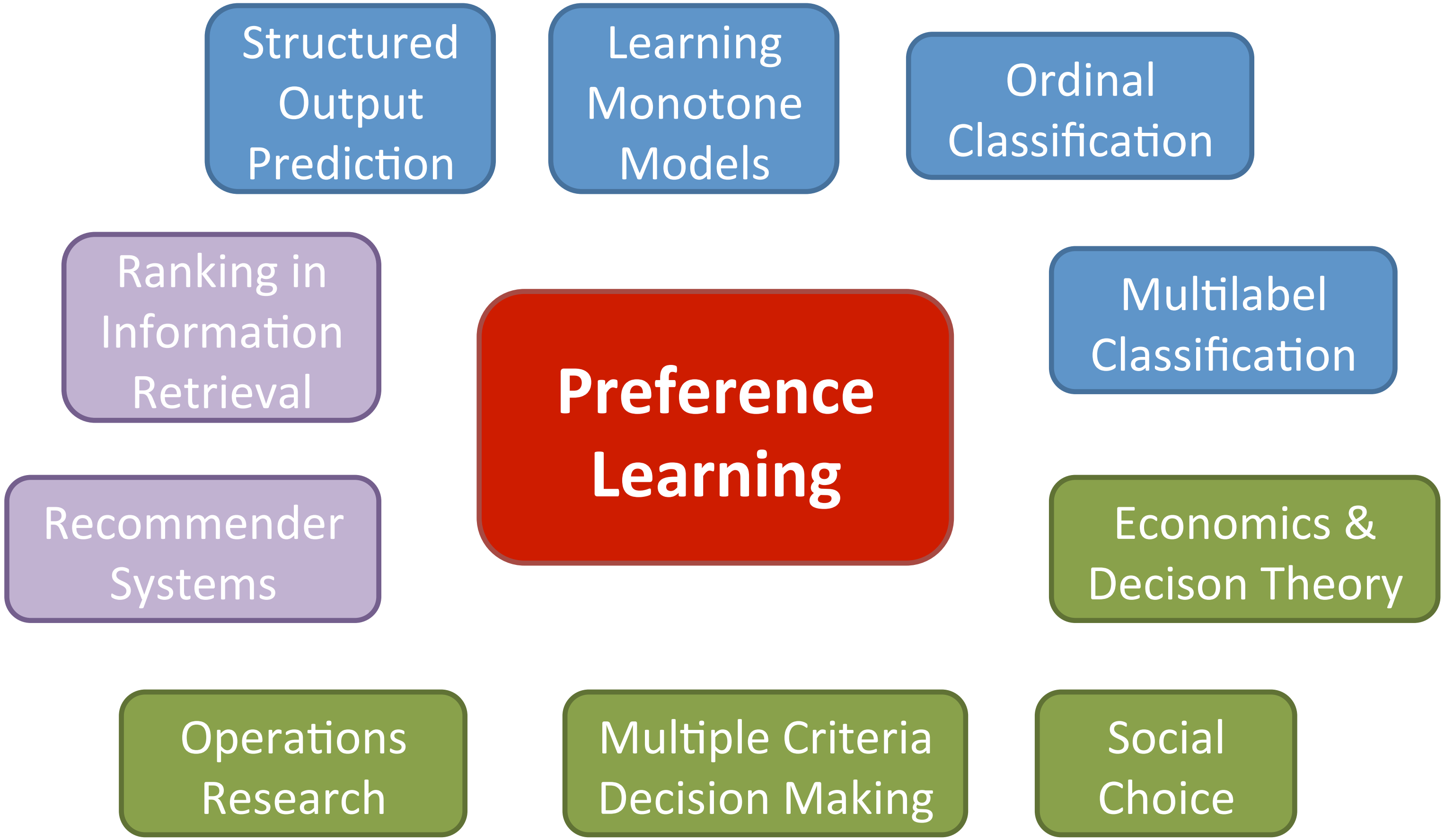
# AGENDA

1. Preference Learning Tasks
2. Performance Assessment and Loss Functions
3. Preference Learning Techniques
4. **Conclusions**

