Efficient Clustering for Orders

Toshihiro Kamishima and Shotaro Akaho
http://www.kamishima.net/
National Institute of Advanced Industrial Science and Technology (AIST), Japan
The 2nd Workshop on Mining Complex Data @ ICDM2006
Hong Kong, China, 18/12/2006

We would like to talk about an efficient method for clustering orders.

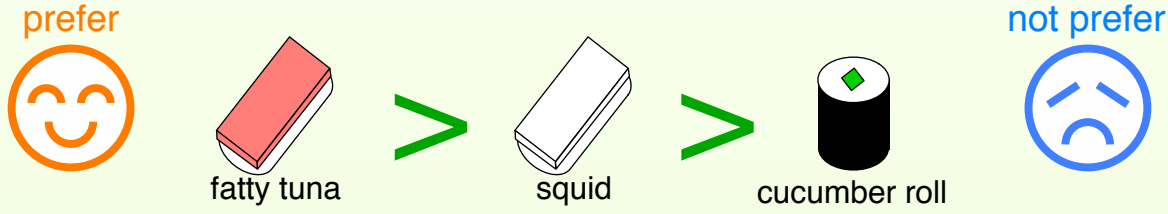
Overview

k-o'means: algorithm for clustering orders

- Order is a basic type of data structure, and is useful to measure subjective quantities
- We propose a technique to improve the efficiency of a *k*-o'means method

Order: object sequence sorted according to a particular property

ex. an order sorted according to my preference in *sushi*



"I prefer fatty tuna to squid" but "The degree of preference is unknown"

Complex data is a collection of primitive data, and these primitives are structured.
Order is a basic type of data structure, and is useful to measure subjective quantities
To cluster such orders, we developed a k-o'means algorithm.
Today, we propose a technique to improve the efficiency of this k-o'means method

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."

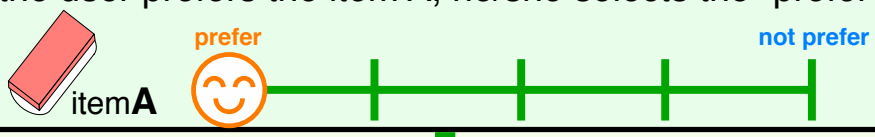
Application: measuring subjective quantities

Order is useful for measuring subjective quantities, such as, the degrees of preference, impression, or sensory

Semantic Differential Method (SD Method)

measured by pointing on a scale whose extremes are represented by antonymous words

Ex: If the user prefers the item A, he/she selects the “prefer”



Ranking Method

Objects are sorted according to the degree of quantities to be measured

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

3

We will show an example task suited for using orders.

Orders are useful for measuring subjective quantities, such as the degrees of preference, impression, or sensory.

Such quantities can be measured by pointing on a scale like this.

For example, If he/she prefers the item A, the user select “prefer”

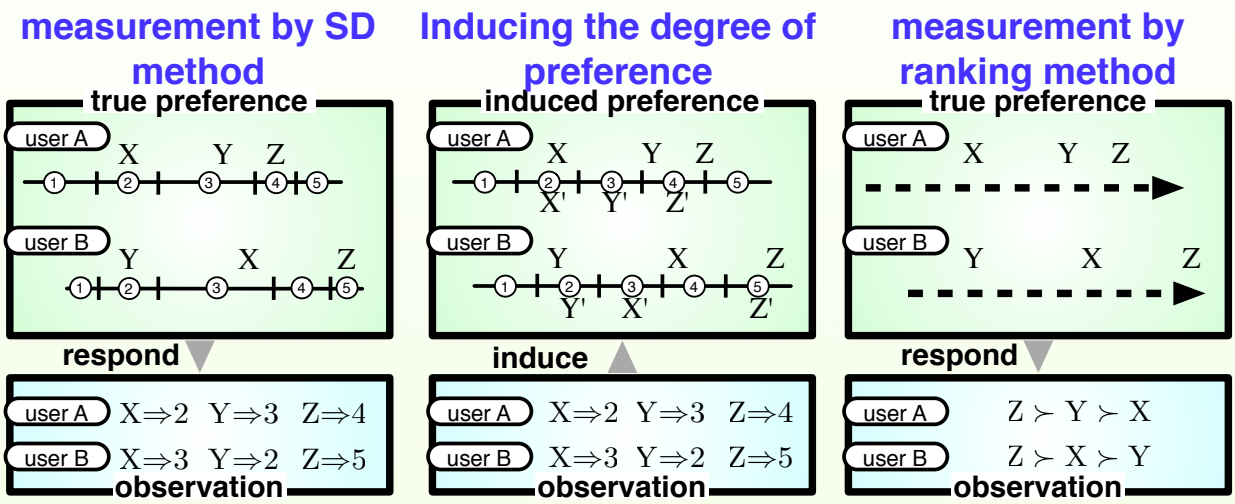
This is called an SD method.

One alternative is a ranking method.

Objects are sorted according to the degree of quantities to be measured.

In this example, the user prefer Item A most, and the item B least.

