Clustering Orders

Toshihiro Kamishima and Jun Fujiki

AIST Tsukuba Central 2, Umezono 1–1–1, Tsukuba, Ibaraki, 305-8568 Japan
mail@kamishima.net, http://www.kamishima.net/

Abstract. We propose a method of using clustering techniques to partition a set of orders. We define the term *order* as a sequence of objects that are sorted according to some property, such as size, preference, or price. These orders are useful for, say, carrying out a sensory survey. We propose a method called the k-o'means method, which is a modified version of a k-means method, adjusted to handle orders. We compared our method with the traditional clustering methods, and analyzed its characteristics. We also applied our method to a questionnaire survey data on people's preferences in types of *sushi* (a Japanese food).

1 Introduction

Clustering is a task to partition a sample set into clusters which have the properties of internal cohesion and external isolation [1], and is a basic tool for exploratory data analysis. Almost all of the traditional methods for clustering are designed to deal with sample sets represented by the attribute vectors or similarity matrices [2]. These clustering methods, therefore, are not suited for handling other types of data. In this paper, we propose a clustering technique for partitioning one such type of data, namely orders.

We define the term *order* as a sequence of objects that are sorted according to some property. For example, given three objects, x^1, x^2, and x^3, one example of the order is the sequence $x^3 \succ x^1 \succ x^2$, which is sorted according to an individual's preference. An example in which performing the clustering of orders would be useful would be in completing a questionnaire survey on preferences in foods. The surveyor presents some kinds of foods to each respondent, and requests that he/she sort the foods according to his/her preference. By clustering these preference data, the surveyor would be able to find groups which have the same preference tendency. For such a sensory survey, it is typical to adapt the Semantic Differential (SD) method [3]. In this method, the respondents' preferences are measured by a scale, the extremes of which are symbolized by antonymous adjectives. For example:

like 5 4 3 2 1 **dislike**

A proper interpretation of these responses should be, for example, that respondents prefer the objects categorized as "5" to the objects categorized as "4". However, due to lack of analysis techniques, it is common to assume that all respondents share an understanding of this scale's range, intervals, and extremes [4]. Such an unrealistic assumption can be avoided by introducing order responses. We therefore developed a clustering technique for orders.

G. Grieser et al. (Eds.): DS 2003, LNAI 2843, pp. 194–207, 2003.

We formalize this clustering task in Section 2. Our clustering methods are presented in Section 3. The experimental results are shown in Section 4. Section 5 summarizes our conclusions.

1.1 Related Work

Clustering techniques for partitioning time series data have been previously proposed [5,6,7]. Though both orders and time series data are sequences of observations, there is an important difference between them. The same observations can appear in the same sequences of time series data, but cannot appear in the orders. Therefore, these clustering techniques are not suited for dealing with orders.

The pioneering work of handling orders is Thurstone's law of comparative judgment [8]. Thurstone proposed a method of constructing a real-value scale from a given set of pairwise precedence information, that indicates which object precedes the other between two objects. Recently, there has been active research in the processing of orders. Cohen et al. [9] and Joachims [10] proposed a method to sort attributed objects associated with pairwise precedence information. Kamishima and Akaho [11] and Kazawa et al. [12] studied the learning problem from ordered object sets. Mannila and Meek [13] tried to establish the structure expressed by partial orders among a given set of orders. Sai et al. [14] proposed association rules between order variables. However, since we don't know of a clustering method for orders, we advocate this new technique in this paper.

2 Clustering Orders

In this section, we formalize a task of clustering orders. An order is defined as a sequence of objects that are sorted according to some property, such as size, preference, or price. An object x^a corresponds to an object, entity, or substance to be sorted. The universal object set, X^*, consists of all possible objects. The order is denoted by $O{=}x^1{\succ}x^2{\succ}\cdots{\succ}x^3$. The transitivity is preserved in the same order, i.e., if $x^1{\succ}x^2$ and $x^2{\succ}x^3$ then $x^1{\succ}x^3$. To express the order of two objects, $x^1{\succ}x^2$, we use the sentence "x^1 precedes x^2." $X_i \subseteq X^*$ denotes the object set that is composed of all the objects that appear in the order O_i. Let $|A|$ be the size of the set A, then $|X_i|$ is equal to the length of the order O_i. The order O_i is called a full-order if $X_i = X^*$, and is called a sub-order if $X_i \subset X^*$.

