
Efficient Clustering for Orders

Toshihiro Kamishima and Shotaro Akaho

National Institute of Advanced Industrial Science and Technology (AIST),
AIST Tsukuba Central 2, Umezono 1-1-1, Tsukuba, Ibaraki, 305-8568 Japan
mail@kamishima.net
http://www.kamishima.net/
s.akaho@aist.go.jp

Abstract. Lists of ordered objects are widely used as representational forms. Such ordered objects include Web search results or best-seller lists. Clustering is a useful data analysis technique for grouping mutually similar objects. To cluster orders, hierarchical clustering methods have been used together with dissimilarities defined between pairs of orders. However, hierarchical clustering methods cannot be applied to large-scale data due to their computational cost in terms of the number of orders. To avoid this problem, we developed an k -o' means algorithm. This algorithm successfully extracted grouping structures in orders, and was computationally efficient with respect to the number of orders. However, it was not efficient in cases where there are too many possible objects yet. We therefore propose a new method (k -o' means-EBC), grounded on a theory of order statistics. We further propose several techniques to analyze acquired clusters of orders.

15.1 Introduction

The term *order* indicates a sequence of objects sorted according to some property. Such orders are widely used as representational forms. For example, the responses from Web search engines are lists of pages sorted according to their relevance to queries. Best-seller lists, which are item-sequence sorted according to sales volume, are used on many E-commerce sites.

Orders have also been exploited for sensory test of human respondents' sensations, impressions, or preference. For such a kind of surveys, it is typical to adopt a scoring method. In this method, a respondents' sensation is measured using a scale on which extremes are represented by antonymous words. One example is a five-point-scale on which 1 and 5 indicate *don't prefer* and *prefer*, respectively. If one very much prefers an apple, he/she rates the apple as 5. Though this scoring method is widely used, it is not the best way for all types of sensory test. For example, as pointed out in [1], a trained expert, e.g., a wine taster, can maintain a consistent mapping from his/her sensation level to rating score throughout a given session. However, users' mappings generally change for each response, especially if the intervals between responses are long. Hence, even if two respondents rate the same item at the same score, their true degrees of sensation may not be the same. When effects of such demerits cannot be ignored, a ranking method is used. In this method, respondents show their degree of sensation by orders, i.e., object sequences according to the degree of a target sensation.

In this case, respondents' sensation patterns are represented by orders, and analysis techniques for orders are required.

Orders are also useful when the absolute level of observations cannot be calibrated. For example, when analyzing DNA microarray data, in order that the same fluorescence level represents the same level of gene expression, experimental conditions must be calibrated. However, DNA databases may consist of data sampled under various conditions. Even in such cases, the higher level of fluorescence surely corresponds to the higher level of gene expression. Therefore, by treating the values in the microarray data as ordinal values, non-calibrated data would be processed. Fujibuchi et al. adopted such use of orders in searching a gene expression database for similar cell types [2].

And clustering is the task of partitioning a sample set into clusters having the properties of internal cohesion and external isolation [3]. This method is a basic tool for exploratory data analysis. Clustering methods for orders are useful for revealing the group structure of data represented by orders such as those described above.

To cluster a set of orders, classical clustering has been mainly used [4, chapter 2]. In these studies, clustering methods were applied to ordinal data of a social survey, sensory test, etc. These data sets have been small in size; the number of objects to be sorted and the length of orders are at most ten, and the number of orders to be clustered are at most thousands. This is because an scoring method has been used to acquire responses for a large-scale survey. Responses can easily be collected by requesting for respondents to mark on rating scales that are printed on paper questionnaire forms. On the other hand, using printed questionnaire forms is not appropriate for ranking method, because respondents must rewrite entire response orders when they want to correct them. Therefore, in a ranking method, respondents generally reply by sorting real objects. For example, respondents are requested to sort glasses of wine according to their preference. However, it would be costly to prepare so many glasses. Due to this reason, ranking method has been used for a small-scale survey, even if its advantage to an scoring method is known as described above. But now, adoption of computer interface clear this obstacle in using a ranking method. Respondents can sort virtual objects instead of real objects. Further, methods to implicitly collect preference orders have proposed [5, 6]. These technical progress has made it easier to collect the large number of ordinal data.

We can now collect a large-scale data that consist of orders. However, current techniques for clustering orders are not fully scalable. For example, to cluster a set of orders, dissimilarities are first calculated for all pairs of orders, and agglomerative hierarchical clustering techniques are applied. This approach is computationally inefficient, because computational cost of agglomerative hierarchical clustering is $O(N^2 \log(N))$ under non-Euclidean metric [7], where N is the number of orders to be clustered. To alleviate this inefficiency in terms of N , we proposed a k -means-type algorithm ***k*-o'means** in our previous work [8]. The computational complexity was reduced to $O(N)$ in terms of the number of orders. Though this method successfully extracted a grouping structure in a set of orders, it was not efficient yet, if the number of possible objects to be sorted was large. In this paper, to alleviate this inefficiency, we propose a new method, ***k*-o'means-EBC**. Note that EBC means **Expected Borda Count**, which is a classic method to find an order so as to be as concordant as possible with a given set of orders. And

incompleteness in orders are processed based on a theory of order statistics. Additionally, we propose several methods for interpreting the clusters of orders.

We formalize this clustering task in Section 15.2. Our previous and new clustering methods are presented in Section 15.3. The experimental results are shown in Sections 15.4 and 15.5. Section 15.6 summarizes our conclusions.

15.2 Clustering Orders

In this section, we formalize the task of clustering orders. We start by defining our basic notations regarding orders. An object, entity, or substance to be sorted is denoted by x_j . The universal object set, X^* , consists of all possible objects, and L^* is defined as $|X^*|$. The order is denoted by $O = x_a \succ \dots \succ x_j \succ \dots \succ x_b$. Note that subscript j of x doesn't mean "The j -th object in this order," but that "The object is uniquely indexed by j in X^* ." The order $x_1 \succ x_2$ represents " x_1 precedes x_2 ." An object set $X(O_i)$ or simply X_i is composed of all objects in the order O_i . The length of O_i , i.e., $|X_i|$, is shortly denoted by L_i . An order of all objects, i.e., O_i s.t. $X(O_i) = X^*$, is called a complete order; otherwise, the order is incomplete. Rank, $r(O_i, x_j)$ or simply r_{ij} , is the cardinal number that indicates the position of the object x_j in the order O_i . For example, for $O_i = x_1 \succ x_3 \succ x_2$, $r(O_i, x_2)$ or r_{i2} is 3. Two orders, O_1 and O_2 , are concordant if ordinal relations are consistent between any object pairs commonly contained in these two orders; otherwise, they are discordant. Formally, for two orders, O_1 and O_2 , consider an object pair x_a and x_b such that $x_a, x_b \in X_1 \cap X_2, x_a \neq x_b$. We say that the orders O_1 and O_2 are concordant w.r.t. x_a and x_b if the two objects are placed in the same order, i.e., $(r_{1a} - r_{1b})(r_{2a} - r_{2b}) \geq 0$; otherwise, they are discordant. Further, O_1 and O_2 are concordant if O_1 and O_2 are concordant w.r.t. all object pairs such that $x_a, x_b \in X_1 \cap X_2, x_a \neq x_b$.