Application: measuring subjective quantities



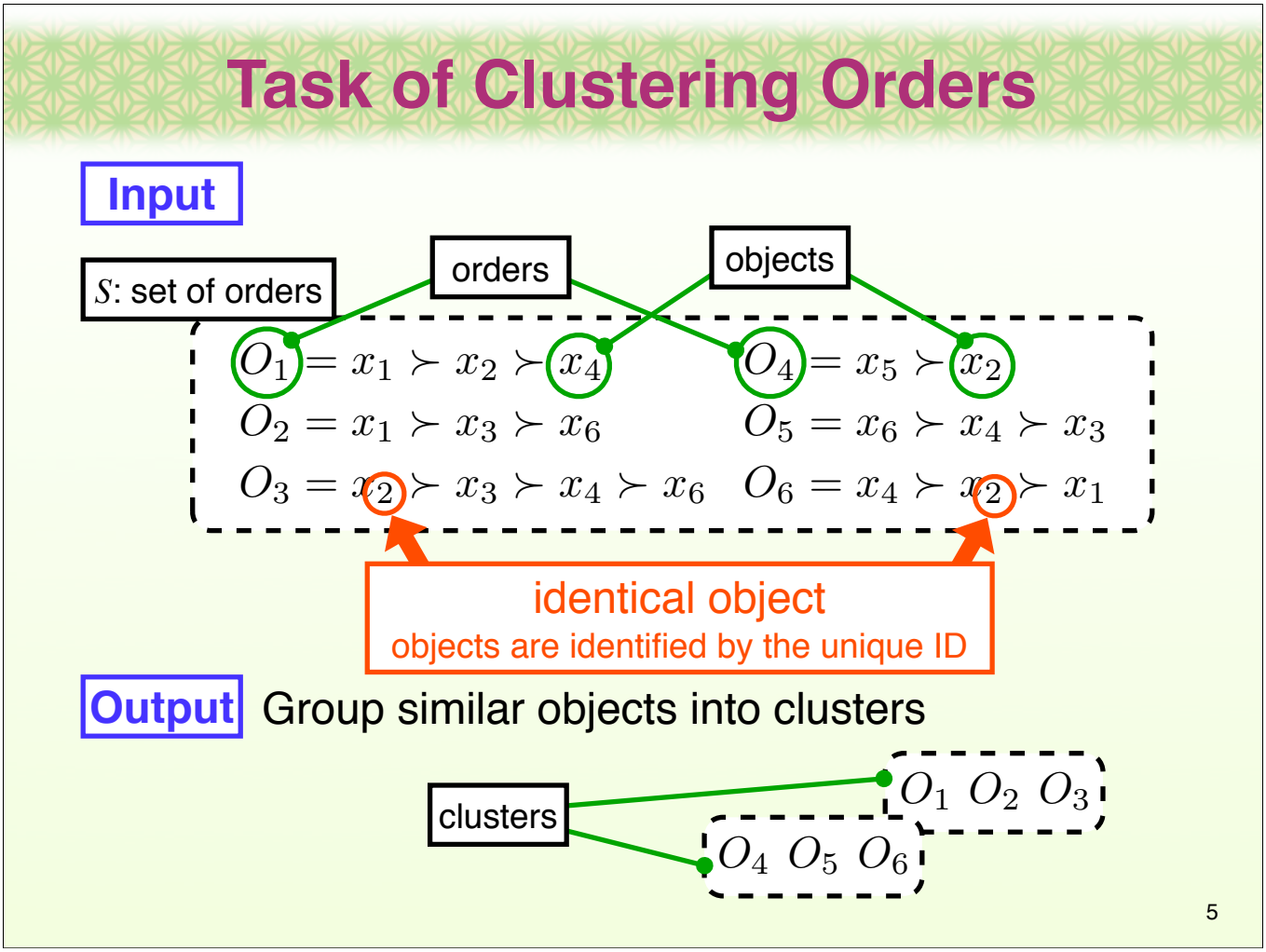
- Each user uses his/her own mapping scale
- Observed scores cannot be comparable among users

- To make the observations comparable, we are forced to assume a common mapping scale.
- The degrees of quantities might be deviated

- In a ranking method, the degrees of preferences are relatively specified
- No need for calibration of mapping scales

4

We show a merit of using a ranking method.
In an SD method, each user uses his/her own mapping scale.
So, observed scores cannot be comparable among users.
Therefore, we are forced to assume a common mapping scale.
However, the degrees of quantities might be deviated to X to X'.
In a ranking method, the degrees of preferences are relatively specified.
So, no need for calibration of mapping scales.



Next, we formalize a task of clustering orders.

The goal of clustering is to partition a set of sample orders into clusters. Clusters consist of similar orders.

Orders are sorted sequences of objects.

Objects are identified by the unique ID. For example, the object x2 in orders, O3 and O6, is identical.

Here, we want to insist that sample orders are generally incomplete.

That is to say, orders might consist of different sets of objects.

This makes it difficult to cluster orders.

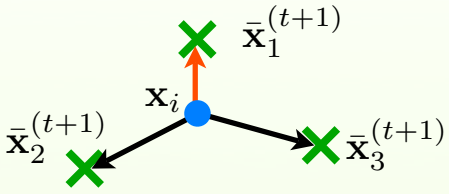
k-o'means algorithm

k-means algorithm

① update of centers ② reassignment of objects

$$\bar{\mathbf{x}}_k^{(t+1)} = \frac{1}{|C_k^{(t)}|} \sum_{\mathbf{x}_i \in C_k^{(t)}} \mathbf{x_i}$$

