Invited Talk:

Object Ranking

Toshihiro Kamishima

http://www.kamishima.net/

National Institute of Advanced Industrial Science and Technology (AIST), Japan

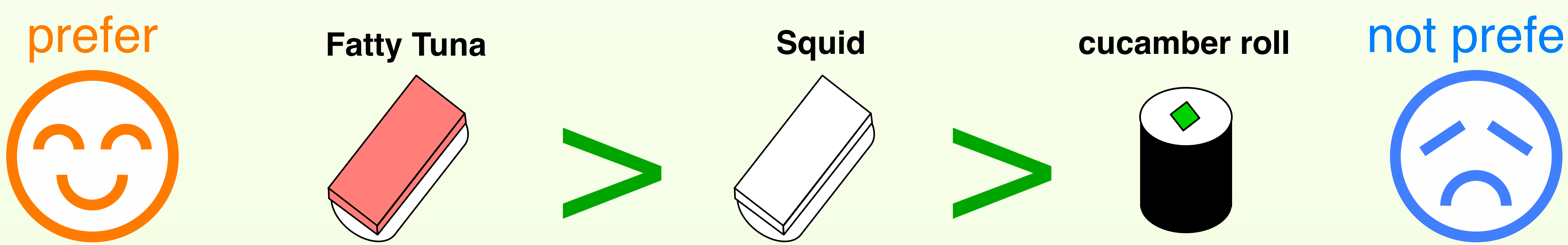
Preference Learning Workshop (PL-09) @ ECML/PKDD 2009, Bled, Slovenia

Introduction

- **Object Ranking:** Task to learn a function for ranking objects from sample orders
- Discussion about methods for this task by connecting with the probabilistic distributions of rankings
- Several properties of object ranking methods

Order / Ranking

object sequence sorted according to a particular preference or property

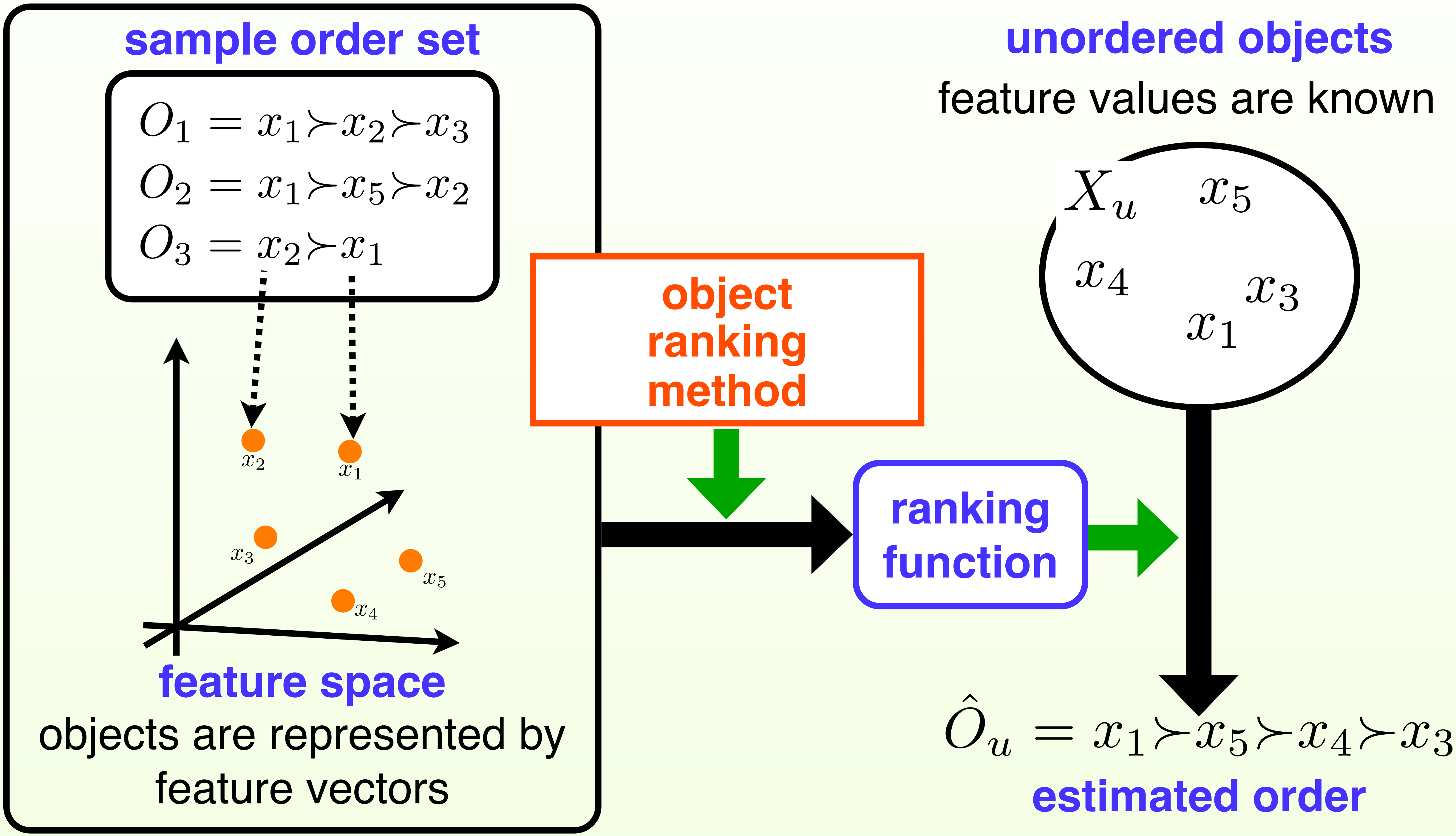


ex. an order sorted according to my preference in *sushi*
“I prefer fatty tuna to squid” but “The degree of preference is not specified”

Outline

- **What's object ranking**
  - Definition of an object ranking task
  - Connection with regression and ordinal regression
  - Measuring the degree of preference
- **Probability distributions of rankings**
  - Thurstonian, paired comparison, distance-based, and multistage
- **Six methods for object ranking**
  - Cohen's method, RankBoost, SVOR (a.k.a. RankingSVM), OrderSVM, ERR, and ListNet
- **Properties of object ranking methods**
  - Absolute and relative ranking
- **Conclusion**

Object Raking Task



- objects that don't appeared in training samples have to be ordered by referring feature vectors of objects

Object Ranking vs Regression

Object Ranking: regression targeting orders

generative model of object ranking

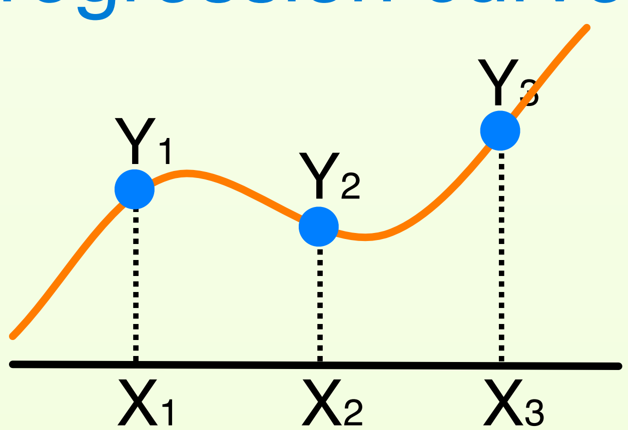
input | regression order | permutation noise | sample