A pair set $\text{Pair}(O_i)$ is composed of all the object pairs $x_a \succ x_b$, such that x_a precedes x_b in the order O_i . For example, from the order $O_1 = x_3 \succ x_2 \succ x_1$, three object pairs, $x_3 \succ x_2$, $x_3 \succ x_1$, and $x_2 \succ x_1$, are extracted. For a set of orders S , the $\text{Pair}(S)$ is composed of all pairs in $\text{Pair}(O_i)$ of $O_i \in S$. Note that if the same object pairs are contained in numbers of $\text{Pair}(O_i)$, these pairs are multiply added into the $\text{Pair}(S)$. For example, if the same object pairs $x_1 \succ x_2$ are extracted from O_5 and O_7 in S , both two ordered pairs $x_1 \succ x_2$ are multiply included in $\text{Pair}(S)$.

The task of clustering orders is as follows. A set of sample orders, $S = \{O_1, O_2, \dots, O_N\}$, $N \equiv |S|$, is given. Note that sample orders may be incomplete, i.e., $X_i \neq X_j, i \neq j$. In addition, O_i and O_j can be discordant. The aim of clustering is to divide the S into a partition. The partition, $\pi = \{C_1, C_2, \dots, C_K\}$, $K = |\pi|$, is a set of all clusters. Clusters are mutually disjoint and exhaustive, i.e., $C_k \cap C_l = \emptyset, \forall k, l, k \neq l$ and $S = C_1 \cup C_2 \cup \dots \cup C_K$. Partitions are generated such that orders in the same cluster are similar (internal cohesion), and those in different clusters are dissimilar (external isolation).

15.2.1 Similarity between Two Orders

Clusters are defined as a collection of *similar* orders; thus, the similarity measures between two orders are required. *Spearman's ρ* [9, 4] is one such measure, signifying the

correlation between ranks of objects. The ρ between two orders, O_1 and O_2 , consisting of the same objects (i.e., $X \equiv X(O_1) = X(O_2)$) is defined as:

$$\rho = \frac{\sum_{x_j \in X} (r_{1j} - \bar{r}_1)(r_{2j} - \bar{r}_2)}{\sqrt{\sum_{x_j \in X} (r_{1j} - \bar{r}_1)^2} \sqrt{\sum_{x_j \in X} (r_{2j} - \bar{r}_2)^2}},$$

where $\bar{r}_i = (1/L) \sum_{x_j \in X} r_{ij}$, $L=|X|$. If no tie in rank is allowed, this can be calculated by the simple formula:

$$\rho = 1 - \frac{6 \sum_{x_j \in X} (r_{1j} - r_{2j})^2}{L^3 - L}. \tag{15.1}$$

The ρ becomes 1 if the two orders are concordant, and -1 if one order is the reverse of the other order. Observing Equation (15.1), this similarity depends only on the term

$$d_S(O_1, O_2) = \sum_{x_j \in X} (r_{1j} - r_{2j})^2. \tag{15.2}$$

This is called **Spearman’s distance**. If two or more objects are tied, we give the same *midrank* to these objects [4]. For example, consider an order $x_5 \succ x_2 \sim x_3$ (“ \sim ” denotes a tie in rank), in which x_2 and x_3 are ranked at the 2nd or 3rd positions. In this case, the midrank 2.5 is assigned to both objects.

Another widely used measure of the similarity of orders is *Kendall’s τ* . Intuitively, this is defined as the number of concordant object pairs subtracted by that of discordant pairs, and then it is normalized. Formally, Kendall’s τ is defined as

$$\tau = \frac{1}{L(L-1)/2} \sum_{x_a \succ x_b \in \text{Pair}(O_1)} \text{sgn}((r_{1a} - r_{1b})(r_{2a} - r_{2b})), \tag{15.3}$$

where $\text{sgn}(x)$ is a sign function that takes 1 if $x > 0$, 0 if $x = 0$, and -1 otherwise. Many other types of similarities between orders have been proposed (see [4, chapter 2]), but the above two are widely used and have been well studied.

In this paper, we adopt Spearman’s ρ rather than Kendall’s τ because of the following reasons: First, these two measures have similar properties. Both measures of similarities between two random orders asymptotically follow normal distribution as the length of the orders grows. Additionally, these are highly correlated, because the difference between the two measures is bounded by Daniels’ inequality [9]:

$$-1 \leq \frac{3(L+2)}{L-2} \tau - \frac{2(L+1)}{L-2} \rho \leq 1.$$

Second, Spearman’s ρ can be calculated more quickly. All of the object pairs have to be checked to derive Kendall’s τ , so $O(L^2)$ time is required. In the case of Spearman’s ρ , the most time consuming task is sorting objects to decide their ranks; thus, the time complexity is $O(L \log L)$. Further, the central orders under Spearman distance is tractable, but the derivation under Kendall’s distance is NP-hard [10].

For the clustering task, distance or dissimilarity is more useful than similarity. We defined a dissimilarity between two orders based on ρ :

$$d_\rho(O_1, O_2) = 1 - \rho(O_1, O_2). \tag{15.4}$$

Since the range of ρ is $[-1, 1]$, this dissimilarity ranges $[0, 2]$. This dissimilarity becomes 0 if the two orders are concordant.

15.3 Methods

Here, we describe exiting clustering methods and our new clustering method.

15.3.1 Hierarchical Clustering Methods

In the literature of psychometrics, questionnaire data obtained by a ranking method have been processed by traditional clustering techniques [4]. First, for all pairs of orders in S , the dissimilarities in Section 15.2.1 are calculated, and a dissimilarity matrix for S is obtained. Next, this matrix can be clustered by standard hierarchical clustering methods, such as the group average method. In these survey researches, the size of the processed data set is rather small ($N < 1000$, $L^* < 10$, $L_i < 10$). Therefore, hierarchical clustering methods could cluster order sets, even though the time complexity of these methods is $O(N^2 \log(N))$ under non-Euclidean metric [7] and is costly. However, these method cannot be applied to a large-scale data, due to their computational cost.