Iterate



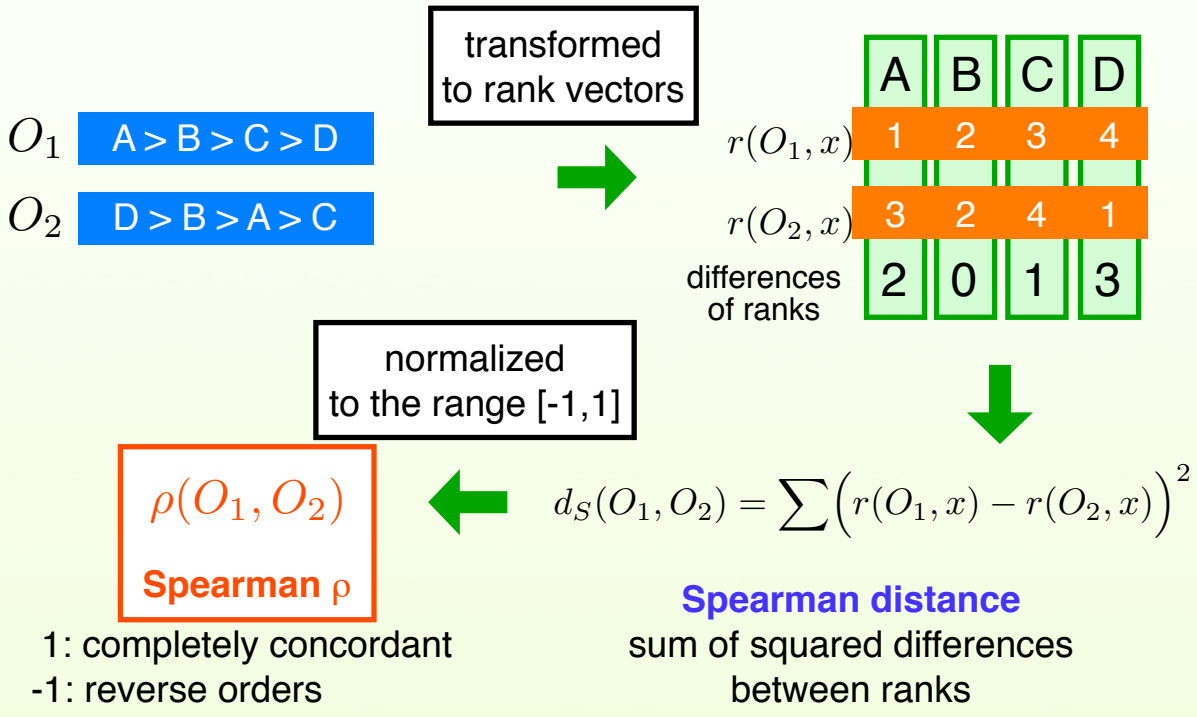
k-o'means algorithm

	k-means		*k*-o'means
objects:	vector	➡	order
similarity:	squared Eucledian	➡	Spearman ρ
center:	simple mean	➡	central order

6

We then show the algorithm for clustering orders, k-o'means.
This algorithm is the modified version of a k-means algorithm.
In the k-means, these two steps are iteratively applied: update of centers and reassignment of objects.
To modify this k-means so as to fit for orders, we introduced similarities and centers for orders.
As similarities, we use Spearman rho, and as centers, we use a central order.
We then show these notions.

Similarity Measures for Orders



7

We first show Spearman rho.

This measure is defined between two orders that consist of the same set of objects.

First, orders are transformed to rank vectors. For example, the rank of the object A in the order O1 is 1, and in the O2 is 3.

Second, we calculate Spearman distance, that is the sum of squared differences between ranks.

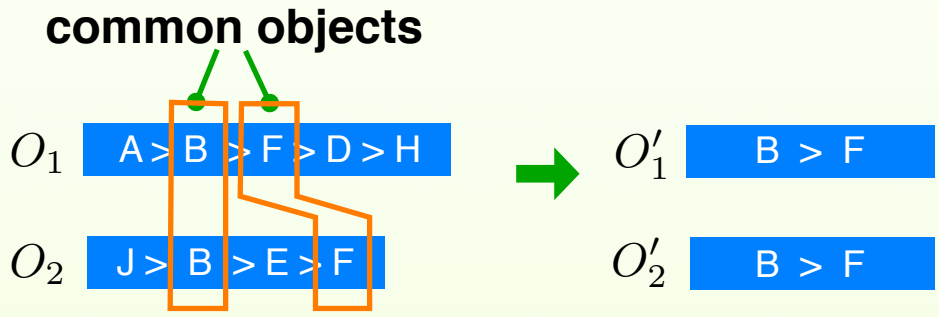
Finally, the value is normalized, and Spearman rho is obtained.

Similarity Measures for Orders

sample orders are generally incomplete

➡ orders may consist of different sets of objects

Spearman ρ is calculated over common objects



Convert similarity to dissimilarity

$$d_\rho(O_1, O_2) = 1 - \rho(O_1, O_2)$$

8

However, sample orders are generally incomplete, that is orders may consist of different sets of objects.