X_1, X_2, X_3 → ranking function: $X_1 \succ X_2 \succ X_3$ → random permutation: $X_1 \succ X_3 \succ X_2$ → $X_1 \succ X_3 \succ X_2$

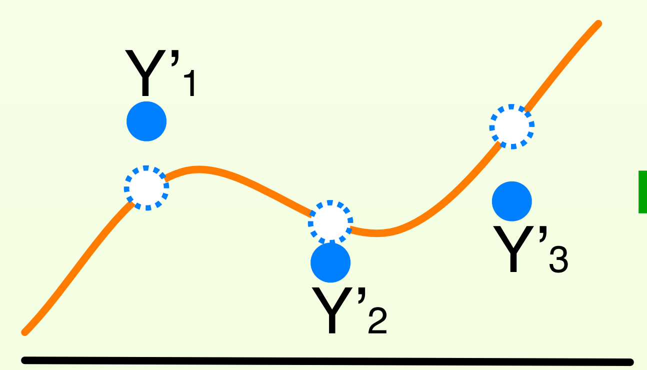
generative model of regression

input | regression curve | additive noise | sample

X_1, X_2, X_3 →



→



→ Y'_1, Y'_2, Y'_3

Ordinal Regression

Ordinal Regression [McCullagh 80, Agresti 96]
Regression whose target variable is ordered categorical

Ordered Categorical Variable
Variable can take one of a predefined set of values that are ordered
ex. { good, fair, poor}

Differences between "ordered categories" and "orders"	
Ordered Category	**Order**
The # of grades is **finite**	The # of grades is **infinite**
ex: For a domain {good, fair, poor}, the # of grades is limited to three	
Absolute Information is contained	It contains purely **relative** information
ex: While "good" indicates absolutely preferred, "$x_1 > x_2$" indicates that x_1 is relatively preferred to x_2	

Object ranking is more general problem than ordinal regression as a learning task

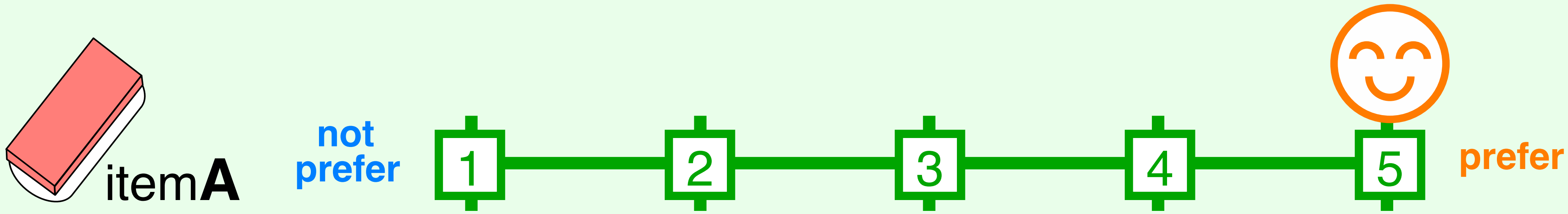
Measuring Preference

ordinal regression (ordered categories)

Scoring Method / Rating Method

Using scales with scores (ex. 1,2,3,4,5) or ratings (ex. gold, silver, bronze)

The user selects “5” in a five-point scale if he/she prefers the item A

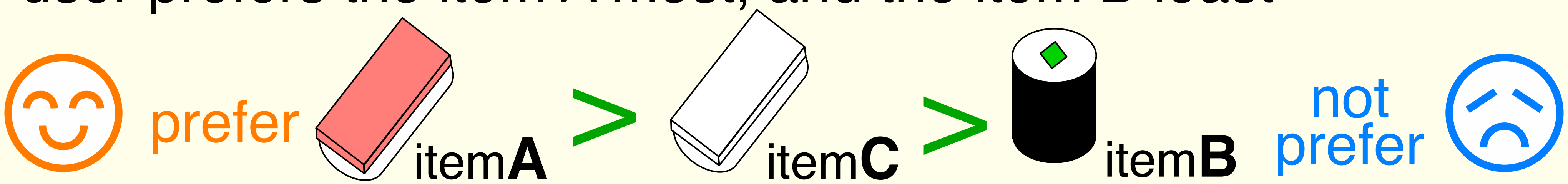


object ranking (orders)

Ranking Method

Objects are sorted according to the degree of preference

The user prefers the item A most, and the item B least



Demerit of Scoring / Rating Methods

Difficulty in caliblation over subjects / items

Mappings from the preference in users' mind to rating scores differ among users

- Standardizing rating scores by subtracting user/item mean score is very important for good prediction [Herlocker+ 99, Bell+ 07]
- Replacing scores with rankings contributes to good prediction, even if scores are standardized [Kmaishima 03, Kamishima+ 06]

presentation bias

The wrong presentation of rating scales causes biases in scores

- When prohibiting neutral scores, users select positive scores more frequently [Cosley+ 03]
- Showing predicted scores affects users' evaluation [Cosley+ 03]

Demerit of Ranking Methods

Lack of absolute information

Orders don't provide the absolute degree of preference

- Even if "$x_1 > x_2$" is specified, x_1 might be the second worst

Difficulty in evaluating many objects

Ranking method is not suitable for evaluating many objects at the same time

- Users cannot correctly sort hundreds of objects
- In such a case, users have to sort small groups of objects in many times

Distributions of Rankings

generative model of object ranking

regression order + permutation noise

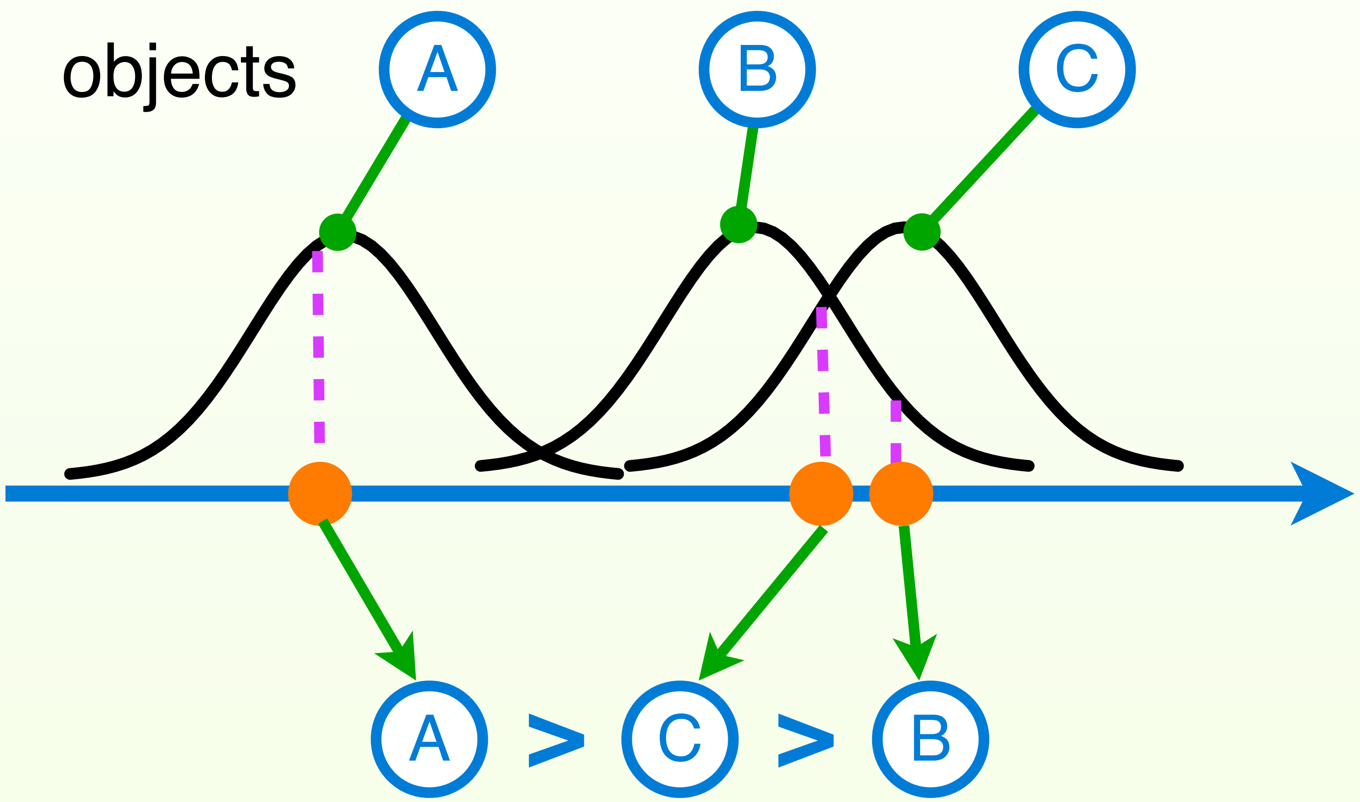
The permutation noise part is modeled by using probabilistic distributions of rankings