Additionally, when the number of objects, L^* , is large, it is hard for respondents to sort all objects in X^* . Therefore, sample orders are generally incomplete, i.e., $X(O_i) \subset X^*$, the dissimilarities cannot be calculated because the dissimilarity measures are defined between two orders consisting of the same objects. One way to deal with incomplete orders is to introduce the notion of an *Incomplete Order Set* (IOS)¹ [4], which is defined as a set of all possible complete orders that are concordant with the given incomplete order. Given the incomplete order O that consists of the object set X , an IOS is defined as

$$\text{ios}(O) = \{O_i^* | O_i^* \text{ is concordant with } O, X(O_i^*) = X^*\}.$$

This idea is not fit for large-scale data sets because the size of the set is $(L^*/L!)$, which grows exponentially in accordance with L^* . Additionally, there are some difficulties in defining the distances between the two sets of orders. One possible definition is to adopt the arithmetic mean of the distances between orders in each of the two sets. However, this is not distance because $d(\text{ios}_a, \text{ios}_a)$ may not be 0. Therefore, more complicated distance, i.e., Hausdorff distance, has to be adopted.

Since the above IOS cannot be derived for a large-scale data set, we adopted the following heuristics in this paper. In such cases, the dissimilarity between the orders is determined based on the the objects included in both. Take, for example, the following two orders:

$$O_1 = x_1 \succ x_3 \succ x_4 \succ x_6, \qquad O_2 = x_5 \succ x_4 \succ x_3 \succ x_2 \succ x_6.$$

¹ In [4], this notion is referred by the term *incomplete ranking*, but we have adopted IOS to insist that this is a set of orders.

From these orders, all objects that are not included in both orders are eliminated. The generated orders become:

$$O'_1 = x_3 \succ x_4 \succ x_6, \quad O'_2 = x_4 \succ x_3 \succ x_6.$$

The ranks of objects in these orders are:

$$\begin{aligned} r(O'_1, x_3) &= 1, \quad r(O'_1, x_4) = 2, \quad r(O'_1, x_6) = 3; \\ r(O'_2, x_3) &= 2, \quad r(O'_2, x_4) = 1, \quad r(O'_2, x_6) = 3. \end{aligned}$$

Consequently, the Spearman's ρ becomes

$$\rho = 1 - \frac{6((1-2)^2 + (2-1)^2 + (3-3)^2)}{3^3 - 3} = 0.5.$$

If no common objects exists between the two orders, $\rho = 0$ (i.e., no correlation).

15.3.2 *k*-o'means-TMSE (Thurstone Minimum Square Error)

In [8], we proposed a *k*-o'means algorithm as a clustering method designed to process orders. To differentiate our new algorithm described in detail later, we call it by a ***k*-o'means-TMSE** algorithm.

A *k*-o'means-TMSE in Figure 15.1 is similar to the well-known *k*-means algorithm [11]. Specifically, an initial cluster is refined by the iterative process of estimating new cluster centers and the re-assigning of samples. This process is repeated until no changes in the cluster assignment is detected or the pre-defined iteration time is reached. However, different notions of dissimilarity and cluster centers have been used to handle orders. For the dissimilarity $d(\bar{O}_k, O_i)$, equation (15.4) was used in step 4. As a cluster center in step 3, we used the following notion of a *central order* [4]. Given a set of

Algorithm *k*-o'means(*S*, *K*, *maxIter*)

S = {*O*₁, ..., *O*_{*N*}}: a set of orders

K: the number of clusters

maxIter: the limit of iteration times

- 1) *S* is randomly partitioned into a set of clusters: $\pi = \{C_1, \dots, C_K\}$,
 $\pi' := \pi, t := 0$.
 - 2) $t := t + 1$, if $t > \text{maxIter}$ goto step 6.
 - 3) for each cluster $C_k \in \pi$,
 derive the corresponding central order \bar{O}_k .
 - 4) for each order O_i in *S*,
 assign it to the cluster: $\arg \min_{C_k} d(\bar{O}_k, O_i)$.
 - 5) if $\pi = \pi'$ then goto step 6; else $\pi' := \pi$, goto step 2.
 - 6) output π .
-

Fig. 15.1. *k*-o'means algorithm

orders C_k and a dissimilarity measure between orders $d(O_a, O_b)$, a central order \bar{O}_k is defined as the order that minimizes the sum of dissimilarities:

$$\bar{O}_k = \arg \min_O \sum_{O_i \in C_k} d(O, O_i). \tag{15.5}$$

Note that the order \bar{O}_k consists of all the objects in C_k , i.e., $X_{C_k} = \cup_{O_i \in C_k} X(O_i)$. The dissimilarity $d(\bar{O}_k, O_i)$ is calculated over common objects as in Section 15.3.1. However, because $X_i \subseteq X(\bar{O}_k)$, the dissimilarity can always be calculated over X_i . Unfortunately, the optimal central order is not tractable except for a special cases. For example, if using a Kendall distance, the derivation of central orders is NP-hard even if all sample orders are complete [10].

Therefore, many approximation methods have been developed. However, to use as a sub-routine in a k -o’means algorithm, the following two constraints must be satisfied. First, the method must deal with incomplete orders that consist of objects randomly sampled from X^* . In [12], they proposed a method to derive a central order of top k lists, which are special kinds of incomplete orders. Top k list is an order that consists of the most preferred k objects, and the objects that are not among the top k list are implicitly ranked lower than these k objects. That is to say, the top k objects of a hidden complete order are observed. In our case, objects are randomly sampled, and such a restriction is not allowed. Second, the method should be executed without using iterative optimization techniques. Since central orders are derived K times in each loop of the k -o’means algorithm, the derivation method of central orders would seriously affect efficiency if it adopts the iterative optimization.

To our knowledge, the method satisfying these two constraints is the following one to derive the minimum square error solution under a generative model of Thurstone’s law of comparative judgment [13]. Because we used this method to derive central orders, we call this clustering algorithm by the k -o’means-TMSE (Thurstone Minimum Square Error) algorithm. We describe this method for deriving central orders. First, the probability $\Pr[x_a \succ x_b]$ is estimated. The pair set of $\text{Pair}(C_k)$ in Section 15.2 is generated from C_k in step 3 of k -o’means-TMSE. Next, we calculate the probabilities for every pair of objects in C_k :

$$\Pr[x_a \succ x_b] = \frac{|x_a \succ x_b| + 0.5}{|x_a \succ x_b| + |x_b \succ x_a| + 1},$$

where $|x_a \succ x_b|$ is the number of the object pairs, $x_a \succ x_b$, in the $\text{Pair}(C_k)$. These probabilities are applied to a model of Thurstone’s law of comparative judgment. This model assumes that scores are assigned to each object x_l , and an order is derived by sorting according to these scores. Scores follow a normal distribution; i.e., $N(\mu_l, \sigma)$, where μ_l is the mean score of the object x_l , and σ is a common constant standard deviation. Based on this model, the probability that object x_a precedes the x_b is

$$\begin{aligned} \Pr[x_a \succ x_b] &= \int_{-\infty}^{\infty} \phi\left(\frac{t - \mu_a}{\sigma}\right) \int_{-\infty}^t \phi\left(\frac{u - \mu_b}{\sigma}\right) du dt \\ &= \Phi\left(\frac{\mu_a - \mu_b}{\sqrt{2}\sigma}\right), \end{aligned} \tag{15.6}$$