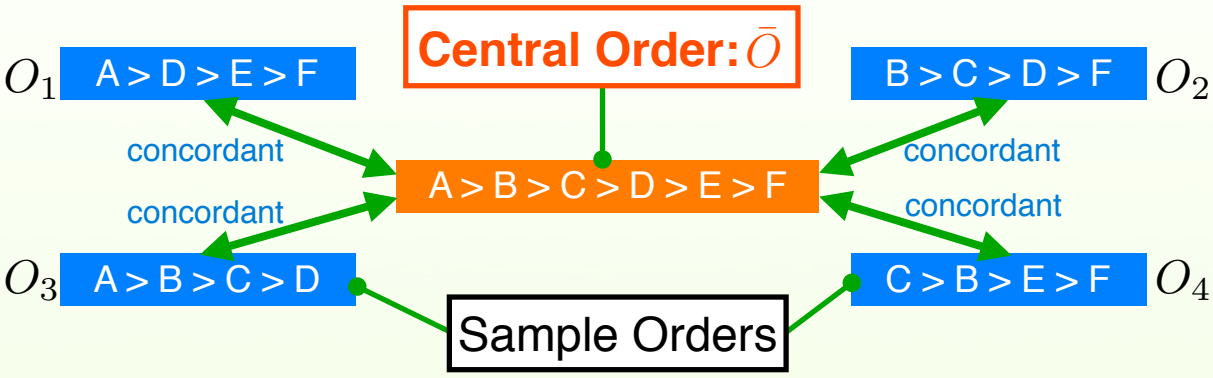
Therefore, Spearman rho is calculated over common objects.

For example, only objects B and F are considered, and the other objects are ignored.

Finally, in clustering, dissimilarity is used rather than similarity.

We defined dissimilarity like this.

Central Orders



Central Order $\bar{O} = \arg\min \sum_{O_i \in S} d_\rho(O, O_i)$

- concordant with sample orders on average
- consist of all objects in sample orders

9

Next, we show a notion of central orders.

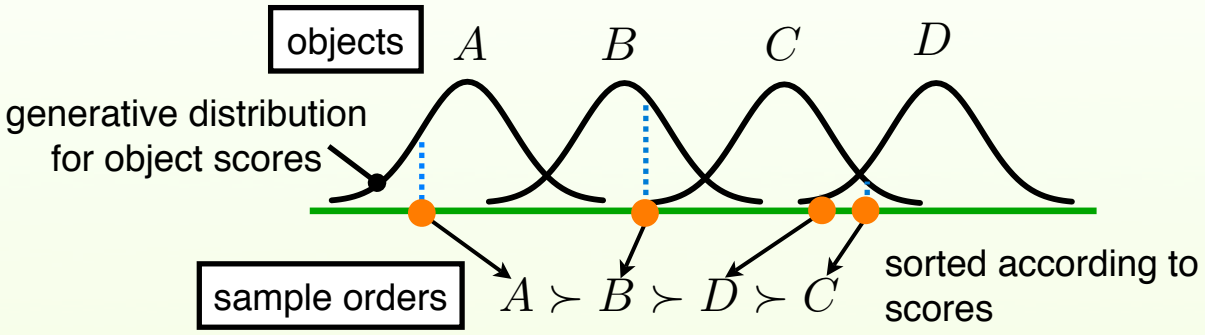
Intuitively speaking, a central order is concordant with sample orders on average, and consists of all objects in all sample orders.

That is to say, the order that minimizes the sum of dissimilarities to sample orders.

Because deriving central orders is a generally NP-hard problem, we used an approximation.

Thurstone Minimum Square Error

Thurstone's law of comparative judgment: [Thurstone 27]
probabilistic generative model of orders



mean parameters of score distributions can be estimated by

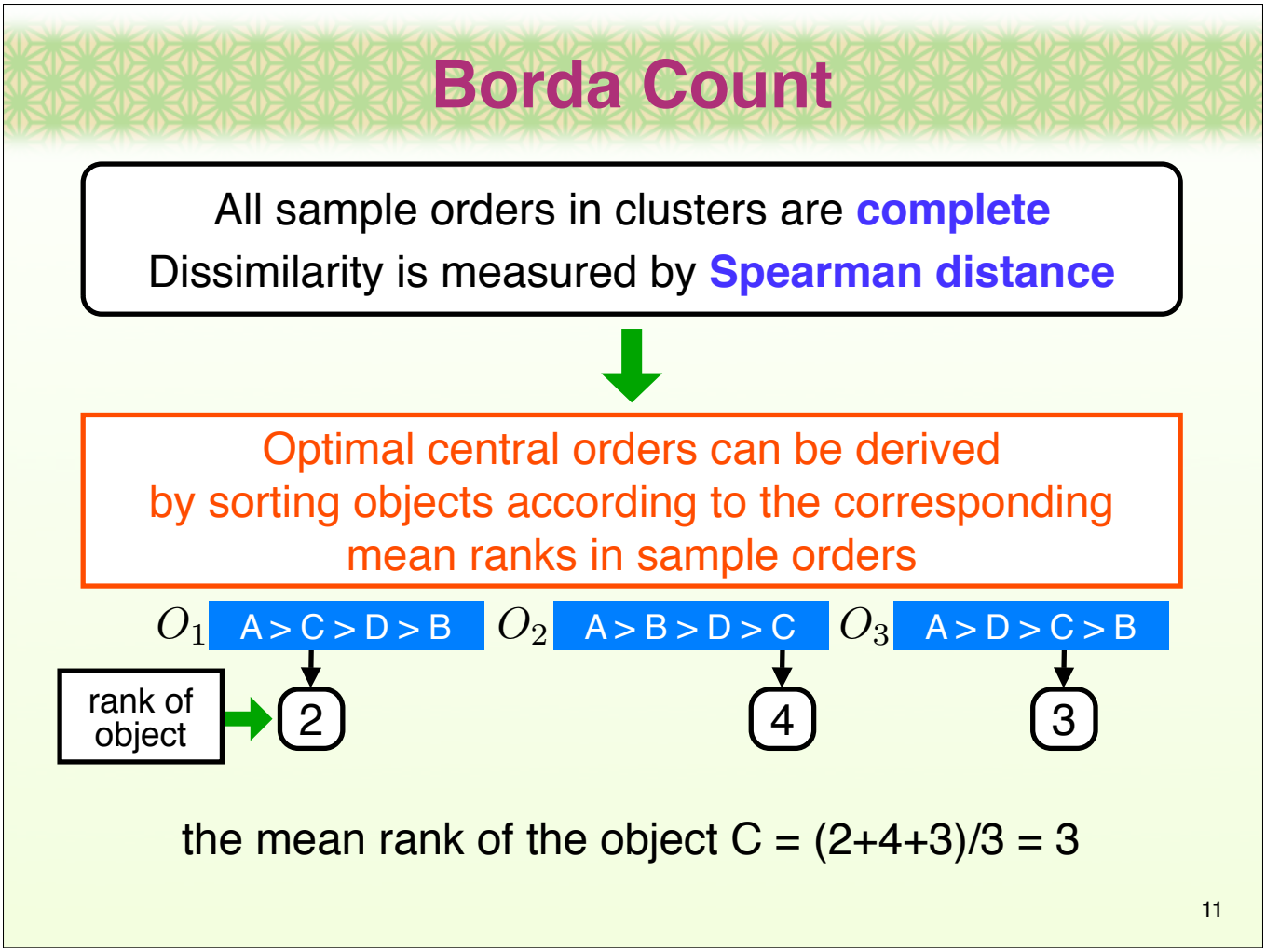
[Mosteller 51]