The task of clustering orders is as follows. A set of sample orders, $S = \{O_1\ , O_2\ , \ldots\ , O_{|S|}\}$ is given. Note that it is allowed to be $X_i \neq X_j (i \neq j)$. In addition, even if $x^1 \succ x^2$ in the order O_i, it may be $x^2 \succ x^1$ in the order O_j. The aim of clustering is to divide the S into a partition. The partition, $\pi = \{C_1\ , C_2\ , \ldots\ , C_{|\pi|}\}$, is a set of all clusters. Clusters are mutually disjoint and exhaustive, i.e., $C_i \cap C_j = \emptyset, \forall i, j, i \neq j$ and $S = C_1 \cup C_2 \cup \cdots \cup C_{|\pi|}$. Partitions are generated such that the orders in the same cluster are similar (internal cohesion), and those in the different clusters are dissimilar (external isolation). The similarity measure of orders and our clustering method is presented in the next section.

3 Methods

We modified a well-known clustering algorithm k-means so as to be able to deal with orders. We named this modified method the k-o'means.

3.1 Similarity between Two Orders

To measure the similarity between two orders, we adopted *Spearman's Rank Correlation*, which is denoted by "ρ" [15]. The ρ is the correlation between ranks of objects. The rank, $r(O\ ,x)$, is the cardinal number that indicates the position of the object x in the order O. For example, for the order $O{=}x^1{\succ}x^3{\succ}x^2$, the $r(O\ ,x^1){=}1$ and the $r(O\ ,x^2){=}3$. The ρ between two orders, O_1 and O_2, consisting of the same objects (i.e., $X_1 = X_2$) is defined as:

$$\rho = \frac{\sum_{x\in X_1}\big(r(O_1\ ,x)-\bar{r}_1\big)\big(r(O_2\ ,x)-\bar{r}_2\big)}{\sqrt{\sum_{x\in X_1}\big(r(O_1\ ,x)-\bar{r}_1\big)^2}\sqrt{\sum_{x\in X_1}\big(r(O_2\ ,x)-\bar{r}_2\big)^2}},$$

where $\bar{r}_i = (1/|X_1|)\sum_{x\in X_i} r(O_i\,,x)$. If no tie in rank is allowed, this can be calculated by the simple formula:

$$\rho = 1 - \frac{6\times\sum_{x\in X_1}\big(r(O_1\ ,x)-r(O_2\ ,x)\big)^2}{|X_1|^3-|X_1|}.$$

The ρ becomes 1 if two orders are coincident, and -1 if one order is a reverse of the other order. The ρ is designed for the two orders consisting of the same objects. In the case of the clustering task in Section 2, the object in one order may not appear in another order. We thus derived the rank correlation from the objects included in both orders, and ignored the rest of the objects. For example, two orders were given:

$$O_1 = x^1{\succ}x^3{\succ}x^4{\succ}x^6, \qquad O_2 = x^5{\succ}x^4{\succ}x^3{\succ}x^2{\succ}x^6.$$

From these orders, all the objects that were not included in the other orders were eliminated. The generated orders were:

$$O'_1 = x^3{\succ}x^4{\succ}x^6, \qquad O'_2 = x^4{\succ}x^3{\succ}x^6.$$

The ranks of objects in these orders were:

$$\begin{array}{lll} r(O'_1, x^3) = 1, & r(O'_1, x^4) = 2, & r(O'_1, x^6) = 3; \\ r(O'_2, x^3) = 2, & r(O'_2, x^4) = 1, & r(O'_2, x^6) = 3. \end{array}$$

Consequently, the ρ was

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

Note that if no common objects existed between the two orders, the $\rho = 0$ (i.e., no correlation). Over or under estimations in similarities will be caused by this heuristic

of ignoring objects. However, if X_i are randomly sampled from X^*, the expectation of the observed similarities will be the equivalent to true similarity. Therefore, such over or under estimations can be treated as same as the other types of noises.