where $\phi(\cdot)$ is a normal distribution density function, and $\Phi(\cdot)$ is a normal cumulative distribution function. Under the minimum square error criterion of this model [14], μ'_l , which is a linearly transformed image of μ_l , is analytically derived as

$$\mu'_l = \frac{1}{|X_{C_k}|} \sum_{x \in X_{C_k}} \Phi^{-1}(\Pr[x_l \succ x]), \tag{15.7}$$

where $X_{C_k} = \bigcup_{O_i \in C_k} X_i$. The value of μ'_l is derived for each object in X_{C_k} . Finally, the central order \bar{O}_k can be derived by sorting according to the corresponding μ'_l . Because the resultant partition by k -o’means-TMSE is dependent on the initial cluster, this algorithm is run multiple times, randomly changing the initial cluster; then, the partition minimizing the following total error is selected:

$$\sum_{C_k \in \pi} \sum_{O_i \in C_k} d(O_i, \bar{O}_k). \tag{15.8}$$

This k -o’means-TMSE could successfully find the cluster structure in a set of incomplete orders due to the following reason: Because the dissimilarity in Section 15.3.1 was measured between two orders, the precision of the dissimilarities was unstable. On the other hand, in the case of k -o’means-TMSE, central orders are calculated based on the $|C_k|$ orders. $|C_k|$ is generally much larger than two, and much more information is available; thus, the central order can be stably calculated. The dissimilarity between the central orders and each sample order can be stably measured, too, because all of objects in a sample order always exist in the corresponding central order and so the full information in the sample orders can be considered.

However, the k -o’means-TMSE is not so efficient in terms of time and memory complexity. Time or memory complexity in N and K is linear, and these are efficient. However, complexity in terms of L^* is quadratic, and further, the constant factor is rather large due to the calculation of the inverse function of a normal distribution. Due to this inefficiency, this algorithm cannot be used if L^* is large. To overcome this inefficiency, we propose a new method in the next section.

15.3.3 k -o’means-EBC (Expected Borda Count)

To improve efficiency in computation time and memory requirement, though we used the k -o’means framework in Figure 15.1 and the dissimilarity measure d_ρ of equation (15.4) in step 4 of Figure 15.1, we employed other types of derivation procedures for the central orders.

Below, we describe this derivation method for a central order \bar{O}_k of a cluster C_k in step 3 of Figure 15.1. We call this the **Expected Borda Count**(EBC) method, and our new clustering method is called a **k -o’means-EBC** algorithm. The Borda Count method is used to derive central orders from complete orders; we modified this so as to make it applicable to incomplete orders. The Borda Count method [15] was originally developed for determining the order of candidates in an election from a set of ranking votes. A set of complete orders, C_k , is given. First, for each object x_j in X^* , the vote count is calculated:

$$\text{vote}(x_j) = \sum_{O_i \in C_k} (L^* - r_{ij} + 1).$$

Then, a central order is derived by sorting objects $x_j \in X^*$ in descending order of $\text{vote}(x_j)$. Clearly, this method is equivalent to sorting the objects in ascending order of the following mean ranks:

$$\bar{r}_j = \frac{1}{|C_k|} \sum_{O_i \in C_k} r_{ij}. \tag{15.9}$$

If all sample orders are complete and Spearman’s distance is used, it is known that the central order derived by the above Borda Count optimally minimizes Equation (15.5) [4, theorem 2.2].

Because all sample orders are complete, Spearman’s distance is proportional to the distance d_ρ . Therefore, even in the case that d_ρ is used as dissimilarity, the optimal central order can be derived by this Borda Count method. This optimal central order can also be considered as a maximum likelihood estimator of the Mallows- θ model [16]. The Mallows- θ model is a distribution model of the complete order O , and is defined as

$$\Pr[O; O_0, \theta] \propto \exp(\theta d_S(O_0, O)), \tag{15.10}$$

where the parameters θ and O_0 are called a dispersion parameter and a modal order, respectively.

Unfortunately, this original Borda Count method cannot be applied to incomplete orders. To cope with incomplete orders, we must show the facts known in the order statistics literature. First, we assume that there is hidden complete order O_h^* which is randomly generated. A sample order $O_i \in C_k$ is generated by selecting objects from this O_h^* uniformly at random. That is to say, from a universal object set X^* , L_i objects are sampled without replacement; then, O_i is generated by sorting these objects so as to be concordant with O_h^* . Now we are given O_i generated through this process. In this case, the complete order O_h^* follows the distribution:

$$\Pr[O_h^* | O_i] = \begin{cases} \frac{L_i!}{L^*!} & \text{if } O_h^* \text{ and } O_i \text{ are concordant,} \\ 0 & \text{otherwise.} \end{cases} \tag{15.11}$$

Based on the theory of order statics from a without-replacement sample [17, section 3.7], if an object x_j is contained in X_i , the conditional expectation of ranks of the object x_j in the order O_h^* given O_i is

$$E[r_j^* | O_i] = r_{ij} \frac{L^* + 1}{L_i + 1}, \quad \text{if } x_j \in X_i, \tag{15.12}$$

where the expectation is calculated over all possible complete orders, O_h^* , and $r_j^* \equiv r(O_h^*, x_j)$. If an object x_j is not contained in X_i , the object is at any rank in the hidden complete order uniformly at random; thus, an expectation of ranks is

$$E[r_j^* | O_i] = \frac{1}{2}(L^* + 1), \quad \text{if } x_j \notin X_i. \tag{15.13}$$

Next, we turn to the case where a set of orders, C_k , consists of orders independently generated through the above process. Each $O_i \in C_k$ is first converted to a set of all complete orders; thus, the total number of complete orders is $L^*|C_k|$. For each complete order, we assign weights that follow equation (15.11). By the Borda Count method, an optimal central order for these weighted complete orders can be calculated. The mean rank of x_j (equation (15.9)) for these weighted complete orders is

$$\begin{aligned} E[\bar{r}_j] &= \frac{1}{|C_k|} \sum_{O_i \in C_k} \sum_{O_h^* \in S(L^*)} \Pr[O_h^*|O_i] r(O_h^*, x_j) \\ &= \frac{1}{|C_k|} \sum_{O_i \in C_k} E[r_j^*|O_i], \end{aligned} \tag{15.14}$$

where $S(L^*)^2$ is a set of all complete orders. A central order is derived by sorting objects $x_j \in X_{C_k}$ in ascending order of the corresponding $E[\bar{r}_j]$. Since objects are sorted according to the means of expectation of ranks, we call this method an **Expected Borda Count** (EBC).

