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/) and 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.

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.

Clustering is the task of partitioning a sample set into clusters having the properties of internal cohesion and external isolation. This method is a basic tool for exploratory data analysis. 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 [6], 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 [3]. 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 2. Our previous and new clustering methods are presented in Section 3. The experimental results are shown in Sections 4 and 5. Section 6 summarizes our conclusions.

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 $O = x_a \succ \cdots \succ x_j \succ \cdots \succ x_b$. 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 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.

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).

Clusters are defined as a collection of *similar* orders; thus, the similarity measures between two orders are required. *Spearman's* ρ [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:

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

where $\bar{r}_i = (1/L) \sum_{x_j \in X} r_{ij},\ L{=}|X|$. The ρ becomes 1 if the two orders are concordant, and -1 if one order is the reverse of the other order. If no tie in rank is allowed, this can be calculated by the simple formula: $\rho = 1 - 6 d_S(O_1, O_2)/(L^3 - L)$, where $d_S(O_1, O_2)$ is **Spearman's distance**:

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

If two or more objects are tied, we give the same *midrank* to these objects [4]. The time complexity of computing Spearman's ρ is $O(L \log L)$. For the clustering task, distance or dissimilarity is more useful than similarity. We defined a dissimilarity between two orders based on ρ:

$$d_\rho(O_1, O_2) = 1 - \rho(O_1, O_2). \qquad (3)$$

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

3 Methods

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

In [3], 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.

```
Algorithm k-o'means($S$, $K$, $maxIter$)
$S = \{O_1, \dots, O_N\}$: 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 $t > maxIter$ 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 $\pi$.
```

Figure 1. *k*-o'means algorithm

A k-o'means-TMSE in Figure 1 is similar to the well-known k-means algorithm. 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 (3) 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 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:

$$\bar{O}_k = \arg\min_O \sum_{O_i \in C_k} d(O, O_i). \qquad (4)$$

Note that the order $\bar{O}_k$ consists of all the objects in C_k, i.e., $X_{C_k} = \cup_{O_i \in C_k} X(O_i)$. Unfortunately, the optimal central order is not tractable except for a special cases. Instead, we use the following method to derive the minimum square error solution under a generative model of Thurstone's law of comparative judgment [7]. We call this clustering algorithm by the k-o'means-TMSE (Thurstone Minimum Square Error) algorithm.

First, we estimate the probability $\Pr[x_a \succ x_b]$ that x_a precedes x_b. This probability can be easily calculated by counting the number of ordered pairs $x_a \succ x_b$ in samples. 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. Under the minimum square error criterion of this model [5], μ'_l, which is a linearly transformed image of μ_l, is analytically derived as

$$\mu'_l = \tfrac{1}{|X_{C_k}|} \sum_{x \in X_{C_k}} \Phi^{-1}\bigl(\Pr[x_l \succ x]\bigr), \qquad (5)$$

where $\Phi(\cdot)$ is a normal cumulative distribution function and $X_{C_k} = \bigcup_{O_i \in C_k} X_i$. 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:

$$\textstyle\sum_{C_k \in \pi} \sum_{O_i \in C_k} d(O_i, \bar{O}_k). \tag{6}$$

This k-o'means-TMSE could successfully find the cluster structure in a set of incomplete orders.

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. To overcome this inefficiency, we propose a new method in the next section.

3.2 *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 1 and the dissimilarity measure d_ρ of equation (3) in step 4 of Figure 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 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 [2] is used to derive central orders from complete orders; we modified this so as to make it applicable to incomplete orders. This method is equivalent to sorting the objects in ascending order of the following mean ranks: $\bar{r}_j = \frac{1}{|C_k|} \sum_{O_i \in C_k} r_{ij}$. 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 (4) [4, theorem 2.2]. In the case where 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.

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:

$$\Pr[O^*_h | O_i] = \begin{cases} \frac{L_i!}{L^*!} & \text{if } O^*_h \text{ and } O_i \text{ are concordant,} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Based on the theory of order statics from a without-replacement sample [1, 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

$$\mathrm{E}[r^*_j | O_i] = r_{ij} \tfrac{L^*+1}{L_i+1}, \quad \text{if } x_j \in X_i, \tag{8}$$

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

$$\mathrm{E}[r^*_j | O_i] = \tfrac{1}{2}(L^* + 1), \quad \text{if } x_j \notin X_i. \tag{9}$$

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 (7). By the Borda Count method, an optimal central order for these weighted complete orders can be calculated. The mean rank of x_j for these weighted complete orders is

$$\mathrm{E}[\bar{r}_j] = \tfrac{1}{|C_k|} \textstyle\sum_{O_i \in C_k} \mathrm{E}[r^*_j | O_i], \tag{10}$$

where the expectation is calculated over all possible complete orders. A central order is derived by sorting objects $x_j \in X_{C_k}$ in ascending order of the corresponding $\mathrm{E}[\bar{r}_j]$. 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 $\sum_{O^*_h \in S(L^*)} \Pr[O^*_h | O_i] d_S(O^*_h, \bar{O}_k)$. Hence, in step 4 of Figure 1, not d_ρ, but this distance should be used. However, it is intractable to compute this distance because its computational complexity is $O(L^*(L^*!/L_i!))$. Therefore, we adopt d_ρ, and it empirically performed well, as is shown later. The time complexity of a k-o'means-EBC is

$$O\big(K \max(N\bar{L}\log(\bar{L}), L^* \log L^*)\big), \tag{11}$$

where $\bar{L}$ is the mean of L_i over S.