## Conclusions

- Preference learning is an **emerging subfield** of machine learning, with many **applications** and **theoretical challenges**.
- Prediction of **preference models** instead of scalar outputs (like in classification and regression), hitherto with a focus on **rankings.**
- Many existing machine learning problems can be cast in the framework of preference learning (→ preference learning „in a broad sense")
- **„Qualitative" alternative** to conventional numerical approaches
  - pairwise comparison instead of numerical evaluation,
  - order relations instead of individual assessment.
- Still many **open problems** (unified framework, predictions more general than rankings, incorporating numerical information, etc.)
- **Interdisciplinary field**, connections to many other areas.

## Connections to Other Fields



# Edited Book on Preference Learning

Preference Learning: An Introduction
A Preference Optimization based Unifying Framework for Supervised Learning Problems

**Part I – Label Ranking**

Label Ranking Algorithms: A Survey
Preference Learning and Ranking by Pairwise Comparison
Decision Tree Modeling for Ranking Data
Co-regularized Least-Squares for Label Ranking

**Part II – Instance Ranking**

A Survey on ROC-Based Ordinal Regression
Ranking Cases with Classification Rules

**Part III – Object Ranking**

A Survey and Empirical Comparison of Object Ranking Methods
Dimension Reduction for Object Ranking
Learning of Rule Ensembles for Multiple Attribute Ranking Problems

**Part IV – Preferences in Multiattribute Domains**

Learning Lexicographic Preference Models
Learning Ordinal Preferences on Multiattribute Domains: the Case of CP-nets
Choice-Based Conjoint Analysis: Classification vs. Discrete Choice Models
Learning Aggregation Operators for Preference Modeling

**Part V – Preferences in Information Retrieval**

Evaluating Search Engine Relevance with Click-Based Metrics
Learning SVM Ranking Function from User Feedback Using Document Metadata and Active Learning in the Biomedical Domain

**Part VI – Preferences in Recommender Systems**

Learning Preference Models in Recommender Systems
Collaborative Preference Learning
Discerning Relevant Model Features in a Content-Based Collaborative Recommender System


J. Fürnkranz &
E. Hüllermeier (eds.)
Preference Learning
Springer-Verlag 2010

# Edited Book on Preference Learning


J. Fürnkranz &
E. Hüllermeier (eds.)
Preference Learning
Springer-Verlag 2010

**Preference Learning: An Introduction**
A Preference Optimization based Unifying Framework for Supervised Learning Problems

**Part I – Label Ranking**

**Label Ranking Algorithms: A Survey**
Preference Learning and Ranking by Pairwise Comparison
Decision Tree Modeling for Ranking Data
Co-regularized Least-Squares for Label Ranking

**Part II – Instance Ranking**

**A Survey on ROC-Based Ordinal Regression**
Ranking Cases with Classification Rules

**Part III – Object Ranking**

**A Survey and Empirical Comparison of Object Ranking Methods**
Dimension Reduction for Object Ranking
Learning of Rule Ensembles for Multiple Attribute Ranking Problems

**Part IV – Preferences in Multiattribute Domains**

Learning Lexicographic Preference Models
Learning Ordinal Preferences on Multiattribute Domains: the Case of CP-nets
Choice-Based Conjoint Analysis: Classification vs. Discrete Choice Models
Learning Aggregation Operators for Preference Modeling

**Part V – Preferences in Information Retrieval**

Evaluating Search Engine Relevance with Click-Based Metrics
Learning SVM Ranking Function from User Feedback Using Document Metadata and Active Learning in the Biomedical Domain

**Part VI – Preferences in Recommender Systems**

**Learning Preference Models in Recommender Systems**
Collaborative Preference Learning
Discerning Relevant Model Features in a Content-Based Collaborative Recommender System

**includes several introductions and survey articles**

## Preference Learning Website

**http://www.preference-learning.org/**

- Working groups
- Software
- Data Sets
- Workshops
- Tutorials
- Books
- ...