4 types of distributions for rankings [Crichlow+ 91, Marden 95]

- **Thurstonian:** Objects are sorted according to the objects' scores
- **Paired comparison:** Objects are ordered in pairwise, and these ordered pairs are combined
- **Distance-based:** Distributions are defined based on the distance between a modal order and sample one
- **Multistage:** Objects are sequentially arranged top to end

Thurstonian

Thurstonian model (a.k.a Order statistics model)
Objects are sorted according to the objects' scores



For each object, the corresponding scores are sampled from the associated distributions

Sort objects according to the sampled scores

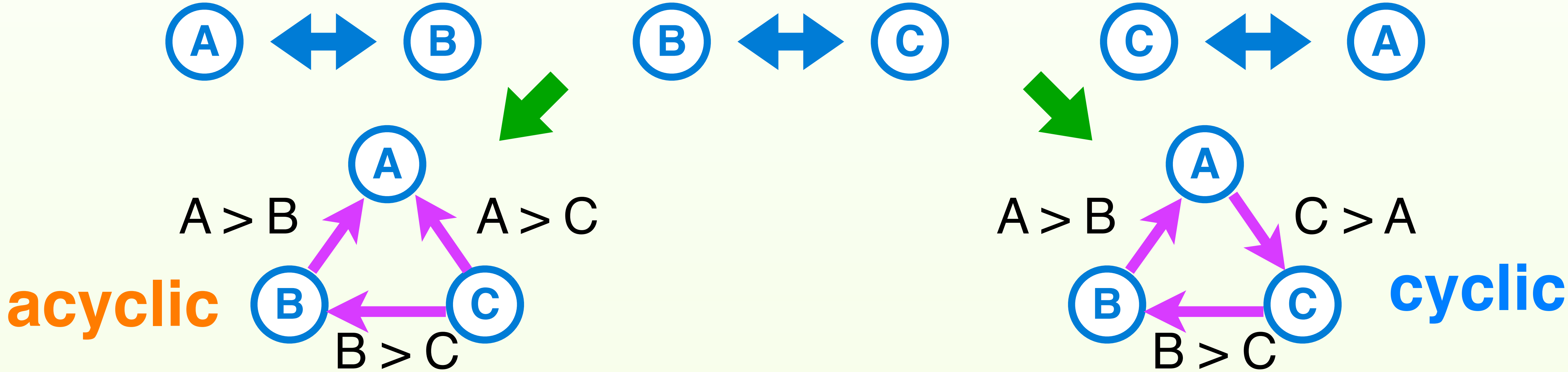
distribution of scores

- **Normal Distribution:** Thurstone's law of comparative judgment [Thurstone 27]
- **Gumbel Distribution:** CDF is $1 - \exp(-\exp((x_i - \mu_i)/\sigma))$

Paired Comparison

Paired comparison model
Objects are ordered in pairwise, and these ordered pairs are combined

Objects are ordered in pairwise



generate the order: A > B > C

Abandon and retry

parameterization

- **Babinton Smith model:** saturated model with ${}_nC_2$ paramaters [Babington Smith 50]
- **Bradley-Terry model:** $\Pr[x_i \succ x_j] = \frac{v_i}{v_i + v_j}$ [Bradley+ 52]

Distance between Orders

squared Euclidean distance between two rank vectors

	A	B	C	D
O_1: A > B > C > D	1	2	3	4
O_2: D > B > A > C	3	2	4	1

rank vectors

Spearman distance

Spearman footrule

Manhattan distance between two rank vectors

O_1: A > B > C	A > B	A > C	B > C
	NO!	**OK**	**OK**
O_2: B > A > C	B > A	A > C	B > C

decompose into ordered pairs

Kendall distance

# of discordant pairs between two orders

Distance-based

Distance-based model
Distributions are defined based on the distance between orders

$$\Pr[O] = \underbrace{C(\lambda)}_{\text{normalization factor}} \exp(-\overbrace{\lambda}^{\text{dispersion parameter}} \underbrace{d(O, \overbrace{O_0}^{\text{modal order/ranking}})}_{\text{distance}})$$

distance

- **Spearman distance:** Mallows' θ model
- **Kendall distance:** Mallows' ϕ model

These are the special cases of Mallows' model (ϕ=1 or θ=1),
which is a paired comparison model that defined as:

$$\Pr[x_i \succ x_j] = \frac{\theta^{i-j}\phi^{-1}}{\theta^{i-j}\phi^{-1}+\theta^{j-i}\phi}$$

[Mallows 57]

Multistage

Multistage model
Objects are sequentially arranged top to end

Plackett-Luce model [Plackett 75]

ex. objects {A,B,C,D} is sorted into A > C > D > B

$$\Pr[A] = \frac{\theta_A}{\theta_A + \theta_B + \theta_C + \theta_D}$$

a param of the top object

total sum of params

$$\Pr[A>C \mid A] = \frac{\theta_C}{\theta_B + \theta_C + \theta_D}$$

a param of the second object

params for A is eliminated

$$\Pr[A>C>D \mid A>C] = \frac{\theta_D}{\theta_B + \theta_D}$$

$\Pr[A>C>D>B \mid A>C>D] = \theta_B / \theta_B = 1$

The probability of the order, A > C > D > B, is

$\Pr[A>C>D>B] = \Pr[A] \Pr[A>C \mid A] \Pr[A>C>D \mid A>C]\ 1$

Object Ranking Methods

Object Ranking Methods

- **permutation noise model:** orders are permutated accoding to the distributions of rankings
- **regression order model:** representation of the most probable rankings
- **loss function:** the definition of the “goodness of model”
- **optimization method:** tuning model parameters

connection between distributions and permutation noise model

- **Thurstonian:** Expected Rank Regression (ERR)
- **Paired comparison:** Cohen’s method
- **Distance-based:** RankBoost, Support Vector Ordinal Regression (SVOR, a.k.a RankingSVM), OrderSVM
- **Multistage:** ListNet