4 Experiments on Artificial Data

We applied the algorithms in Section 3 to artificially generated data, in order to examine the characteristics of each algorithm.

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) [3], 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. The range of the RIL is $[0, 1]$; it becomes 0 if two partitions are identical.

The generation procedure of artificial data sets is the same as that in [3]. The parameters of the data generator are summarized as

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

The inter-cluster isolation is measured by the probability that the ρ between the firstly generated central order and another one is smaller than that between a pivot and a random order. The larger the isolation, the more easily clusters are separated. The intra-cluster 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 (6) was selected. Below, we show the means of RIL over these sets.

The experimental results on artificial data are shown in Figure 2. The results are shown in Figure 2. The two k-o'means methods were abbreviated to TMSE and EBC, respectively. In addition, AVE indicates the result derived by a group average hierarchical clustering in [3]. TMSE was slightly better than EBC, and 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 (11). 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.

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.

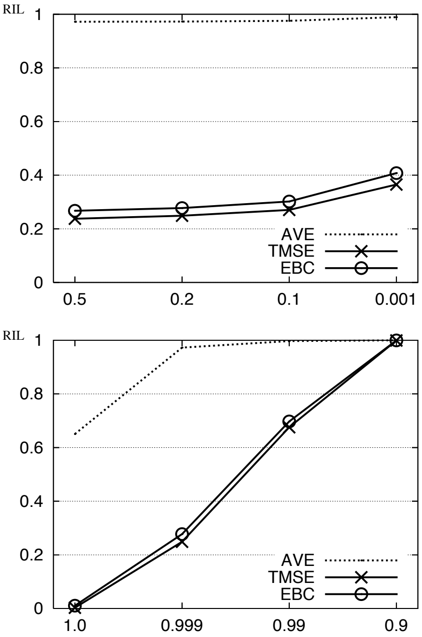
Since the notion of true clusters is meaningless for real data sets, we used the k-o'means as tools for exploratory analysis of a questionnaire survey of preference in sushi (a Japanese food). This data set was collected by the procedure in our previous works [3]. In this data set, $N = 5000$, $L_i = 10$, and $L^* = 100$; in the survey, the probability distribution of sampling objects was not uniform as in equation (7). 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 (6) ware selected from 10 trials. The number of clusters, K, was set to 2. Note that responses of both authors were clustered into 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,

$$\mathrm{rankup}(x_j) = r(\bar{O}^*, x_j) - r(\bar{O}_k, x_j), \qquad (12)$$

was derived. We say that x_j is ranked up if $\mathrm{rankup}(x_j)$ is positive, and that it is ranked down if $\mathrm{rankup}(x_j)$ 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 $\mathrm{rankup}(x_j)$ were large, we investigated the characteristics of each cluster. Table 1 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 $\mathrm{rankup}(x_j)$ 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.

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. However, we could not conclude that the respondents in cluster 2



NOTE: The upper chart shows the variation of RIL in the inter-cluster isolation when the intra-cluster cohesion is fixed to 0.999. The lower chart shows the variation of RIL in the intra-cluster cohesion when the inter-cluster isolation is fixed to 0.2.

Figure 2. Artificial data

Table 1. The top 10 ranked up & 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	*tsubu* 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	hardtail ♢	+13
	⋮		⋮	
91	flying fish ♢	-10	chili cod roe roll ♣	-15
92	young yellowtail ♢	-12	pickled plum roll ♣	-15
93	*battera* ♢	-13	shrimp	-17
94	sea bass	-14	tuna roll ♣	-19
95	amberjack ♢	-37	egg ♣	-19
96	hardtail ♢	-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 $\mathrm{rankup}(x_j)$ (Equation (12)). 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 $\mathrm{rankup}(x_j)$ values are shown.

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 (12) be observed, but also the kind of objects that were ranked up or ranked down.

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. Additionally, we advocated the method to interpret the acquired clusters.

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

References

[1] B. C. Arnold, N. Balakrishnan, and H. N. Nagaraja. *A First Course in Order Statistics*. John Wiley & Sons, Inc., 1992.

[2] J.-C. de Borda. On elections by ballot (1784). In I. McLean and A. B. Urken, editors, *Classics of Social Choice*, chapter 5, pages 81–89. The University of Michigan Press, 1995.

[3] T. Kamishima and J. Fujiki. Clustering orders. In *Proc. of The 6th Int'l Conf. on Discovery Science*, pages 194–207, 2003. [LNAI 2843].

[4] J. I. Marden. *Analyzing and Modeling Rank Data*, volume 64 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, 1995.

[5] F. Mosteller. 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.

[6] C. F. Olson. Parallel algorithms for hierarchical clustering. *Parallel Computing*, 21:1313–1325, 1995.

[7] L. L. Thurstone. A law of comparative judgment. *Psychological Review*, 34:273–286, 1927.