To use for the clustering task, distance or dissimilarity is more suitable than similarity. We defined a dissimilarity between two orders based on the ρ:

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

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

We comment on the reason for adopting Spearman's ρ as a similarity measure. Similarities between orders are based on one of the following two quantities. The one is the differences between ranks of objects. The Spearman's ρ is an example of this type. The other is the number of discordant object pairs among all object pairs. Formally, an object pair, x^a and x^b, is discordant if $r(O_1,x^a) < r(O_1,x^b)$ and $r(O_2,x^a) > r(O_2,x^b)$, or vice versa. The Kendall's τ is an example of this type. Though the computational complexity of the former quantity is O($|X|$), that of the later is O($|X|^2$). The Spearman's ρ can be calculated faster, so we adopt this as a similarity.

3.2 Order Means

Before describing the algorithm of k-o'means, we would like to give the definition of an *order mean*.

In the case of k-means, the mean of the cluster C is derived such that

$$\bar{\mathbf{x}} = \arg\min_{\mathbf{x}^i} \sum_{\mathbf{x}^j \in C} \|\mathbf{x}^i - \mathbf{x}^j\|,$$

where $\mathbf{x}^i$ are data points, C is a cluster, and $\|\cdot\|$ is the L_2 norm. We extend this notion so as to fit for orders. That is to say, by employing Equation (1) as the loss function, we define an *order mean*, $\bar{O}$, as follows:

$$\bar{O} = \arg\min_{O_j} \sum_{O_i \in C} d(O_i\,,O_j). \tag{2}$$

Note that the order mean consists of all objects in all the orders in the C, that is $\bar{X} = \cup_{O_i \in C} X_i$.

If every order consists of all the objects in X^* (i.e., $X_i = X^*\,, \forall i$), the order means are derived by the Borda rule in the 18th century. The rule is equivalent to the following algorithm:

1) For each object x^a in the X^*, calculate the following value:

$$\tilde{r}^*(x^a) = \frac{1}{|C|} \sum_{O_i \in C} r(O_i\,,x^a). \tag{3}$$

2) By sorting objects according to the $\tilde{r}^*(x^a)$ in ascending order, the order mean of C can be derived.

Note that, if $\tilde{r}^*(x^a) = \tilde{r}^*(x^b)\,,\ x^a \neq x^b$, either $x^a \succ x^b$ or $x^b \succ x^a$ is allowed.

The proof of the optimality of this algorithm is as follows: First, we relax the limitations of ranks. While the *strict* ranks take one of $1, \ldots, |X|$, the relaxed ranks, $\tilde{r}(x)$, are real number and satisfy the condition:

$$\sum_{x \in X} \tilde{r}(x) = \sum_{i=1}^{|X|} i.$$

Clearly, strict ranks satisfy this condition. Since all the $|X_i|$ are equal, Equation (1) is proportional to the sum of the squared difference between the ranks of two orders. Therefore, the optimal relaxed ranks can be found by minimizing:

$$\sum_{O_i \in C} \sum_{x \in X^*} (\tilde{r}(x) - r(O_i\,, x))^2.$$

This is minimized at $\tilde{r}(x) = \tilde{r}^*(x)\,, \forall x$ in Equation (3). We then have to find the strict order, O_j, that minimizes the error. Minimizing the Equation (2) is equivalent to minimizing:

$$\begin{aligned} &\sum_{O_i \in C} \sum_{x \in X^*} \big(r(O_j\,, x) - r(O_i\,, x)\big)^2 \\ &= \sum_{O_i \in C} \sum_{x \in X^*} \big(r(O_i\,, x) - \tilde{r}^*(x))\big)^2 + |C| \sum_{x \in X^*} \big(\tilde{r}^*(x) - r(O_j\,, x)\big)^2. \end{aligned} \tag{4}$$

Since the first term is already minimized, the strict order mean $\bar{O}$ corresponds to the strict order O_j that minimizes the second term. We next show that the order mean $\bar{O}$ is equivalent to $\tilde{O}^*$, that is the order sorted according to $\tilde{r}^*(x)$. Assume that there is at least one discordant pair of objects between $\bar{O}$ and $\tilde{O}^*$. Formally, there exists an object pair, x^a and x^b, such that

$$\big(\tilde{r}^*(x^a) - \tilde{r}^*(x^b)\big)\big(r(\bar{O}\,, x^a) - r(\bar{O}\,, x^b)\big) < 0.$$

Let d_1 be the second term of the Equation (4) in this case. By swapping these objects only in the order mean, this error becomes d_2. Then,

$$d_1 - d_2 = -2|C|(\tilde{r}^*(x^a) - \tilde{r}^*(x^b))(r(\bar{O}\,, x^a) - r(\bar{O}\,, x^b)\big) > 0.$$

The fact that the error decreases by swapping objects is contradicted by the assumption that $\bar{O}$ is an order mean. Therefore, the order mean must not be discordant with $\tilde{O}^*$. Consequently, the above algorithm leads the order mean.