Regression Order Model

linear ordering: Cohen's method

1. Given the features of any object pairs, $\mathbf{x}_i$ and $\mathbf{x}_j$, $f(\mathbf{x}_i, \mathbf{x}_j)$ represents the preference of the object i to the object j
2. All objects are sorted so as to maximize: $\sum_{\mathbf{x}_i \succ \mathbf{x}_j} f(\mathbf{x}_i, \mathbf{x}_j)$

This is known as **Linear Ordering Problem** in an OR literature [Grötschel+ 84], and is NP-hard ➡ Greedy searching solution $O(n^2)$

sorting by scores: ERR, RankBoost, SVOR, OrderSVM, ListNet

1. Given the features of an object, $\mathbf{x}_i$, $f(\mathbf{x}_i)$ represents the preference of the object i
2. All objects are sorted according to the values of $f(\mathbf{x})$

Computational complexity for sorting is $O(n \log(n))$

Cohen's Method

[Cohen+ 99]

- **permutation noise model = paired comparison**
- **regression order model = linear ordering**

sample orders are decomposed into ordered pairs

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

training sample orders

$$A \succ B,\, A \succ B,\, B \succ C$$
$$D \succ E,\, D \succ B,\, D \succ C,\, \cdots$$
$$A \succ D,\, A \succ C,\, D \succ C$$

ordered pairs

the preference function that one object precedes the other

$$f(\mathbf{x}_i, \mathbf{x}_j) = \Pr[\mathbf{x}_i \succ \mathbf{x}_j; \mathbf{x}_i, \mathbf{x}_j]$$

Unordered objects can be sorted by solving linear ordering problem

RankBoost

[Freund+ 03]

- permutation noise model = distance based (Kendall distance)
- regression order model = sorting by scores

find a linear combination of weak hypotheses by boosting

objects

weak hypotheses

partial information about the target order

A B

$h_t(A)$ $h_t(B)$

$h_t(A) \succ h_t(B)$

or

$h_t(B) \succ h_t(A)$

score function: $f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$

This function is learned so that minimizing the number of discordant pairs

↓

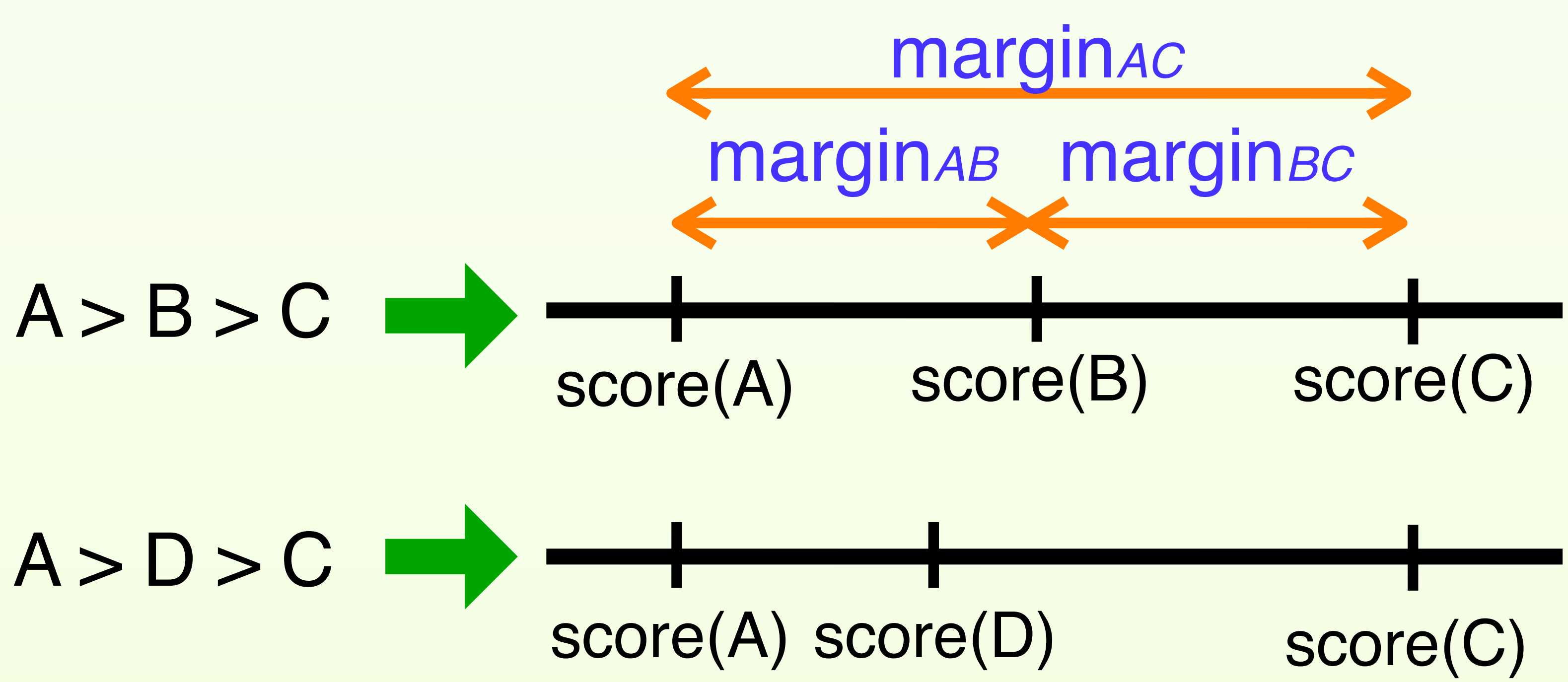
minimizing the Kendall distance between samples and the regression order

Support Vector Ordinal Regression (SVOR; a.k.a RankingSVM) [Herbrich+ 98, Joachims 02]

- **permutation noise model = distance based (Kendall distance)**
- **regression order model = sorting by scores**

find a score function that maximally separates preferred objects from non-preferred objects

sample orders | score & margin | Objective



maximize:

$$\sum_{X,Y} \mathrm{margin}_{XY}$$

OrderSVM

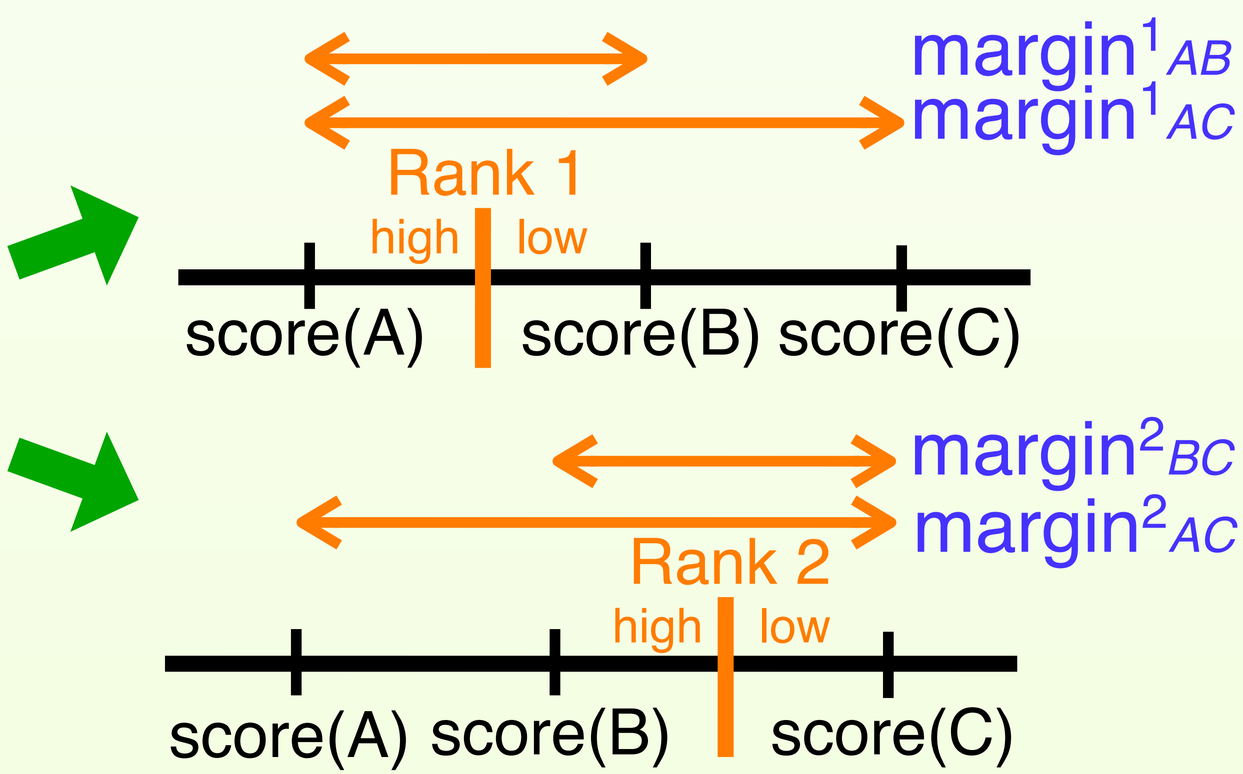
[Kazawa+ 05]

- **permutation noise model = distance based (Spearman footrule)**
- **regression order model = sorting by scores**

find a score function which maximally separates higher-ranked objects from lower-ranked ones on average

sample orders | score & margin | Objective

A > B > C



maximize:

$$\sum_j \sum_{X,Y} \mathrm{margin}^j_{XY}$$

SVM and Distance-based Model

SVOR (RankingSVM)

minimizing the # of misclassifications in orders of object pairs

↓

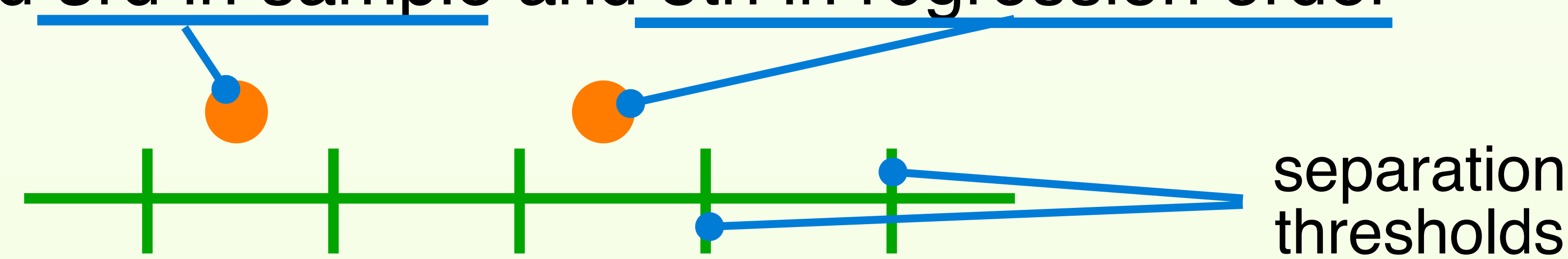
minimizing the Kendall distance between regression order and samples

OrderSVM

separate the objects that ranked lower than *j*-th from the higher ones, and these separations are summed over all ranks *j*

↓

ex: object A is ranked 3rd in sample and 5th in regression order



$\boxed{\text{\# of misclassifications = absolute difference between ranks}}$

↓

minimizing the Spearman footrule between regression order and samples

Expected Rank Regression (ERR)

[Kamishima+ 05]

- **permutation noise model = Thurstonian**
- **regression order model = sorting by scores**

expected ranks in a complete order are estimated from samples, and a score function is learned by regression from pairs of expected ranks and feature vectors of all objects

complete order

A > B > C > D > E > F

consisting of all possible objects, free from permutation noise, unobserved

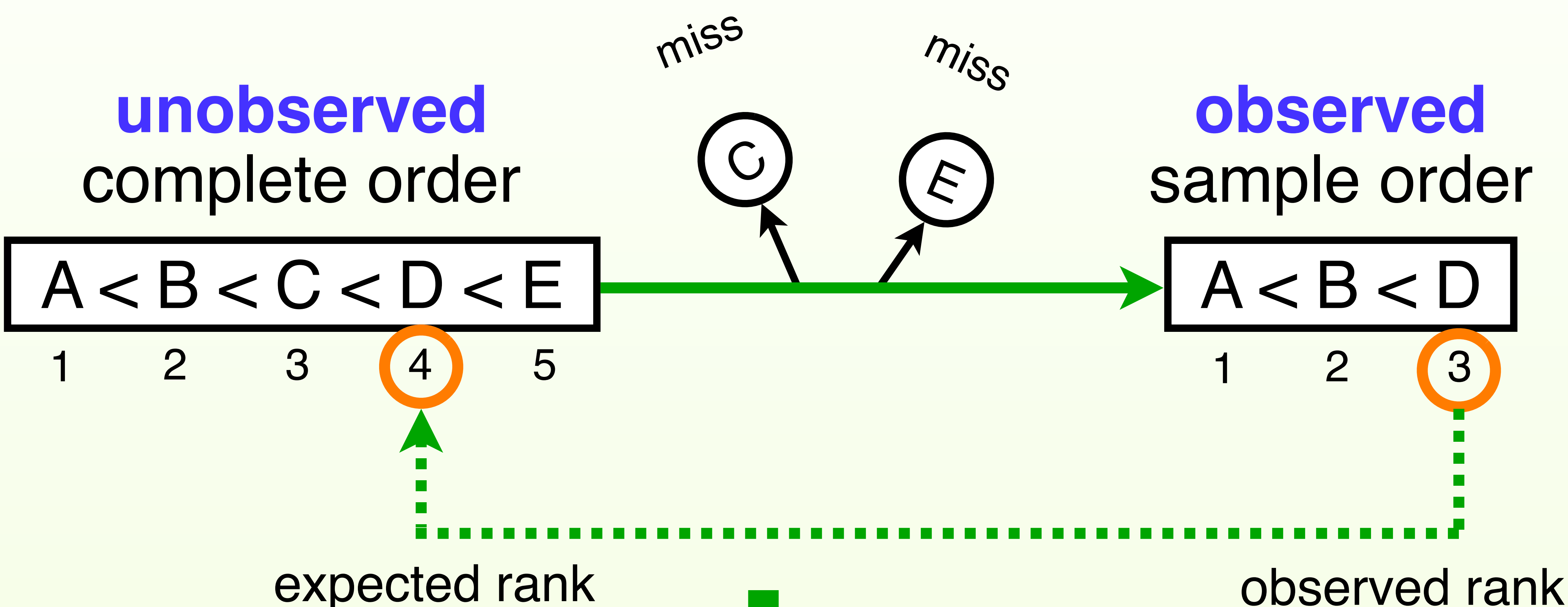
sample order

A > C > D

consisting of sub-sampled objects, with permutation noise, observed

Because expected ranks are considered as the location parameters of score distributions, this method is based on Thurstonian model

