Supervised Ordering

An Empirical Survey

T.Kamishima[1], H.Kazawa[2], S.Akaho[1]

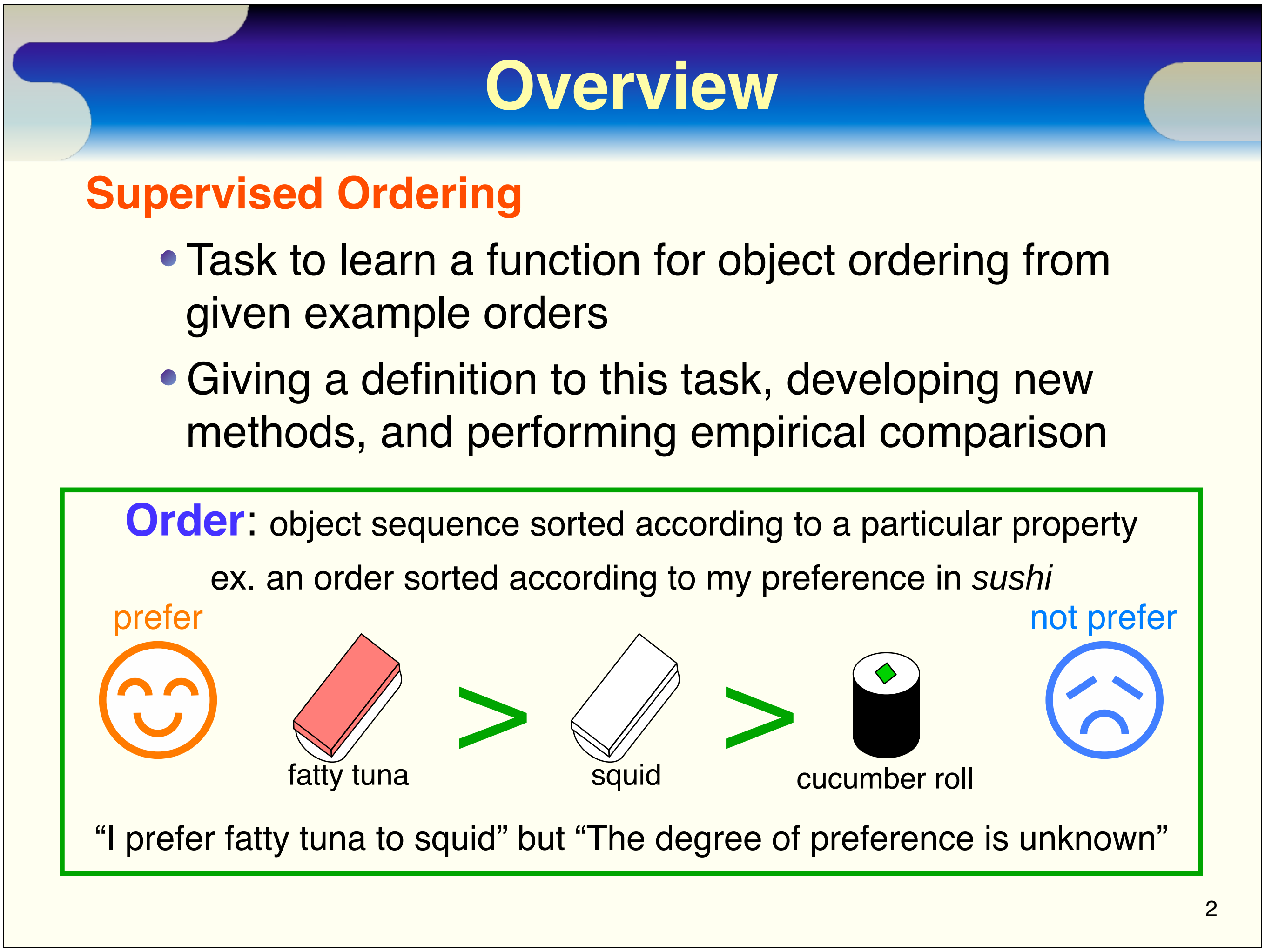
1) National Institute of AIST, Japan

2) NTT Communication Science Labs, Japan

5th ICDM, Houston, USA, 27-30/11/2005

START

Today, we would like to talk about a learning task targeting orders.



A supervised ordering task is to learn a function for object ordering from given example orders.