Unfortunately, it is practically a rare situation in which all orders consist of the same object set. If such conditions are not satisfied, the above algorithm cannot be applied. The calculation of the order mean is difficult in this case, since this is a discrete optimization problem. Instead of deriving a strictly optimal solution, we investigated several *ad-hoc* methods. We tried the methods in [11], such as the one compatible to Cohen's greedy methods [9]. As a result of the empirical comparisons, we found that the following method based on Thurstone's model achieved less error in Equation (2) than the others.

The Thurstone's law of comparative judgment (case V) [8] is the generative model of orders. This model assumes that the score $R(x^a)$ is assigned to each object x^a. The orders are derived by sorting according to the score. The scores follow the normal distribution, i.e., $R(x^a) \sim N(\mu_a, \sigma)$, where μ_a is the mean score of the object x^a and σ is a common constant standard deviation. Based on this model, the probability that object x^a precedes the x^b is

$$\begin{aligned}\Pr[x^a \succ x^b] &= \int_{-\infty}^{\infty} \phi(\frac{t-\mu_a}{\sigma}) \int_{-\infty}^{t} \phi(\frac{u-\mu_b}{\sigma}) du\, dt \\ &= \Phi\left(\frac{\mu_a - \mu_b}{\sqrt{2}\sigma}\right), \end{aligned} \tag{5}$$

where $\phi(\cdot)$ is a normal distribution density function, and $\Phi(\cdot)$ is a normal distribution function. Accordingly,

$$\mu_a - \mu_b = \sqrt{2}\sigma\Phi^{-1}\big(\Pr[x^a \succ y^b]\big).$$

The μ_a is rescaled into $\bar{\mu}_a$ by dividing $\sqrt{2}\sigma$, and the origin is set to the mean of $\bar{\mu}_1$, $\ldots \bar{\mu}_{|\bar{X}|}$, i.e., $\sum_{x^a \in \bar{X}} \bar{\mu}_a = 0$. To estimate these means, Thurstone's paired comparison method tries to minimize the function:

$$Q = \sum_{x^a \in \bar{X}} \sum_{x^b \in \bar{X}} \left(\Phi^{-1}\big(\Pr[x^a \succ x^b]\big) - (\bar{\mu}_a - \bar{\mu}_b) \right)^2.$$

To minimize the Q, it is differentiated by $\bar{\mu}_a$:

$$\frac{\partial Q}{\partial \bar{\mu}_a} = -2 \sum_{x^b \in \bar{X}} \left(\Phi^{-1}\big(\Pr[x^a \succ x^b]\big) - (\bar{\mu}_a - \bar{\mu}_b) \right) = 0.$$

This formula is derived for each $x^a = x^1, \ldots, x^{|\bar{X}|}$. By solving these linear equations, we obtain

$$\bar{\mu}_a = \frac{1}{|\bar{X}|} \sum_{x^b \in \bar{X}} \Phi^{-1}\big(\Pr[x^a \succ x^b]\big). \tag{6}$$

The order means can be derived by sorting according to these $\bar{\mu}_a$. To derive these values, we have to estimate the $\Pr[x^a \succ x^b]$ from the orders in the cluster, C. This probability is estimated by the following process. From the order O in the cluster, all the object pairs, (x^a, x^b), are extracted such that x^a precedes x^b in the order. For example, from the order $O = x^3 \succ x^1 \succ x^2$, three object pairs, (x^3, x^1),(x^3, x^2), and (x^1, x^2), are extracted. Such pairs are extracted from all $|C|$ orders in the cluster, and are collected into the set P_C. As the probability $\Pr[x^a \succ x^b]$, we adopted the following Bayesian estimator with Dirichlet prior distribution in order that the probability remains at non-zero:

$$\Pr[x^a \succ x^b] = \frac{|x^a, x^b| + 0.5}{|x^a, x^b| + |x^b, x^a| + 1},$$

where $|x^a, x^b|$ is the number of the object pairs, (x^a, x^b), in P_C.