$$\mu_i = \sum_{x \in X_C} \Phi^{-1}\Big(\Pr[x_i \succ x]\Big)$$

sorting objects according to the corresponding mean parameters

10

In our previous work, we adopted a method based on Thurstone's law of comparative judgement.
This is a probabilistic generative model of orders.
For each object, score of the object follow its own normal distribution.
Sample orders are generated by sorting objects according to these scores.
Under some condition, mean parameters of score distributions can be analytically solved like this.
Central orders are derived by sorting objects according to the corresponding mean parameters.



To obtain central orders more efficiently, we propose a new method based on a Borda count method.

Under the condition that all sample orders in clusters are complete, and that dissimilarity is measured by Spearman Distance, the optimal central orders can be derived efficiently.

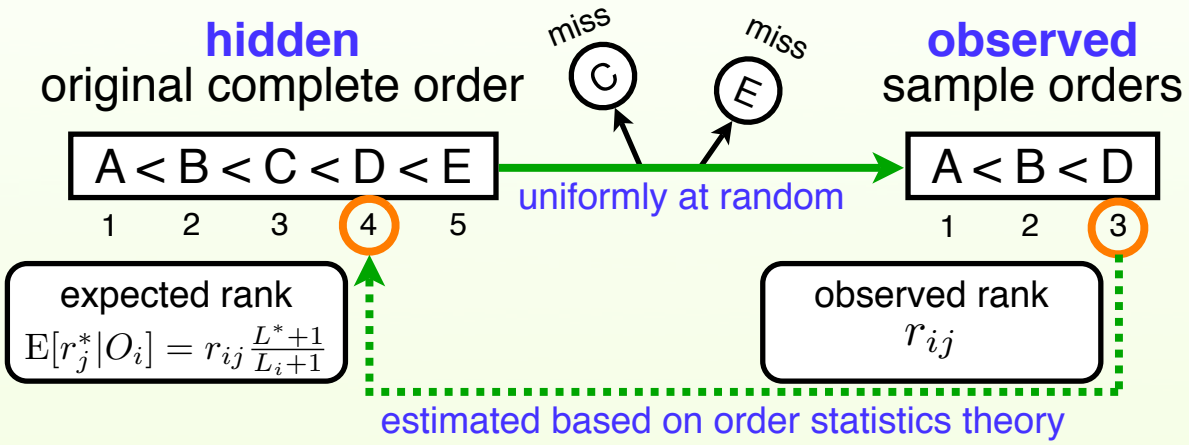
This method is equivalent to sorting according to the corresponding mean ranks in sample orders.

For example, ranks of the object C in these orders are 2, 4, and 3, respectively.

So, the mean rank of the object C is 3.