Expected Rank



$$\text{expectation of ranks in a unobserved complete order} \propto \frac{\text{rank in a observed order}}{(\text{length of a observed order }) + 1}$$

[Arnold+ 92]

ListNet

[Cao+ 07]

- **permutation noise model = Multistage**
- **regression order model = sorting by scores**

Straightforward modification of Plackett-Luce model, and parameters are optimized by using neural networks

Parameters of objects are replaced with score functions of object features

$$\frac{f(\mathbf{x}_i)}{\sum_j f(\mathbf{x}_j)}$$

score for the next ranked object

sum of scores for the not yet ranked objects

scores functions, $f(\mathbf{x}_i)$, are linear, and these weights are estimated by maximum likelihood

Absolute / Relative Ranking

absolute ranking function

objects {A,B,C} → **absolute ranking function** → are sorted as: A > B > C

C is replaced with D {A,B,D} → **absolute ranking function** → A must be always ranked higher than B

In other words, either D>A>B, A>D>B, or A>B>D is allowed

relative ranking function

Other than absolute ranking function

- If you know Arrow's impossibility theorem, this is related to its condition I

Absolute / Relative Ranking

regression order model

sorting by scores ↓ absolute ranking function	linear ordering ↓ relative ranking function

- For IR or recommendation tasks, absolute ranking functions should be learned. For example, the fact that an apple is preferred to an orange is independent from the existence of a banana.
- Only few tasks suited for relative ranking

Relevance Feedbacks

[Joachims 02, Radlinski+ 05]

Leaning from relevance feedback is a typical absolute ranking task

Ranked List for the query Q

1: document A

2: document B

3: document C

4: document D

5: document E

selected by user

The user scans this list from the top, and selected the third document C.

The user checked the documents A and B, but these are not selected.

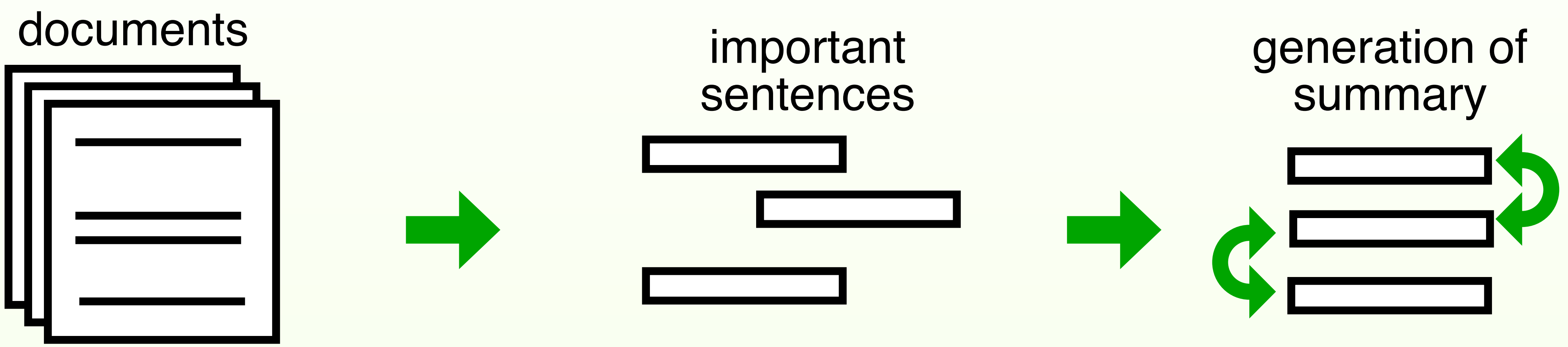
This user's behavior implies relevance feedbacks: C>A and C>B.

Object ranking methods can be used to update document's relevance based on these feedbacks

Multi-Document Summarization

[Bollegala+ 05]

Example of relative ranking task: Multi-Document Summarization (MDS)



Generating summaries is sorting sentences appropriately

From the samples of correctly sorted sentences, object ranking methods learns ranking functions

- features of sentences: chronological info, precedence, relevance among sentences

Appropriate order of sentences are influenced by the relevance to the other sentences or the importance relative to the other sentences

Absolute ranking functions are not appropriate for this task

Attribute and Order Noise

Order Noise

noiseless order → observed sample

$$\mathrm{A} \succ \mathrm{B} \succ \mathrm{C} \;\rightarrow\; \mathrm{A} \succ \mathrm{C} \succ \mathrm{B}$$

order noise is the permutation in orders

Attribute Noise

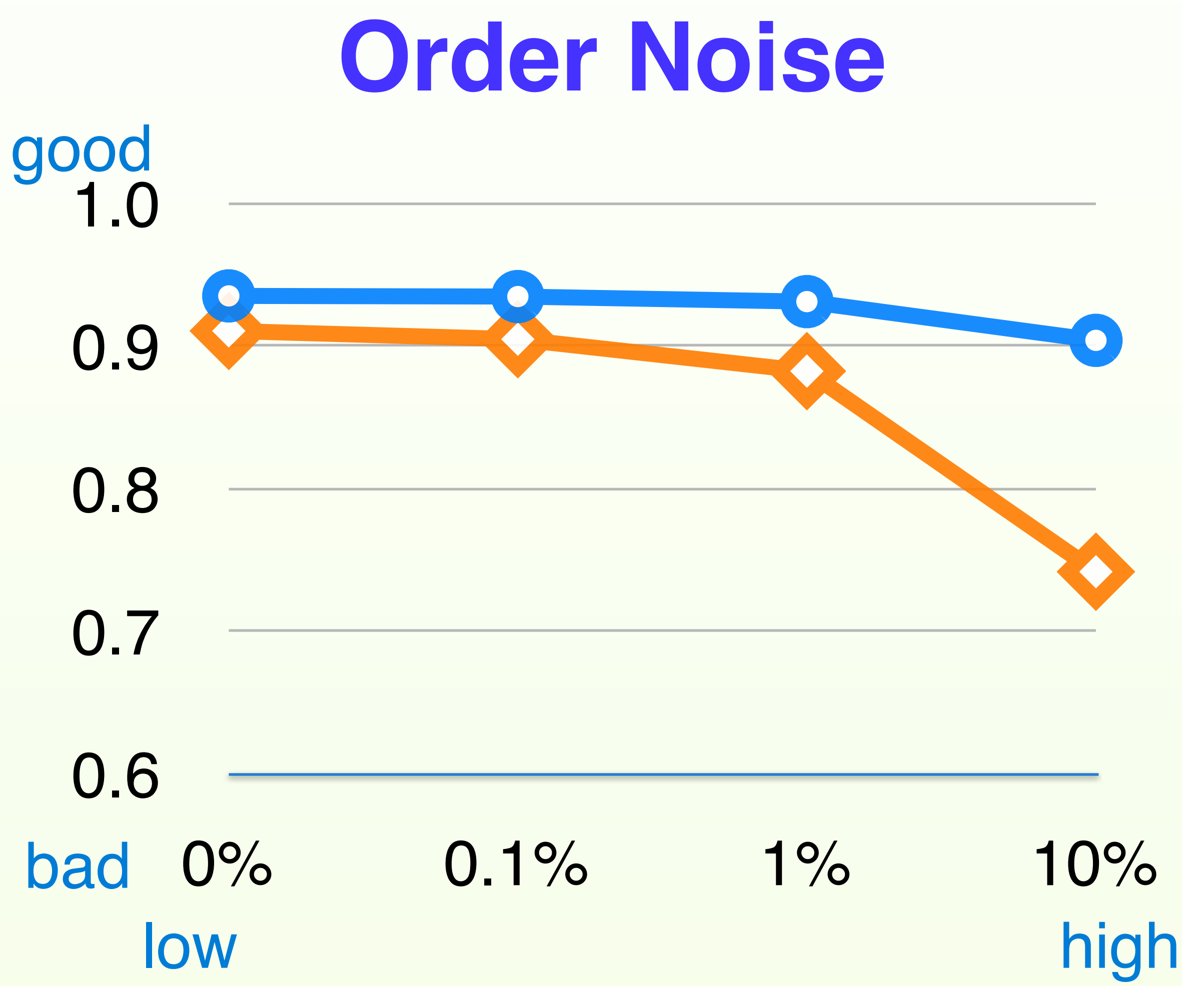
objects are represented by attribute vectors