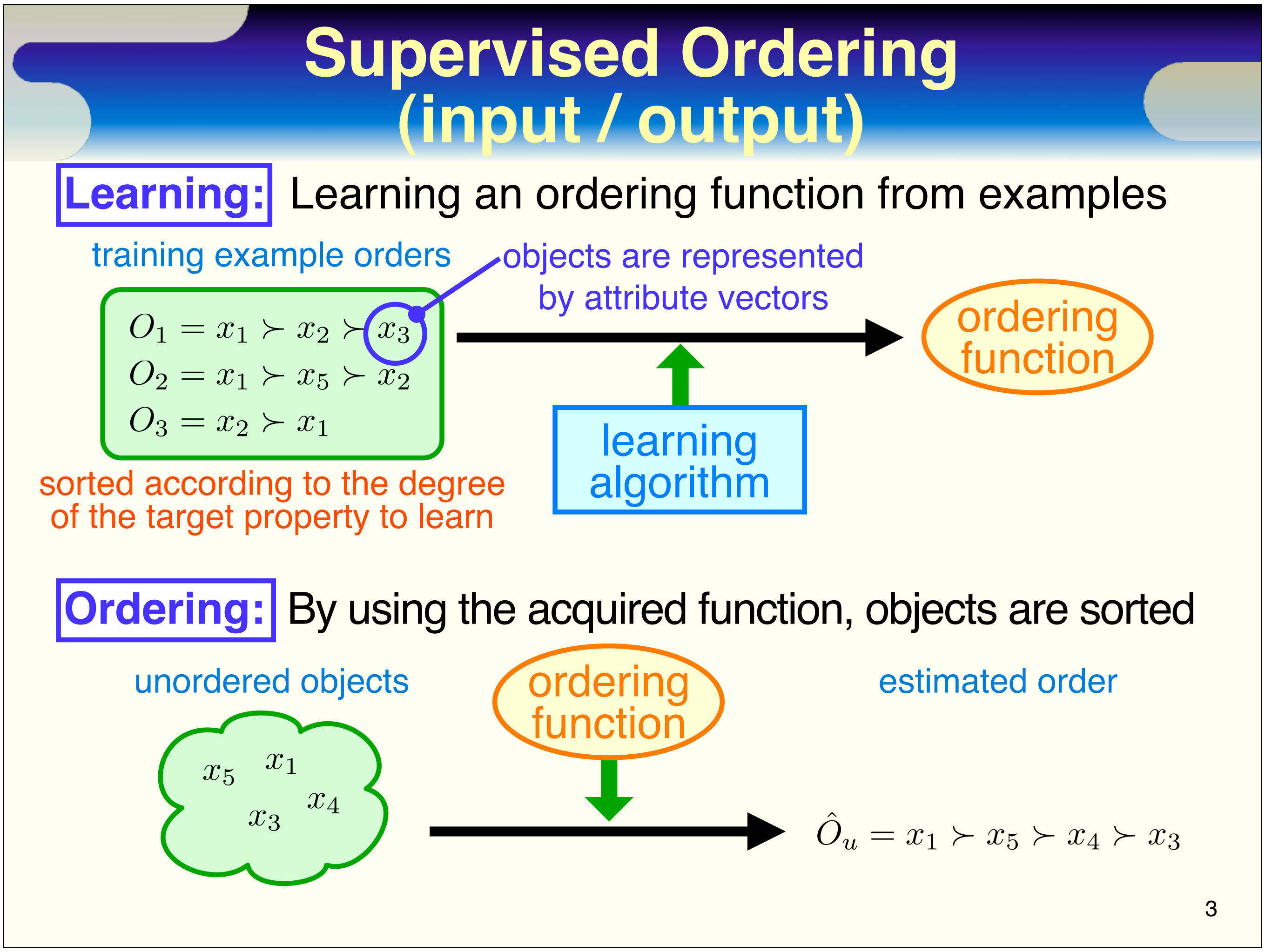
Then, we develop a new method, and perform empirical survey.

We begin with what is an order.

An order is an object sequence sorted according to a particular property.

For example, this is an order sorted according to my preference in sushi.

This order indicates that “I prefer a fatty tuna to squid”, but “The degree of preference is unknown.”



We first show an overview of a supervised ordering task.