A central order derived by an EBC method is optimal if the distance $d(O_i, \bar{O}_k)$ is measured by

$$\sum_{O_h^* \in S(L^*)} \Pr[O_h^*|O_i] d_S(O_h^*, \bar{O}_k). \tag{15.15}$$

Hence, in step 4 of Figure 15.1, not d_ρ , but this equation (15.15) should be used. However, it is intractable to compute equation (15.15), because its computational complexity is $O(L^*(L^*/L_i!))$. Therefore, we adopt d_ρ , and it empirically performed well, as is shown later. Furthermore, if all sample orders are complete, d_ρ is compatible with equation (15.15). Note that we also tried

$$d(\bar{O}, O_i) \sum_{x_j \in X^*} (r(\bar{O}_k, x_j) - E[r_j^*|O_i])^2,$$

but empirically, it performed poorly.

The time complexity of a k -o’means-EBC is

$$O(K \max(N\bar{L} \log(\bar{L}), L^* \log L^*)), \tag{15.16}$$

where \bar{L} is the mean of L_i over S . First, in step 3 of Figure 15.1, the K central orders are derived. For each cluster, $O((N/K)\bar{L})$ time is required for the means of expected ranks and $O(L^* \log L^*)$ time for sorting objects. Hence, the total time required for deriving K central orders is $O(\max(N\bar{L}, KL^* \log L^*))$. Second, in step 4, N orders are classified into K clusters. Because $O(\bar{L} \log \bar{L})$ time is required for calculating one dissimilarity, $O(N\bar{L} \log(\bar{L})K)$ time is required in total. The number of iterations is constant. Consequently, the total complexity becomes equation (15.16).

Note that the uniformity assumption of missing objects might look too strong. However, in the case of a questionnaire survey by ranking methods, the objects to be ranked by respondents can be controlled by surveyors.

² $S(L^*)$ is equivalent to a permutation group of order L^* .

Further, if all the sample orders are first converted into the expected rank vectors, $\langle E[r_1^*|O_i], \dots, E[r_{L^*}^*|O_i] \rangle$, then an original k -means algorithm is applied to these vectors. One might suppose that this k -means is equivalent to our k -o'means-EBC, but this is not the case. A k -means is different from this k -o'means-EBC in terms of the derivation of centers; In the k -means case, the mean vectors of the expected ranks are directly used as cluster centers; in a k -o'means case, these means are sorted and converted to rank values. Therefore, in the k -means case, the centers that correspond to the same central orders are simultaneously kept during clustering. For example, two mean rank vectors $\langle 1.2, 1.5, 4.0 \rangle$ and $\langle 1, 5, 10 \rangle$, correspond to the same central order $x_1 \succ x_2 \succ x_3$, but these two vectors are not differentiated. On the other hand, in a k -o'means-EBC algorithm, they are considered as equivalent, and thus we suppose that the k -o'means-EBC algorithm can find the cluster structure reflecting the ordinal similarities among data.

15.4 Experiments on Artificial Data

We applied the algorithms in Section 15.3 to two types of data: artificially generated data and real questionnaire survey data. In the the former experiment, we examined the characteristics of each algorithm. In the latter experiment of the next section, we analyzed a questionnaire survey data on preferences in sushi.

15.4.1 Evaluation Criteria

The evaluation criteria for partitions was as follows. The same object set was divided into two different partitions: a true partition π^* and an estimated one $\hat{\pi}$. To measure the difference of $\hat{\pi}$ from π^* , we adopted the *ratio of information loss* (RIL) [18], which is also called the uncertainty coefficient in numerical taxonomy literature. The RIL is the ratio of the information that is not acquired to the total information required for estimating a correct partition. This criterion is defined based on the contingency table for indicator functions [11]. The indicator function $I((x_a, x_b), \pi)$ is 1 if an object pair (x_a, x_b) are in the same cluster; otherwise, it is 0. The contingency table is a 2×2 matrix consisting of elements, a_{st} , that are the number of object pairs satisfying the condition $I((x_a, x_b), \pi^*)=s$ and $I((x_a, x_b), \hat{\pi})=t$, among all the possible object pairs. RIL is defined as

$$\text{RIL} = \frac{\sum_{s=0}^1 \sum_{t=0}^1 \frac{a_{st}}{a_{..}} \log_2 \frac{a_{..}}{a_{st}}}{\sum_{s=0}^1 \frac{a_{s.}}{a_{..}} \log_2 \frac{a_{..}}{a_{s.}}}, \tag{15.17}$$

where $a_{.t} = \sum_s a_{st}$, $a_{s.} = \sum_t a_{st}$, and $a_{..} = \sum_{s,t} a_{st}$. The range of the RIL is $[0, 1]$; it becomes 0 if two partitions are identical.

15.4.2 Data Generation Process

Test data were generated in the following two steps: In the first step, we generated the K orders to be used as central orders. One permutation (we called it a *pivot*) consisting of all objects in X^* was generated. The other $K - 1$ centers were generated by

Table 15.1. Parameters of experimental data

1) the number of sample orders: $N = 1000$
2) the length of the orders: $L_i = 10$
3) the total number of objects: $L^* = 10, 100$
4) the number of clusters: $K = \{2, 5, 10\}$
5) the inter-cluster isolation: $\{0.5, 0.2, 0.1, 0.001\}$
6) the intra-cluster cohesion: $\{1.0, 0.999, 0.99, 0.9\}$

transforming this pivot. Two adjacent objects in the pivot were randomly selected and exchanged. This exchange was repeated at specified times. By changing the number of exchanges, the inter-cluster isolation could be controlled.

In the second step, for each cluster, constituent orders were generated. From the central order, L_i objects were randomly selected. These objects were sorted so as to be concordant with the central order. Again, two adjacent object pairs were randomly exchanged. By changing the number of times that objects were exchanged, the intra-cluster cohesion could be controlled. Note that the sizes of clusters are equal.

The parameters of the data generator are summarized in Table 15.1. The differences between orders cannot be statistically tested if L_i is too short; on the other respondents cannot sort too many objects. Therefore, we set the order length to $L_i = 10$. **Param 1–2** are common for all the data. The total number of objects (**Param 3**) is set to 10 or 100. All the sample orders are complete if $L^* = 10$, and these are examined in Section 15.4.3. We examine the incomplete case ($L^* = 100$) in Section 15.4.4. **Param 4** was the number of clusters. It is difficult to partition if this number is large, since the sizes of the clusters then decrease. **Param 5** was the inter-cluster isolation that could be tuned by the number of times that objects are exchanged in the first step of the data generation process. This isolation is measured by the probability that the ρ between a pivot and another central order is smaller than that between a pivot and a random order. The larger the isolation, the more easily clusters are separated. **Param 6** was the the intra-cluster cohesion indicating the number of times that objects are exchanged in the second step of the data generation process. This cohesion is measured by the probability that the ρ between the central order and a sample one is larger than that between the central order and a random one. The larger the cohesion, the more easily a cluster could be detected.

For each setting, we generated 100 sample sets. For each sample set, we ran the algorithms five times using different initial partitions; then the best partition in terms of Equation (15.8) was selected. Below, we show the means of RIL over these sets.