Note that, the calculation time for one order mean is $\mathrm{O}(|\bar{X}|^2|C|)$, since the counting up time for deriving $\Pr[x^a \succ x^b]$ is $\mathrm{O}(\sum_i^{|C|} |X_i|^2) \leq \mathrm{O}(|\bar{X}|^2|C|)$, the estimation time of $\bar{\mu}_a$ for all $x^a \in \bar{X}$ is $\mathrm{O}(|\bar{X}|^2)$, and the sorting time is $\mathrm{O}(|\bar{X}| \log |\bar{X}|)$.

Finally, we comment on the method for estimating $\bar{\mu}_a$ based on the maximum likelihood principle (e.g., in [16]). Similar to the original k-means, our k-o'means depends on the the initial partition. To cancel this unstable factor, one has to select the best result among a number of trials. The maximum likelihood based method is a kind of gradient decent, so the resultant order also depends on a initial state. More trials are required to cancel this unstable factor. Moreover, since such an iterative method is time-consuming, we adopted the above non-iterative method.

Algorithm k**-o'means(**S**,** k**,** $maxIter$**)**
$S = \{O_1 , \ldots , O_{|S|}\}$: a set of sample orders
k: the number of clusters
$maxIter$: the limit of iteration times
1) initial partition: S is randomly partitioned into $\pi = \{C_1 , \ldots , C_k\}$, $\pi' := \pi$, $t := 0$
2) $t := t + 1$, if $t > maxIter$ then goto step 6
3) for each cluster $C_j \in \pi$, derive the order means $\bar{O}_j$ by the procedure in Section 3.2
4) for each order O_i in S, assign it to the cluster: $\arg\min_{C_j} d(\bar{O}_j , O_i)$
5) if $\pi = \pi'$ then goto step 6 else $\pi' := \pi$, goto step 2
6) output π

Fig. 1. The k-o'means algorithm

3.3 k-o'means

The k-o'means algorithm is the same as the well-known k-means algorithm, except for the notion of a mean and dissimilarity. The algorithm is shown in Figure 1. First, initial clusters are generated by randomly partitioning S. These clusters are improved by iteratively performing two steps: deriving the order mean for each cluster, and assigning each order to the nearest cluster. If the number of iterations exceeds the threshold or the partitions do not change, the algorithm stops and outputs the current partition. Note that, as the original k-means algorithm, the k-o'means cannot find the global optimal solution. Therefore, multiple partitions are derived by starting from different initial partitions, and then select π minimizing:

$$\sum_{C_i \in \pi} \sum_{O_j \in C_i} d(\bar{O}_i , O_j). \tag{7}$$

We here comment on the time complexity of this algorithm. First, the calculation time for k order means is $\mathrm{O}(|X^*|^2|S|)$, since one mean is calculated in $\mathrm{O}(|\bar{X}|^2|C|)$ time (see Section 3.2), $|C| \approx |S|/k$, and $|\bar{X}| \leq |X^*|$. Second, the time for the assignment of one order is $\mathrm{O}(|X_i|k)$, because dissimilarity is derived for each of the k clusters and one dissimilarity is calculated in $\mathrm{O}(|X_i|)$ time. Thus the total assignment time is $\mathrm{O}(\sum_i^{|S|} |X_i|k) \leq \mathrm{O}(|X^*||S|k)$. Since the number of iteration times is constant, the time

complexity of one iteration is equivalent to the total complexity. Consequently, the total complexity becomes $O(|X^*|^2|S| + |X^*||S|k)$. In terms of $|S|$ and k, this is equivalent to that of the original k-means. However, in terms of $|X^*|$, the complexity is quadratic.

4 Experiments

We applied our k-o'means algorithm to two types of data: artificially generated data and real preference survey data. In the former experiment, we compared our k-o'means and the traditional hierarchical clustering method using the dissimilarity of Equation (1). In addition, by applying the k-o'means to data which had various properties, we revealed the characteristics of this algorithm. In the latter experiment, we analyzed a questionnaire survey data on preferences in sushi (a kind of Japanese food). We used the k-o'means algorithm exploratory tools for analysis. Note that, in the experiments described below, Thurstone's method is used for deriving the order means since X_i are always proper subsets of X^*.