Expected Borda Count

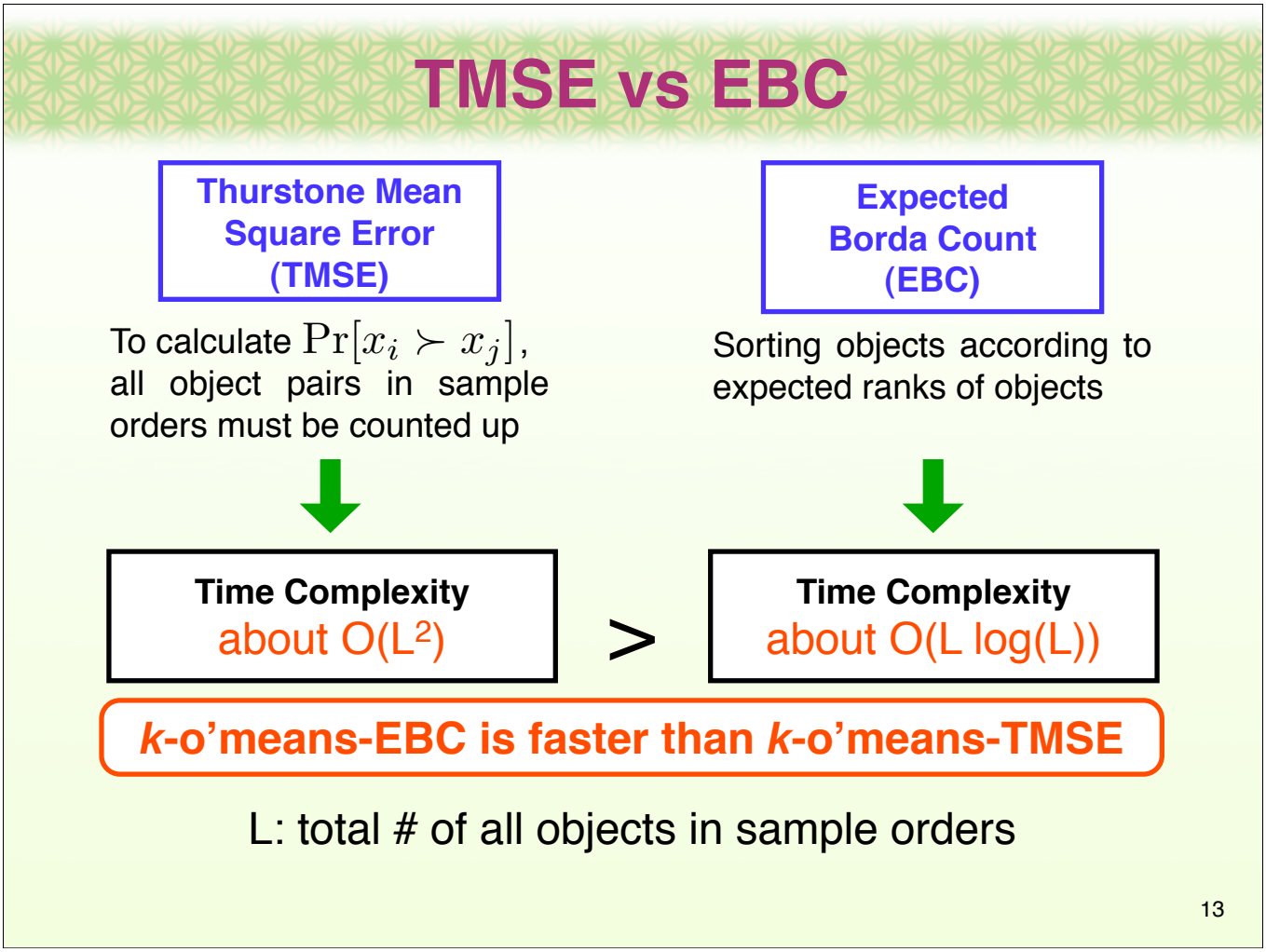
But, sample orders are generally incomplete...



Approximated central orders are derived by sorting according to the corresponding EXPECTED mean ranks

12

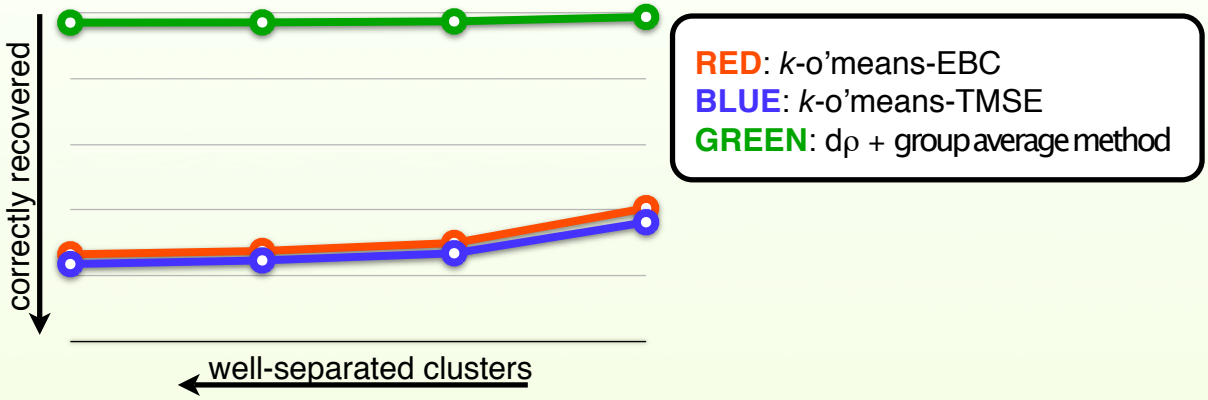
But, sample orders are generally incomplete. We then use the expected ranks.
Expected ranks can be calculated by very simple formula if this generative model is assumed.
We assume hidden original complete orders. Then, objects are missed uniformly at random. Finally, sample orders are observed.
Under this assumption, expected rank in this hidden orders can be estimated based on order statistics theory.
Approximated central order are derived by sorting according to the corresponding expected mean ranks.



The time complexity of our old TMSE is about the square of L, where L is the total # of all objects in sample orders.
In our new method, this is reduced to L log(L).
Consequently, our new k-o'means-EBC is faster.
Now, we have shown our new k-o'means method.
Next, we show experimental results.