15.4.3 Complete Order Case

We analyzed the characteristics of the methods in Section 15.3 by applying these to artificial data of complete orders. The two k -o'-means methods were abbreviated to TMSE and EBC, respectively. Additionally, a group average hierarchical clustering method using dissimilarity as described in Section 15.2.1 was tested, and we denoted

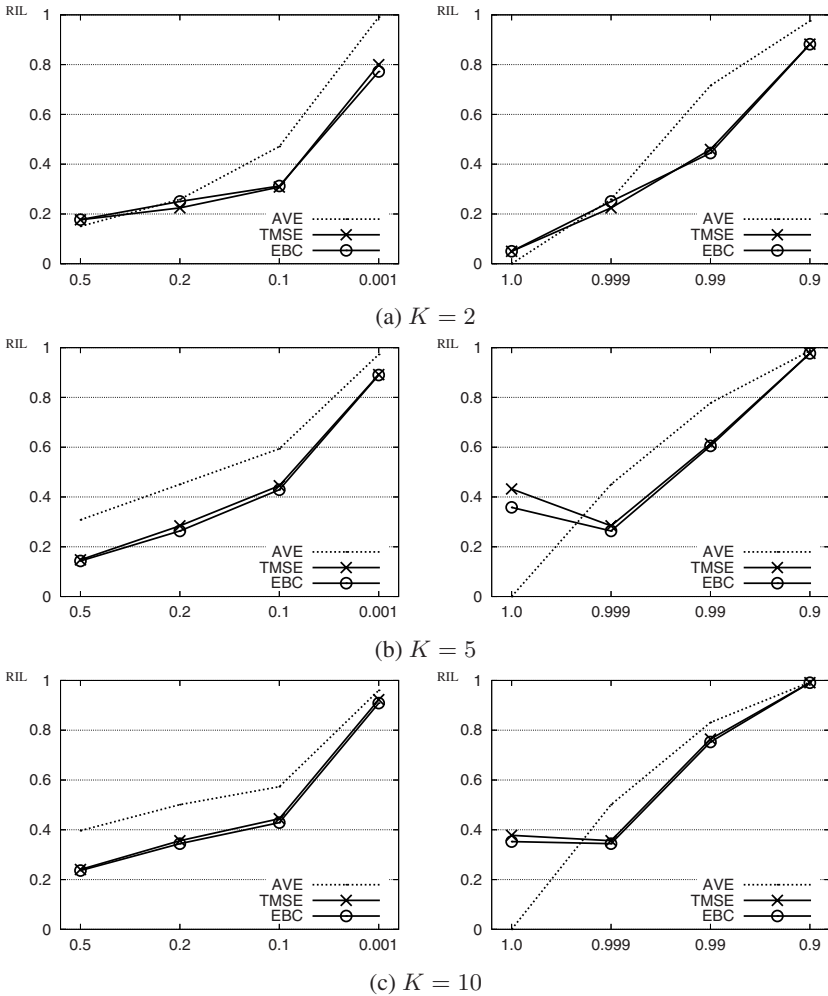


Fig. 15.2. Experimental results on artificial data of complete orders

NOTE: The left charts show the variation of RIL in the inter-cluster isolation when the intra-cluster cohesion is fixed to 0.999. The right charts show the variation of RIL in the intra-cluster cohesion when the inter-cluster isolation is fixed to 0.2.

this result by AVE. The experimental results on artificial data of complete orders (i.e., $L^* = 10$) are shown in Figure 15.2. In Figures 15.2(a), (b), and (c), the means of RIL are shown in cases of $K = 2, 5$, and 10 , respectively. The left three charts show the variation of RIL in the inter-cluster isolation when the intra-cluster cohesion is fixed to 0.999. The right three charts show the variation of RIL in the intra-cluster cohesion when the inter-cluster isolation is fixed to 0.2.

As expected, the more inappropriate clusters were obtained when the inter-cluster isolation or the intra-cluster cohesion decreased and the number of clusters increased.

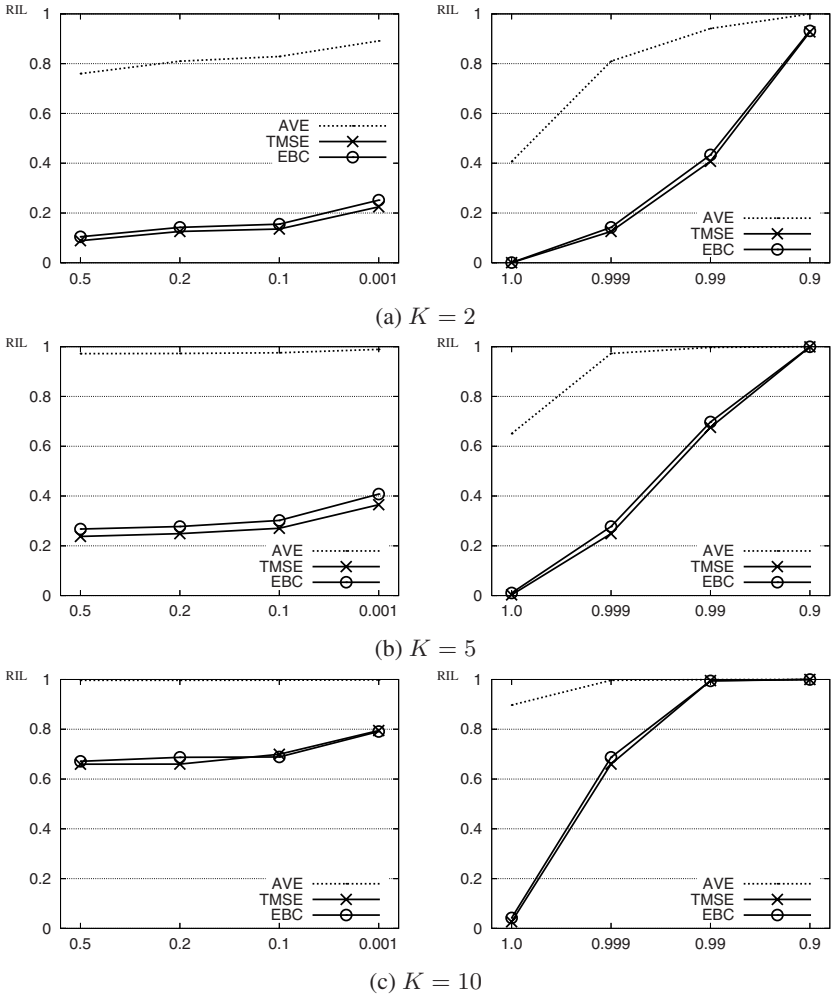


Fig. 15.3. Experimental results on artificial data of incomplete orders

NOTE: See note in Figure 15.2.