4.1 Evaluation Criteria

Before reporting the experimental results, the evaluation criteria for partitions will be described. In this section, the same object set was divided into two different partitions: π^* and $\hat{\pi}$. We present two criteria to measure the difference of $\hat{\pi}$ from π^*.

The first measure is called *purity*, and is widely used (e.g., in [17]). Assume that the objects in cluster $C_i^* \in \pi^*$ are classified into the true class labeled i. If all the objects in the $\hat{C}_i \in \hat{\pi}$ classified into the majority true class, the purity corresponds to the classification accuracy. Formally speaking, the purity is defined as:

$$\text{purity} = \frac{1}{|S|} \sum_{\hat{C}_i \in \hat{\pi}} \left(\max_{C_j^* \in \pi^*} |\hat{C}_i \cap C_j^*| \right). \tag{8}$$

The range of the purity is $[0\ ,\ 1]$, and becomes 1 if two partitions are identical.

Though this purity is widely used, its lower bound changes according as π^*, and the resulting scale normalization problem makes it difficult to use this as the basis for calculating the means of these criteria. Therefore, we introduce the second criteria, 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 [2]. The indicator function $I((x^i\,, x^j)\,, \pi)$ is 1 if an object pair $(x^i\,, x^j)$ are in the same cluster, and 0 if they are in different clusters. The contingency table is a 2×2 matrix consisting of elements a_{st}, that are the number of object pairs satisfying the condition $I((x^i\ , x^j)\ , \pi^*){=}s$ and $I((x^i\ , x^j)\ , \hat{\pi}){=}t$, among all the possible object pairs. RIL is defined as

$$\text{RIL} = \frac{\sum_{s=0}^{1} \sum_{t=0}^{1} \frac{a_{st}}{a_{\cdot\cdot}} \log_2 \frac{a_{\cdot t}}{a_{st}}}{\sum_{s=0}^{1} \frac{a_{s\cdot}}{a_{\cdot\cdot}} \log_2 \frac{a_{\cdot\cdot}}{a_{s\cdot}}}\ , \tag{9}$$

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

4.2 Experiments on Artificial Data

Data Generation Process. We applied the k-o'means to artificial data in order to compare it with traditional clustering methods and to analyze the method. Test data were generated in the following two steps: In the first step, we generated the k of order means. One random permutation (we called it a *pivot*) consisting of all objects in X^* was generated. The other $k-1$ means were generated by transforming this pivot. Two adjacent objects in the pivot were randomly selected and swapped. This swapping was repeated at specified times. By changing the number of swapping times, the inter-cluster closeness could be controlled.

In the second step, for each of the clusters, its constituent orders were generated. From the order mean, the $|X_i|$ of the objects were randomly selected. These objects were sorted so as to be concordant with the order mean. Namely, if x^a precedes x^b in the order mean, it should be the case in the generated orders. Again, two adjacent object pairs were randomly swapped. By changing the number of swapping times, the intra-cluster closeness could be controlled.

Table 1. Parameters of experimental data

1) the total number of objects: $\|X^*\| = 100$
2) the number of sample orders: $\|S\| = 1000$
3) the length of the orders: $\|X_i\| = 10$
4) the number of clusters: $\|\pi\| = \{2\ ,\ 5\ ,\ 10\ ,\ 50\}$
5) the swapping times of the order means: {a:∞, b:230000, c:120000}
6) the ratio of the minimum cluster size to the the maximum: $\{1/1\ ,\ 1/2\ ,\ 1/5\ ,\ 1/10\}$
7) the swapping times of the sample orders: {a:0 , b:30 , c:72}

The parameters of the data generator are summarized in Table 1. The parameters 1–3 are common for all the data. $|X^*|$ and $|S|$ are set so as to be roughly the same as those for the survey data in Section 4.3. If $|X_i|$ is too short, then the differences between orders cannot be tested. However, it is hard for respondents to sort too many objects. By considering these factors, we set the order length to 10. The parameter 4 was the number of clusters. It is difficult to partition if this number is large, since the sizes of the clusters then decrease. The parameter 5 was the swapping time in the first step of the data generation process. Three cases were examined. According to the simulation result, ρ means between a pivot and the other order means were 0.0, 0.1, and 0.3 in the cases of a, b, and c, respectively. Since the case a was the most separated, it was the easiest to partition. The parameter 6 controls the deviation of cluster sizes. If the sizes of the clusters are diverged, the relatively small clusters tend to be ignored. Thus, $1/10$ is the hardest to cluster. The last parameter is the swapping times in the second step of the data generation process. The means of ρ between a sample order and an order mean was 1.0, 0.715, and 0.442 in the cases of a, b, and c, respectively. ρ between two random orders

becomes larger than these values with probabilities 0.0, 0.01, and 0.10. Since the case a is the tightest, it was the easiest to partition.