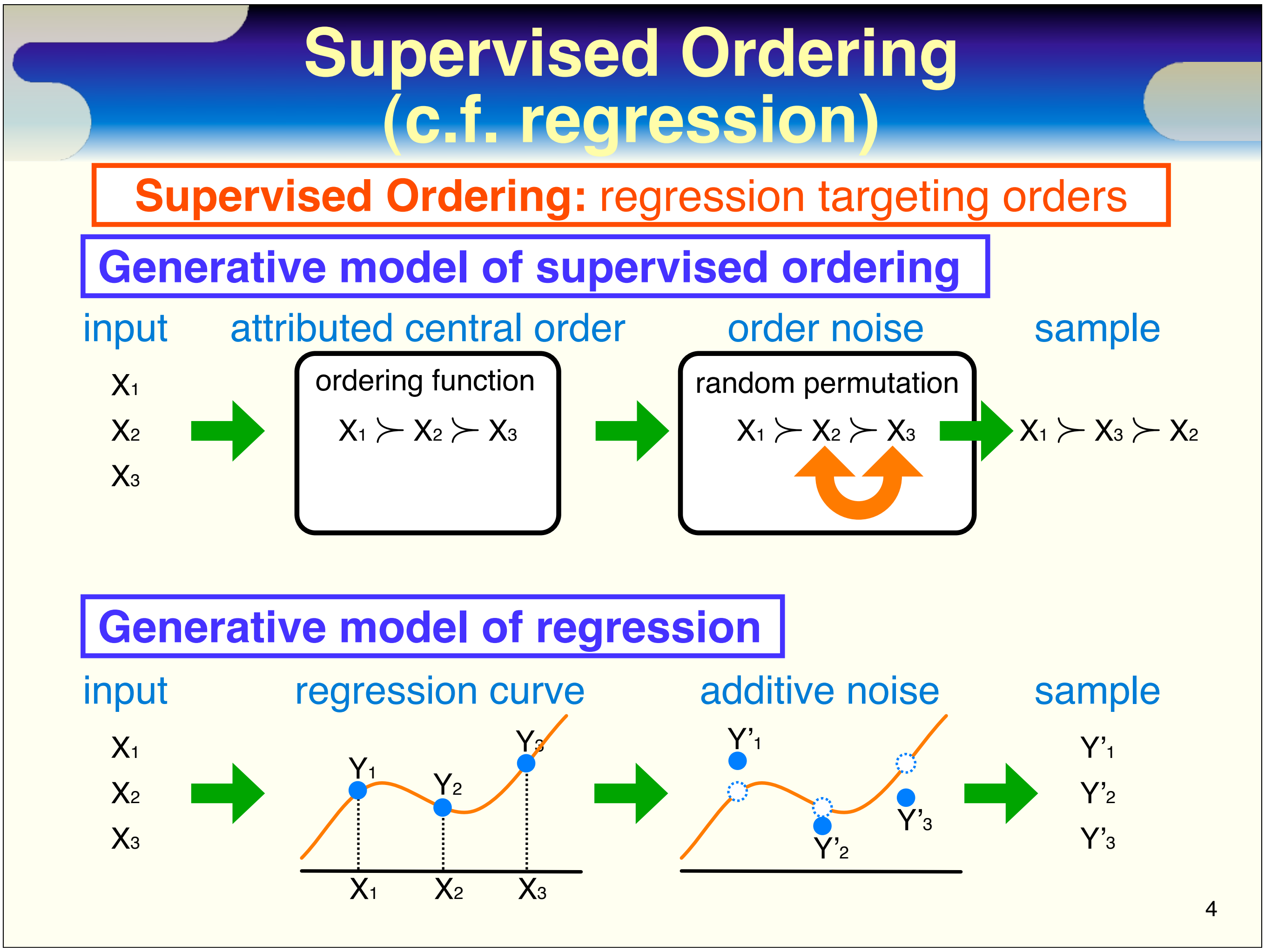
Training example orders are sorted according to the degree of the property to learn.

Objects in these orders are represented by attribute vectors.