We begin with the variation of estimation performance according to the decrease of intra-cluster cohesion. If the cohesion is 1, sample orders are exactly concordant with their corresponding true central orders. In this trivial case, the AVE method succeeds almost perfectly in recovering the embedded cluster structure. Because the dissimilarities between sample orders are 0 if and only if they are in the same cluster, this method could lead to perfect clusters. Though both the EBC and TMSE methods found almost perfect clusters in the $K=2$ case, the performance gradually worsened when K increased. In a k -o’means clustering, a central order is chosen from the finite set, $S(L^*)$. This is contrasted to the fact that a domain of centers is an infinite set in the clustering

of real value vectors. Hence, the central orders of two clusters happen to agree, and one of these clusters is diminished during execution of the k -o'means. As the increase of K , clusters are merged with higher probability. For example, in the EBC case, when $K = 2$ and $K = 5$, clusters are merged in 7% and 35% of the trials, respectively. Such occurrence of merging degrades the ability of recovering clusters. As the cohesion increases, the performance of AVE became more drastically worse than the other two methods. Furthermore, in terms of the inter-cluster isolation, the performance of AVE became drastically worse as K increased, except for the trivial case in which the cohesion was 1. In the AVE method, the determination to merge clusters is based on local information, that is, a pair of clusters. Hence, the chance that orders belonging to different clusters would happen to be merged increases when orders are broadly distributed. When comparing EBC and TMSE, these two methods are almost completely the same.

15.4.4 Incomplete Order Case

We move to the experiments on artificial data of incomplete orders (i.e., $L^* = 100$). The results are shown in Figure 15.3. The meanings of the charts are the same as in Figure 15.2.

TMSE was slightly better than EBC when $K = 2$ and $K = 5$ cases; but EBC overcame TMSE when $K = 10$. AVE was clearly the worst. We suppose that this advantage of the k -o'means is due to the fact that the dissimilarities between order pairs could not be measured precisely if the number of objects commonly included in these two orders is few. Furthermore, the time complexity of AVE is $O(N^2 \log N)$, while the k -o'means algorithms are computationally more inexpensive as in Equation (15.16). When comparing TMSE and EBC, TMSE would be slightly better. However, in terms of time complexity, TMSE's $O(NL^* \max(L^*, K))$ is much worse than EBC's $O(K \max(N\bar{L} \log(\bar{L}), L^* \log L^*))$ if L^* is large. In addition, while the required memory for TMSE is $O(L^{*2})$, EBC demands far less $O(KL^*)$. Therefore, it is reasonable to conclude that k -o'means-EBC is an efficient and effective method for clustering orders.

15.5 Experiments on Real Data

We applied our two k -o'means to questionnaire survey data, and proposed a method to interpret the acquired clusters of orders.

15.5.1 Data Sets

Since the notion of true clusters is meaningless for real data sets, we used the k -o'means as tools for exploratory analysis of a questionnaire survey of preference in sushi (a Japanese food). This data set was collected by the procedure in our previous works [19, 8]. In this data set, $N = 5000$, $L_i = 10$, and $L^* = 100$; in the survey, the probability distribution of sampling objects was not uniform as in equation (15.11). We designed it so that the more frequently supplied sushi in restaurants were more

frequently shown to respondents. Objects were selected independently with probabilities ranging from 3.2% to 0.13%. Therefore, the assumption of the uniformity of the sampling distribution, introduced by the EBC method, was violated. The best result in terms of Equation (15.8) were selected from 10 trials. The number of clusters, K , was set to 2. Note that responses of both authors were clustered into Cluster 1.

15.5.2 Qualitative Analysis of Order Clusters

In [8], we proposed a technique to interpret the acquired clusters based on the relation between attributes of objects and central orders. We applied this method to clusters derived by the EBC and TMSE methods. Table 15.2 shows Spearman’s ρ between central orders of each cluster and an order of objects sorted according to the specific object attributes. For example, the third row presents the ρ between the central order and the sorted object sequence according to their price. Based on these correlations, we were able to learn what kind of object attributes affected the preferences of the respondents in each cluster. We will comment next on each of the object attributes.

Almost the same observations were obtained by both EBC and TMSE. The attribute A1 shows whether the object tasted heavy (i.e., high in fat) or light (i.e., low in fat). The positive correlation indicate a preference for heavy testing. The cluster 2 respondents preferred heavy-tasting sushi. The attribute A2 shows how frequently the respondent eats the sushi. The positive correlation indicates a preference for the sushi that the respondent infrequently eats. Respondents in both clusters preferred the sushi they usually eat. No clear difference was observed between clusters. The attribute A3 is the prices of the objects. The positive correlation indicates a preference for economical sushi. The cluster 2 respondents preferred more expensive sushi. The attribute A4 shows how frequently the objects are supplied at sushi shops. The positive correlation indicates a preference for the objects that fewer shops supply. Though the correlation of cluster 1 was rather larger, the difference was not very clear. Roughly speaking, the members of cluster 2 preferred more heavy-tasting and expensive sushi than those of cluster 1.

In this paper, we propose a new technique based on the changes in object ranks. First, a central order of all the sample orders was calculated, and was denoted by \bar{O}^* . Next, for each cluster, the central orders were also calculated, and were denoted by \bar{O}_k . Then, for each object x_j in X^* , the difference of ranks,

$$\text{rankup}(x_j) = r(\bar{O}^*, x_j) - r(\bar{O}_k, x_j), \tag{15.18}$$

Table 15.2. Relations between clusters and attributes of objects

Attribute	Cluster 1		Cluster 2	
	EBC	TMSE	EBC	TMSE
A1	0.0999	0.0349	0.3656	0.2634
A2	-0.5662	-0.7852	-0.4228	-0.6840
A3	-0.0012	-0.0724	-0.4965	-0.6403
A4	-0.1241	-0.4555	-0.1435	-0.5838

Table 15.3. The top 10 ranked up and the worst 10 ranked down sushi

Cluster 1			Cluster 2	
#	2313		2687	
1	egg ♣	+74	ark shell ♥	+63
2	cucumber roll ♣	+62	crab liver ♠	+39
3	fermented bean roll ♣	+38	turban shell ♥	+26
4	octopus	+36	sea bass	+23
5	deep-fried tofu ♣	+33	abalone ♥	+22
6	salad ♣	+29	<i>tsubu</i> shell	+16
7	pickled plum & perilla leaf roll ♣	+28	angler liver ♠	+16
8	fermented bean ♣	+26	sea urchin ♠	+15
9	perilla leaf roll ♣	+24	clam ♥	+13
10	raw beef	+21	hardtall ◇	+13
	⋮		⋮	
91	flying fish ◇	-10	chili cod roe roll ♣	-15
92	young yellowtail ◇	-12	pickled plum roll ♣	-15
93	<i>battera</i> ◇	-13	shrimp	-17
94	sea bass	-14	tuna roll ♣	-19
95	amberjack ◇	-37	egg ♣	-19
96	hardtall ◇	-41	salad roll ♣	-27
97	fluke fin	-46	deep-fried tofu ♣	-30
98	abalone ♥	-63	salad ♣	-32
99	sea urchin ♠	-84	octopus	-57
100	salmon roe	-85	squid	-82