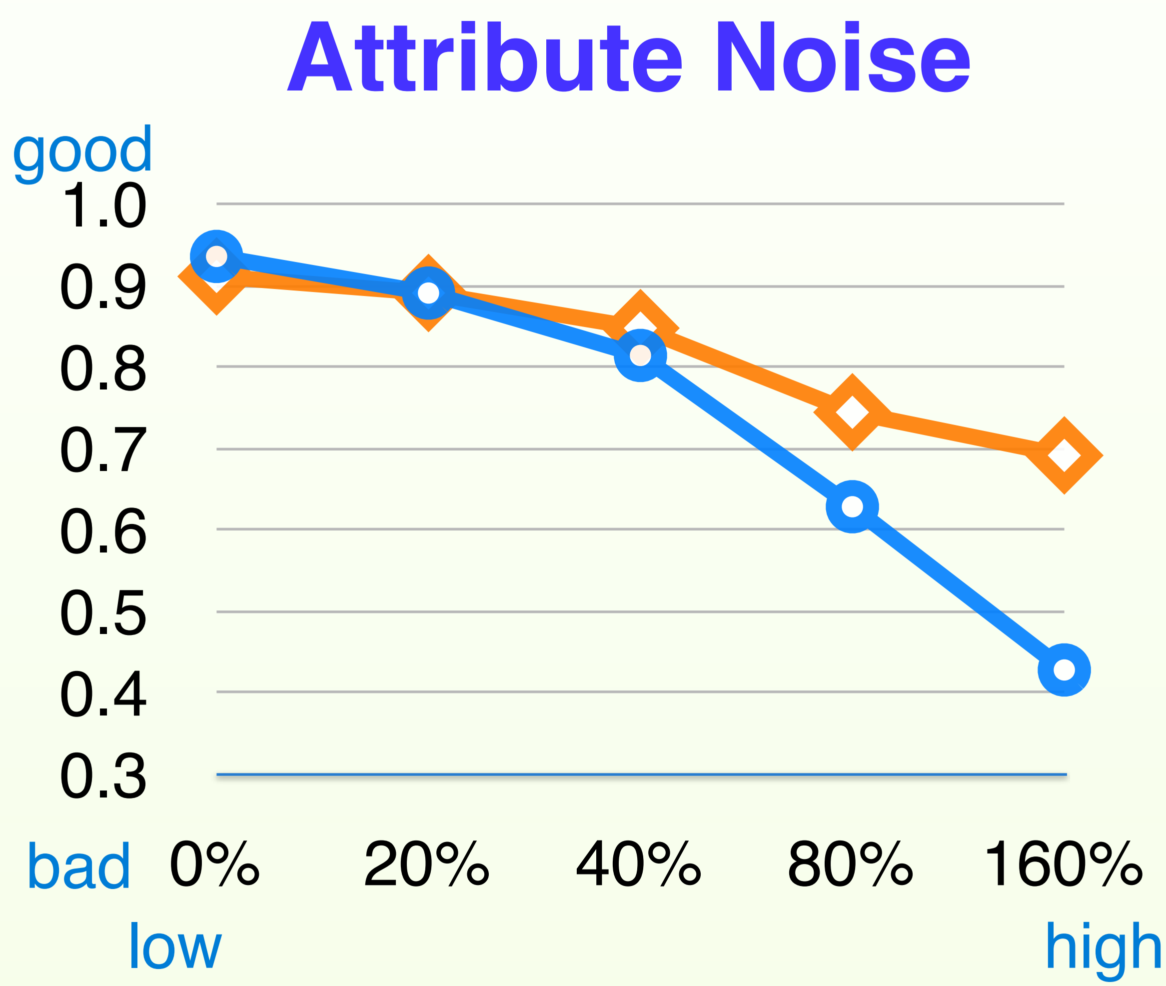
$$x_i = (x_{i1}, \ldots, x_{ik})$$

attribute noise is the perturbation in attribute values

Robustness against Noises

[Kamishima+ 05]





ERR SVOR

Vertical: prediction concordance Horizontal: noise level

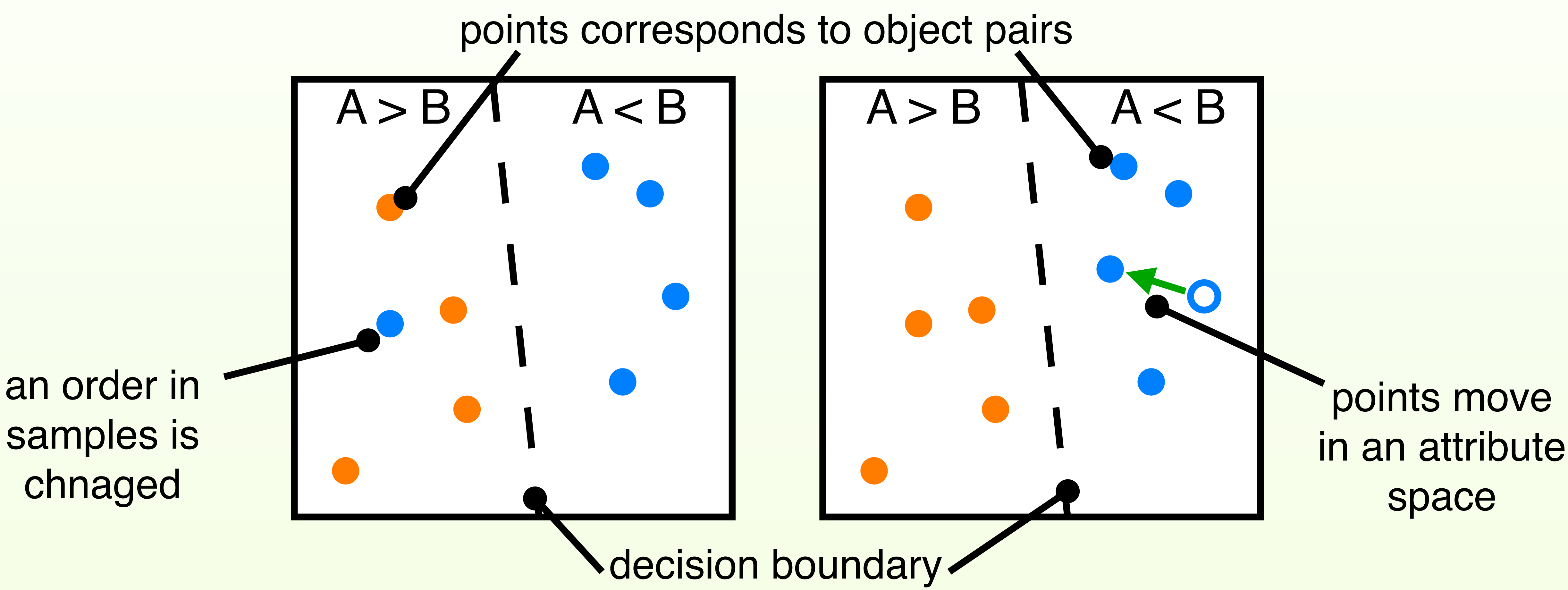
SVM-based ➡ robust against attribute noise