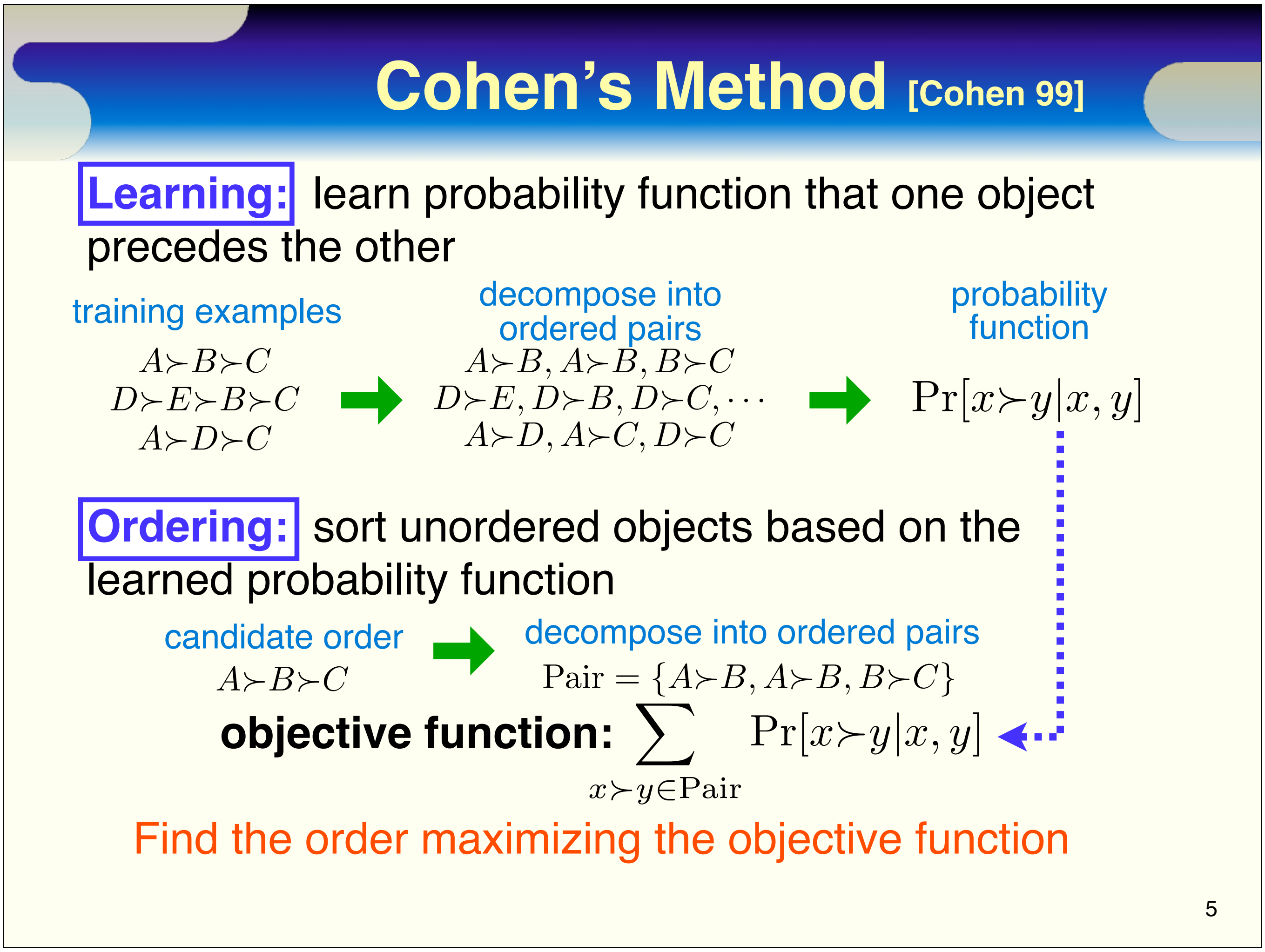
From these examples, the algorithm learn an ordering function.

By using this acquired ordering function, unordered objects are sorted according to the degree of the target property.

Furthermore, objects not appeared in training examples can be ordered based on the values of attribute vectors.



Supervised ordering can be considered as regression targeting orders.
This is a generative model of supervised ordering.
Unordered objects are given.
These objects are sorted according to the degree of the target property.
This order is then affected by order noise, and finally a sample order is generated.
This model is very similar to that of regression, like this.
This supervised ordering task is also related to ordinal regression or central orders.
Now, we have defined a supervised ordering task.
We next show five methods for this task.



Cohen's method adopts a paired comparison approach.
Training examples are first decomposed into ordered pairs.
From these pairs, the algorithm learns a probability function that one object precedes the other.
When sorting unordered objects, the algorithm tries to find the order maximizing this objective function among candidate orders.
The objective function is composed of learned probability functions.