NOTE: Sushi in each cluster derived by k -o'-means-EBC were sorted in descending order of $\text{rankup}(x_j)$ (Equation (15.18)). In top row labeled “#”, the sizes of clusters were listed. The upper half of the tables show the ranked up sushi, and the bottom half show the ranked down sushi. Just to the right of each sushi name, the $\text{rankup}(x_j)$ values are shown.

was derived. We say that x_j is ranked up if $\text{rankup}(x_j)$ is positive, and that it is ranked down if $\text{rankup}(x_j)$ is negative. If the object x_j was ranked up, it was ranked higher in cluster center \bar{O}_k than in the entire center \bar{O}^* . By observing the sushi whose the absolute values of $\text{rankup}(x_j)$ were large, we investigated the characteristics of each cluster. Table 15.3 list the most 10 ranked up and the most 10 ranked down sushi in clusters derived by k -o'-means-EBC. That is to say, we show the objects whose $\text{rankup}(x_j)$ were the 1st to 10th largest, and were the 1st to 10th smallest. The upper half of the tables shows the ranked up sushi, and the bottom half shows the ranked down sushi. In the top row labeled “#”, the sizes of the clusters are listed. Sushi names that we were not able to translate into English were written using their original Japanese names in *italics*. Just to the right of each sushi name, the $\text{rankup}(x_j)$ values are shown.

We interpreted this table qualitatively. In this table, the mark ♠ indicates objects whose internal organs, such as liver or sweetbread, are eaten. The sushi marked by ◇ are so-called *blue fish*, and those marked by ♥ are clams or shells. These sushi were rather substantial and oily, as revealed in the A1 row of Table 15.2. However, we could not conclude that the respondents in cluster 2 preferred simply oily sushi. For example,

sushi categorized as a *red fish meat*, e.g., fatty tuna, were not listed in the table, because the preference of sushi in this category were similar in both clusters. We can say that the respondents in cluster 2 preferred rather oily sushi, especially blue fish, clam/shell, or liver. The sushi marked by ♣ are very economical. Though these sushi were fairly ranked up in cluster 1, this would not indicate a preference for economical sushi. These would be ranked up because these respondents had sushi that they disliked more than these inexpensive types of sushi. Therefore, to interpret the acquired cluster of orders, not only should the values of equation (15.18) be observed, but also the kind of objects that were ranked up or ranked down.

15.6 Conclusions

We developed a new algorithm for clustering orders called the k -o' means-EBC method. This algorithm is far more efficient in computation and memory usage than k -o' means-TMSE. Therefore, this new algorithm can be applied even if the number of objects L^* is large. In the experiments on artificial data, our k -o' means outperformed the traditional hierarchical clustering. For artificial data, the prediction ability of k -o' means-TMSE is almost equal to that of k -o' means-EBC. Therefore, by taking computational cost into account, it could be concluded that the k -o' means-EBC method was superior to the k -o' means-TMSE for clustering orders. Additionally, we advocated the method to interpret the acquired ordinal clusters.

We plan to improve this method in the following ways. During clustering orders, undesired merges of clusters more frequently occur than in clustering of real value vectors. To overcome this defect, it is necessary to improve the initial clusters. For applying ordinal clustering to DNA microarray data, the curse of dimensionality must be solved. We want to develop a dimension reduction technique for orders like PCA. In the case of an Euclidean space, there are many points far from one point. However, in a case of a space of orders (a permutation group), the order most distant from one order is unique, i.e., the reverse order. Therefore, there are biases for central orders to become exact reversals of themselves. We also would like to lessen this bias.

Acknowledgments. This work is supported by the grants-in-aid 14658106 and 16700157 of the Japan society for the promotion of science.

References

1. Luaces, O., Bayón, G.F., Quevedo, J.R., Díez, J., del Coz, J.J., Bahamonde, A.: Analyzing sensory data using non-linear preference learning with feature subset selection. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 286–297. Springer, Heidelberg (2004)
2. Fujibuchi, W., Kiseleva, L., Horton, P.: Searching for similar gene expression profiles across platforms. In: Proc. of the 16th Int'l Conf. on Genome Informatics, p. 143 (2005)
3. Everitt, B.S.: Cluster Analysis, 3rd edn. Edward Arnold (1993)
4. Marden, J.I.: Analyzing and Modeling Rank Data. Monographs on Statistics and Applied Probability, vol. 64. Chapman & Hall, Boca Raton (1995)

5. Branting, L.K., Broos, P.S.: Automated acquisition of user preference. *Int'l Journal of Human-Computer Studies* 46, 55–77 (1997)
6. Joachims, T.: Optimizing search engines using clickthrough data. In: *Proc. of The 8th Int'l Conf. on Knowledge Discovery and Data Mining*, pp. 133–142 (2002)
7. Olson, C.F.: Parallel algorithms for hierarchical clustering. *Parallel Computing* 21, 1313–1325 (1995)
8. Kamishima, T., Fujiki, J.: Clustering orders. In: Grieser, G., Tanaka, Y., Yamamoto, A. (eds.) *DS 2003. LNCS (LNAD)*, vol. 2843, pp. 194–207. Springer, Heidelberg (2003)
9. Kendall, M., Gibbons, J.D.: *Rank Correlation Methods*, 5th edn. Oxford University Press, Oxford (1990)
10. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the Web. In: *Proc. of The 10th Int'l Conf. on World Wide Web*, pp. 613–622 (2001)
11. Jain, A.K., Dubes, R.C.: *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs (1988)
12. Fligner, M.A., Verducci, J.S.: Distance based ranking models. *Journal of The Royal Statistical Society (B)* 48(3), 359–369 (1986)
13. Thurstone, L.L.: A law of comparative judgment. *Psychological Review* 34, 273–286 (1927)
14. Mosteller, F.: Remarks on the method of paired comparisons: I — the least squares solution assuming equal standard deviations and equal correlations. *Psychometrika* 16(1), 3–9 (1951)
15. de Borda, J.C.: On elections by ballot (1784). In: McLean, I., Urken, A.B. (eds.) *Classics of Social Choice*, pp. 81–89. The University of Michigan Press (1995)
16. Mallows, C.L.: Non-null ranking models. I. *Biometrika* 44, 114–130 (1957)
17. Arnold, B.C., Balakrishnan, N., Nagaraja, H.N.: *A First Course in Order Statistics*. John Wiley & Sons, Inc., Chichester (1992)
18. Kamishima, T., Motoyoshi, F.: Learning from cluster examples. *Machine Learning* 53, 199–233 (2003)
19. Kamishima, T.: Nantonac collaborative filtering: Recommendation based on order responses. In: *Proc. of The 9th Int'l Conf. on Knowledge Discovery and Data Mining*, pp. 583–588 (2003)