Result on Artificial Data

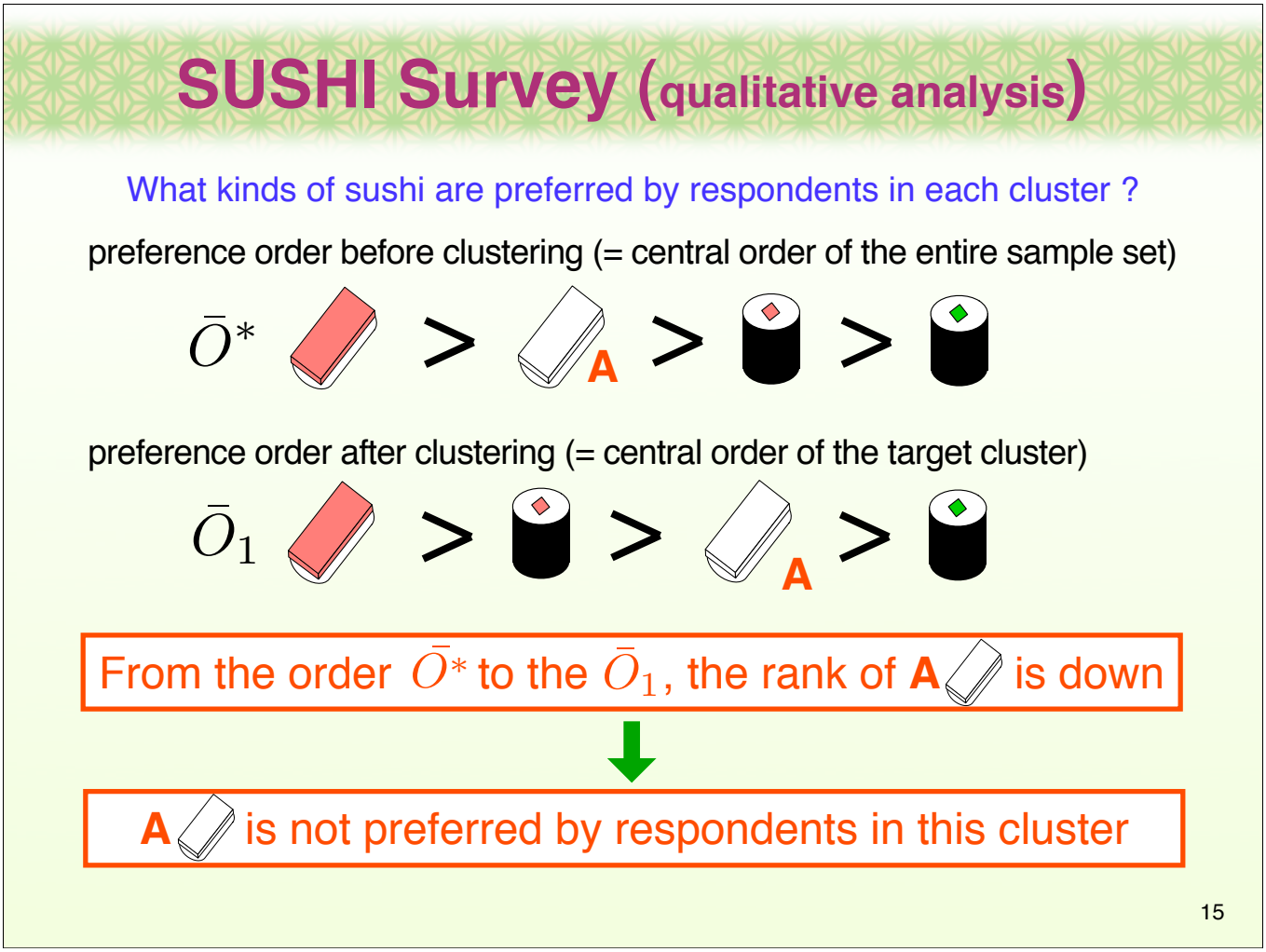
Tested on artificially generated data
How correctly clusters can be recovered?



- Two k-o'means are superior to traditional hierarchical clustering
- The difference between EBC and TMSE are very small

14

Two k-o'means and a traditional hierarchical clustering method are tested on artificially generated data.
We checked how correctly clusters can be recovered.
Two k-o'means are clearly superior to a traditional hierarchical clustering with the d_rho dissimilarity.
Further, the difference between EBC and TMSE are very small.
Therefore, we can conclude that our new k-o'means-EBC method can improve the efficiency without sacrificing the prediction accuracy.



Next, we applied our new method to a survey data in terms of preference in sushi, a Japanese food.
I explored what kinds of sushi are preferred by respondents in each cluster.
To this aim, for each kind of sushi, we checked its rank in the following two orders.
The one is the preference order before clustering, that is the central order of the entire sample set.
We consider that this order indicates the neutral preferences.
The other is the preference order after clustering, that is the central order of the target cluster.
From this order to this order, the rank of the sushi A is down, it can be concluded that the sushi A is not preferred by the respondents in this cluster.

SUSHI Survey (qualitative analysis)

10 most ranked up or down sushi of each cluster

C 1	prefer	egg +74	cucumber roll +62	fermented bean roll +38	octopus +36	*inari* +33
		salad +29	pickled plum & perilla leaf roll +28	fermented bean +26	perilla leaf roll +24	raw beef +21
	not prefer	flying fish -10	young yellowtail -12	*battera* -13	sea bass -14	amberjack -37
		hardtail -41	fluke fin -46	abalone -63	sea urchin -84	salmon roe -85
C 2	prefer	ark shell +63	crab liver +39	turban shell +26	sea bass +23	abalone +22
		tsubu shell +16	angler liver +16	sea urchin +15	clam +13	hardtail +13
	not prefer	chili cod roe roll -15	pickled plum roll -15	shrimp -17	tuna roll -19	egg -19
		salad roll -27	*inari* -30	salad -32	octopus -57	squid -82

RED: prefer (rank up) BLUE: not prefer (rank down)

VALUES: (rank before clustering) - (rank after clustering)

16

This table might make you feel hungry. But, please wait going supper for a while.