RankBoost [Freund 98]

Learning: find a linear combination of weak hypotheses by using boosting approach

objects

weak hypotheses

partial information regarding 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: $\sum_{t=1}^{T} \alpha_t h_t(x)$

Weights and weak hypotheses are chosen so that scores are concordant with example orders

Ordering: sort unordered objects according to their scores

The algorithm RankBoost tries to find a score function, which is a linear combination of weak hypotheses.
Given an object, weak hypotheses provides some partial information regarding the target order to learn.
Weights and weak hypotheses are chosen so that scores are concordant with example orders by using a boosting technique.
Once this score function is learned, unordered objects can be sorted according to their scores.

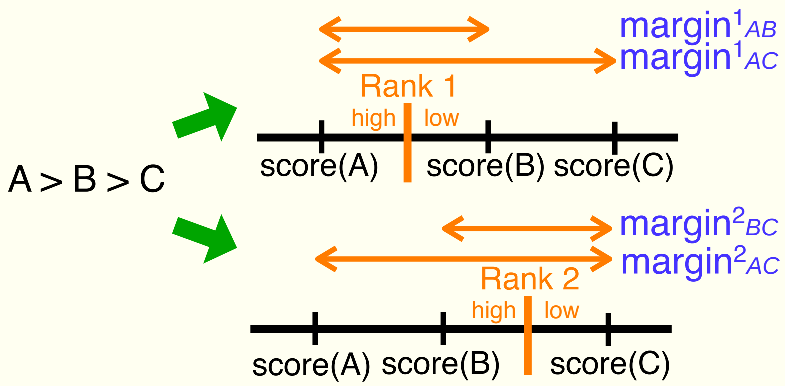
SVM-Based Method (Order SVM)

Learning: find a score function which maximally separate higher-ranked from lower-ranked on average

Order SVM [Kazawa 03]

Rank as category & SVM-like formulation & Kernel ready

sample orders — score & margin — Objective



maximize:

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

Ordering: sort unordered objects according to their scores

7

I developed SVM-based algorithm, Order SVM for supervised ordering.

In learning stage, Order SVM finds near-parallel hyperplanes in the attribute vector space, which separate higher ranked objects from lower ranked ones.

Then the objects are ordered along the direction perpendicular to the hyperplanes at prediction.

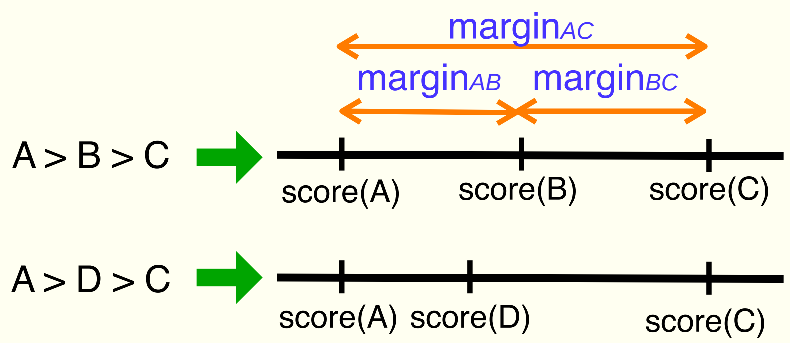
SVM-Based Method (SVOR)

Learning: find a score function that maximally separates preferred from non-preferred

SVOR (Support Vector Ordinal Regression) [Herbrich 98]