non-SVM-based ➡ robust against order noise

SVM-based Cases

SVM-based methods solves object ranking tasks as classification: A>B or A<B

Order Noise | **Attribute Noise**

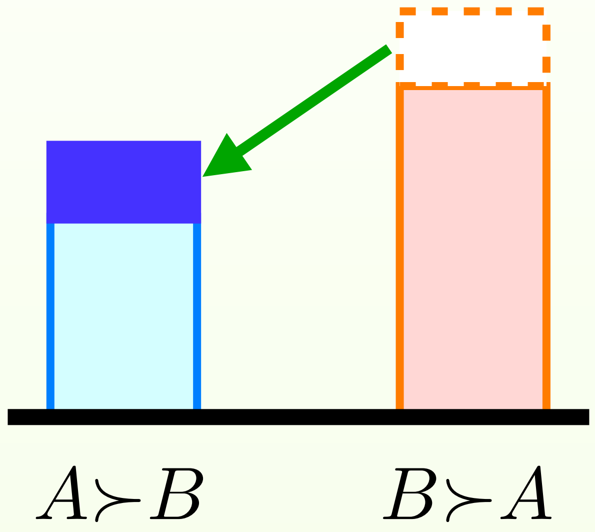


Changed points become support-vectors with high probability, and the results are seriously influened

Slight change in features never influences the results, if changing within decision boundary

non-SVM-based Cases

Order Noise



samples are moved from B>A to A>B

Results are not influenced, if majority class between these two do not change

Attribute Noise

Any little changes in features influences the loss function, due to the lack of the robustness features like hinge loss of SVMs

Performance of Object Ranking Methods

Accuracy

We compared the prediction accuracies of object ranking methods except for ListNet [Kamishima+ 05]. Though several differences are observed, we think that, like other ML tasks, the appropriate choices for the target task is primally important.

Efficiency

Two SVMs are slow than non-SVMs, and our ERR is fast in almose cases

Powerful linear model

Linear models for ranking functions are more powerful than in standard regression or classification. This is because any monotonic functions are equivalent to linear function as ranking score function.

Conclusion

- define object ranking task and discuss relation with regression and ordinal regression problems
- introduce four types of distributions for rankings: Thurstonian, paired comparison, distance-based, and multistage
- show six methods for object ranking tasks: Cohen's method, RankBoost, SVOR(=RankingSVM), OrderSVM, ERR, and ListNet
- propose the notion of absolute and relative ranking tasks
- discuss about the prediction accuracy of object ranking methods

SUSHI data: preference in sushi surveyed by ranking method
http://www.kamishima.net/sushi/

Bibliography

- [Agresti 96] A.Agresti, "Categorical Data Analysis", John Wiley & Sons, 2nd eds. (1996)
- [Arnold+ 92] B.C.Arnold et. al. "A First Course in Order Statistics", John Wiley & Sons, Inc. (1992)
- [Babington Smith 50] B.Babington Smith, "Discussion on Professor Ross's Paper", JRSS (B), vol.12 (1950)
- [Bell+ 07] R.M.Bell & Y.Koren, "Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights", ICDM2007
- [Bladley+ 52] R.A.Bradley & M.E.Terry, "Rank Analysis of Incomplete Block Designs — I. The Method of Paired Comparisons", Biometrika, vol.39 (1952)
- [Bollegala+ 05] D.Bollegala et. al. "A Machine learning Approach to Sentence Ordering for Multidocument Summarization and its Evaluation", IJCNLP2005

Bibliography

- [Cao+ 07] Z.Cao et. al. "Learning to Rank: From Pairwise Approach to Listwise Approach" ICML2007
- [Cohen+ 99] W.W.Cohen et. al "Learning to Order Things", JAIR, vol. 10 (1999)[Cosley+ 03] D.Cosley et. al. "Is Seeing Believing? How Recommender Interfaces Affect Users' Opnions", SIGCHI 2003
- [Critchlow+ 91] D.E.Critchlow et. al. "Probability Models on Rankings", J. of Math. Psychology, vol.35 (1991)
- [Freund+ 03] Y.Freund et. al. "An Efficient Boosting Algorithm for Combining Preferences", JMLR, vol.4 (2003)
- [Grötschel+ 84] M.Grötschel et. al. "A Cutting Plane Algorithm for the Linear Ordering Problem", Operations Research, vol.32 (1984)
- [Herbrich+ 98] R.Herbrich et. al. "Learning Preference Relations for Information Retrieval", ICML1998 Workshop: Text Categorization and Machine Learning

Bibliography

- [Herlocker+ 99] J.L.Herlocker et. al. "An Algorithmic Framework for Performing Collaborative Filtering", SIGIR1999
- [Joachims 02] T.Joachims, "Optimizing Search Engines Using Clickthrough Data", KDD2002
- [Kamishima 03] T.Kamishima, "Nantonac Collaborative Filtering: Recommendation Based on Order Responses", KDD2003
- [Kamishima+ 05] T.Kamishima et. al. "Supervised Ordering — An Empirical Survey", ICDM2005
- [Kamishima+ 06] T.Kamishima et. al., "Nantonac Collaborative Filtering — Recommendation Based on Multiple Order Responses", DMSS Workshop 2006
- [Kazawa+ 05] H.Kazawa et. al. "Order SVM: a kernel method for order learning based on generalized order statistics", Systems and Computers in Japan, vol.36 (2005)

Bibliography

- [Mallows 57] C.L.Mallows, "Non-Null Ranking Models. I", Biometrika, vol.44 (1957)
- [Marden 95] J.I.Marden "Analyzing and Modeling Rank Data", Chapman & Hall (1995)
- [McCullagh 80] P.McCullagh, "Regression Models for Ordinal Data", JRSS(B), vol.42 (1980)
- [Thurstone 27] L.L.Thurstone "A Law of Comparative Judgment", Psychological Review, vol.34 (1927)
- [Plackett 75] R.L.Plackett, "The Analysis of Permutations", JRSS (C), vol.24 (1975)
- [Radlinski+ 05] F.Radlinski & T.Joachims, "Query Chains: Learning to Rank from Implicit Feedback", KDD2005