Respondents are divided into 2 clusters. We picked up 10 most ranked up and down sushi in each cluster.

Red entries show preferred, that is, ranked up sushi.

Blue entries show not preferred, that is, ranked down sushi.

Now, we observe C1 cluster.

[PUSH] These (red marked) are so-called “blue fish,” rather oily and smelly.

This result show that C1 respondents don’t prefer oily sushi.

[PUSH] These (blue marked) are very economic sushi.

These are ranked up, but are still in the middle portions of the preference order.

Therefore, it should say that C1 respondents don’t dislike these sushi.

Conclusion

***k*-o'means**: a method for clustering orders

- We proposed a faster technique, **Expected Borda Count method**, for deriving central orders based on order statistics theory
- We successfully improved the efficiency without sacrificing the the accuracy

Errata

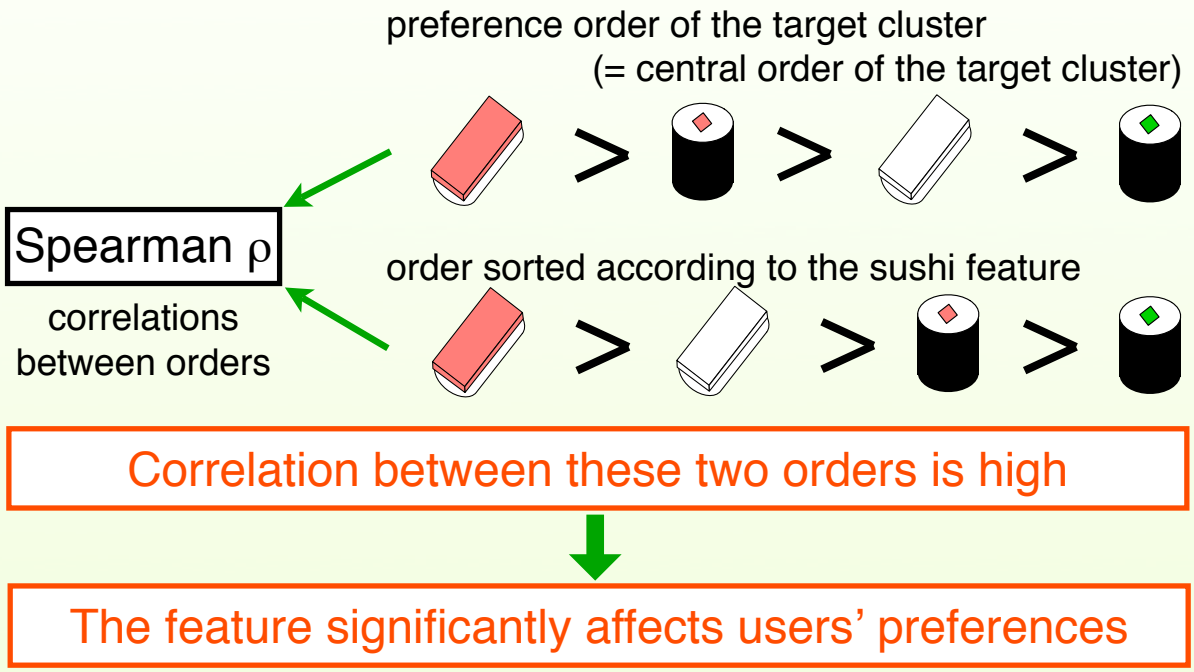
- Table 1: upper-right part: clam ♠ ➡ clam ♡

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

We would like to conclude our talk.
We proposed a new method for clustering orders: k-o'means-EBC.
We successfully improved the efficiency without sacrificing the accuracy.
And, sorry for a minor error in our article.
That's all we have to say. Thank you for your attention.

SUSHI Survey (influences of features)

Influences of sushi features (ex. price, taste) to preferences



Next, we applied our new method to a survey data in terms of preference in sushi, a Japanese food.
First, we analyzed influences of sushi features to preferences.
For this aim, we compared the two orders.
The one is the preference order of the target cluster, that is, the central order of the target cluster.
The other is the order that sorted according to the sushi feature.
We can say that, if correlation between these two orders are high, the feature significantly affects preferences.

SUSHI Survey (influences of features)

Divided into 2 clusters by *k*-o'means-EBC

	C1		C2
# of respondents	2313		2687
prefer heavy tasting sushi	+0.0999	$\prec$	0.3656
prefer sushi that respondents infrequently eat	-0.5662	$\approx$	-0.4228
prefer inexpensive sushi	-0.0012	$\succ$	-0.4965
prefer sushi that fewer shops supply	-0.1241	$\approx$	-0.1435

SUMMARY: C2 respondents prefer more expensive and heavy tasting sushi than C1 respondents

19

Respondents are divided into 2 clusters.

These are correlation between preferences and features.

In terms of heaviness, or oiliness, of sushi, the correlation of the cluster C2 is clearly higher than that of C1.

So, C2 respondents prefer heavy tasting sushi.

In summary, C2 respondents prefer more expensive and heavy tasting sushi than C1.