SVM-like formulation & Kernel ready

sample orders | score & margin | Objective

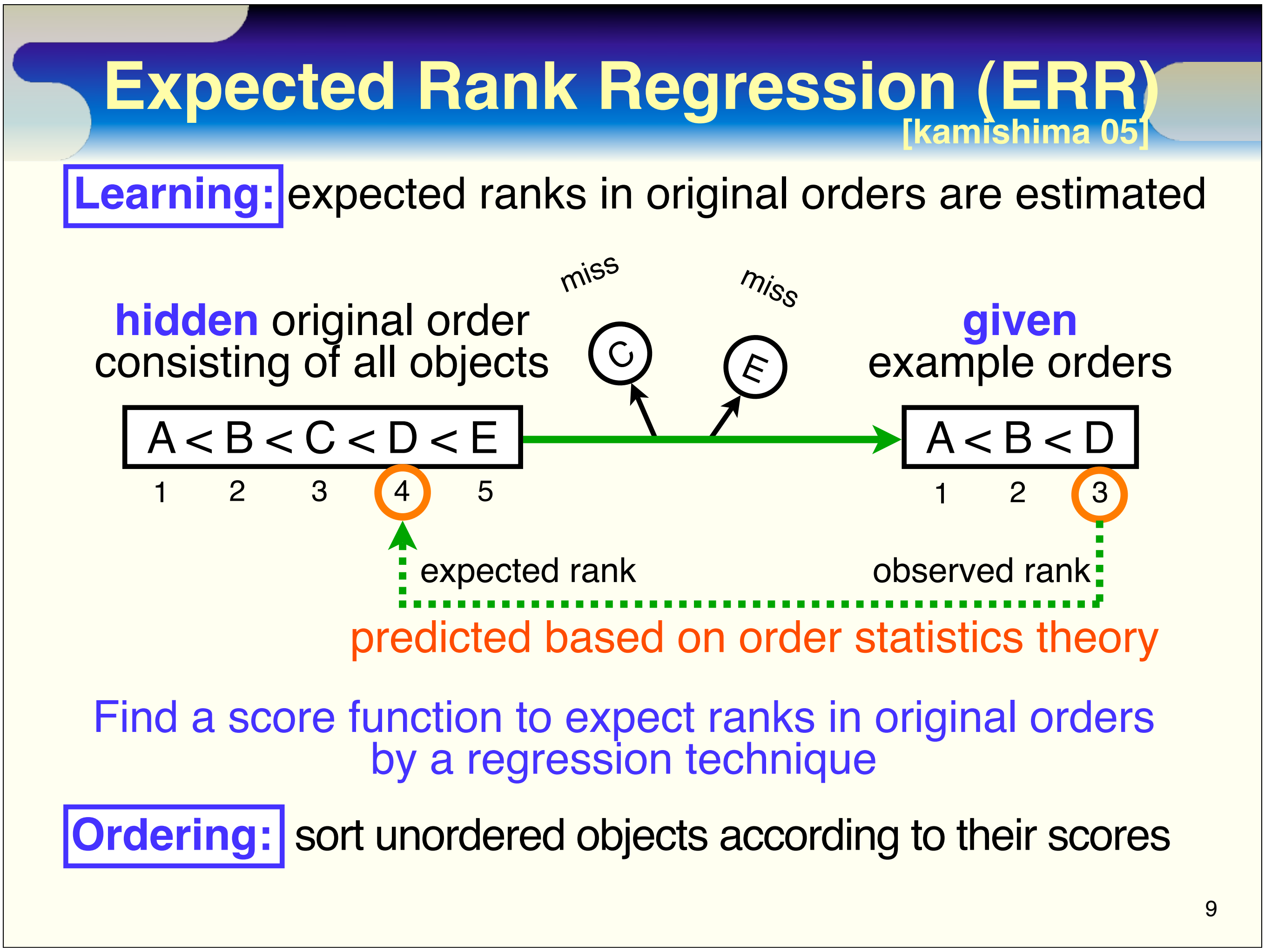


maximize:

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

Ordering: sort unordered objects according to their scores

Another SVM-based method was proposed by Herbrich.
We call the method Support Vector Ordinal Regression here.
In learning stage, SVOR finds an optimal direction so that along the direction the margins between objects should be large.
SVOR is different from Order SVM at the point that SVOR only considers the margins within a sample order, whereas Order SVM considers the margins over all the sample orders.



In the expected rank regression method, a score function is learned by using standard regression.

These scores are proportional to the expected ranks in hidden original order consisting of all objects.

These expected ranks are predicted from observed ranks in given example orders based on order statistics theory.

Now, we have shown five supervised ordering methods.

Next, we will pick up one interesting result of our empirical comparison.

Order Noise / Attribute Noise

Order Noise

ordering function → noiseless order $A \succ B \succ C$ → observed sample $A \succ C \succ B$

order noise is the permutation in orders

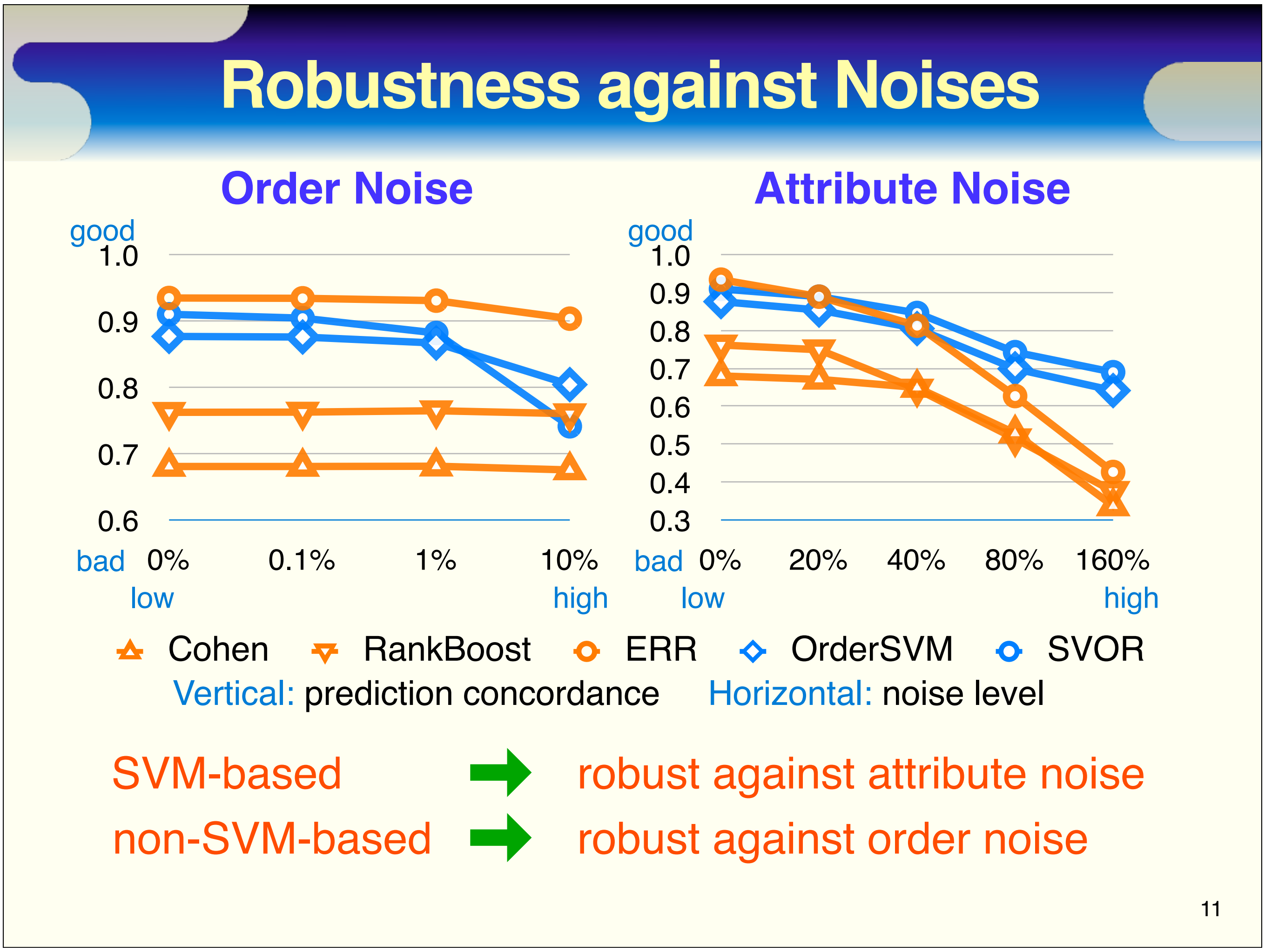
Attribute Noise

objects are represented by attribute vectors

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

attribute noise is the perturbation in attribute values

We investigated robustness of five methods against two types of noise.
One type is order noise,which is the permutation in orders.
The other type is attribute noise, which is the perturbation in attribute values.



These figures show the variation of prediction concordance in accordance with the noise level.
Two blue curves are SVM-based results, and the others are non-SVM-based results.
These two sets of curves are clearly contrasted.
SVM-based methods are robust against attribute noise, but are not robust against order noise.
On the other hand, non-SVM-based methods conversely behaves.
Reasons for these phenomena are in our article.

Conclusion

- define a supervised ordering task
- give a unified view to independently proposed tasks
- show relations to existing problem regarding orders, such as central orders, probability models for rankings, and ordinal regression
- develop new method
- empirically show the variation of algorithms' performance in various factors

Detailed experimental results and more Information

`http://www.kamishima.net/`

We would like to conclude our talk.

Our contributions are as follows.

More information is available in our Web site.

That's all we have to say. Thank you.