The number of the total parameter combinations was $4{\times}3{\times}4{\times}3{=}144$. For each setting, we generated 100 sample sets. Below, we will show the means of purity and RIL of these sets.

Table 2. The means of purities and of RIL on artificial data

	purity	RIL
KOM	0.561 (0.3629)	0.705 (0.4105)
AVE	0.466 (0.2966)	0.910 (0.1679)
MIN	0.315 (0.2663)	0.999 (0.0014)
MAX	0.371 (0.2430)	0.994 (0.0112)

Comparison with Traditional Clustering Methods. We compared our k-o'means algorithm with the traditional hierarchical clustering methods: the minimum distance, maximum distance, and group average methods. Since data are not represented by attribute vectors, the original k-means cannot partition a set of sample orders. However, hierarchical methods can partition a set of sample orders by adopting Equation (1) as dissimilarities between any of the order pairs. We applied our k-o'means and three traditional algorithms to all 144 types of artificial data. The correct number of clusters were given as an algorithm's parameter. The means of purities and RIL are shown in Table 2. The symbols, KOM, AVE, MIN, and MAX indicate the means derived by the k-o'means, the group average, the minimum distance, and the maximum distance method, respectively. In parentheses, standard deviations are also shown. Clearly, our k-o'means is superior to the other three methods. According to the paired t-test between the k-o'means and each of the other three methods, the difference is significant, even at the significance level of 0.1%. We think that this advantage of the k-o'means is due to the following reasons. The dissimilarity between an order pair tends to be 1.0, since common objects are found infrequently. However, our k-o'means adopted a notion of an order mean. Since order means are derived from far more orders than two, common objects can be found far more frequently. In other words, while the traditional methods are based on only local information, the k-o'means can capture more global features of clusters. In addition, it is well known that a minimum or a maximum are not robust against the outliers, due to effects such as chaining [1].

Effects of Data Parameters. We next show the characteristics of the k-o'means according to the changes of the parameters 4–7 in Table 1. Table 3 shows the means of the purities and of the RIL on each of the data groups that are separated on the specific parameter. For example, in the column labeled "2" in Table 3(a), the means of purities the and of the RIL on 36 types of sample sets whose parameter 4 is 2, i.e., $|\pi| = 2$, are shown. Overall, parameters 4 and 7 affected partitioning, but the others did not. We will next comment on each of these parameters.

Table 3. The means of purities and of RIL on each of the order sets with the same parameters.

(a) parameter 4: the number of clusters

	2	5	10	50
purity	0.909	0.705	0.492	0.139
RIL	0.525	0.598	0.697	0.999

(b) parameter 5: inter-cluster closeness

	a:∞ (separated)	b:230000	c:120000 (close)
purity	0.566	0.566	0.552
RIL	0.695	0.698	0.723

(c) parameter 6: deviation of cluster sizes

	$1/1$ (equal)	$1/2$	$1/5$	$1/10$ (deviated)
purity	0.543	0.547	0.569	0.586
RIL	0.684	0.689	0.710	0.738

(d) parameter 7: intra-cluster tightness

	a:0 (tight)	b:30	c:72 (scattered)
purity	0.782	0.531	0.370
RIL	0.278	0.843	0.994

Parameter 4: The more the number of clusters increases, the poorer the performance becomes. If the number becomes 50, it is almost impossible to recover the original partition, even in the case of tightest intra-cluster closeness (i.e., the parameter 7 is 0). This can be explained as follows. The number of possible order means can be bounded $|X^*|!$. To choose one from these means, $\log_2(|X^*|!) \approx 525$ bits information is required. Roughly speaking, since the number of permutations of $|X_i|$ objects is $|X_i|!$, one order provides $\log_2(|X_i|!)$ bits information. In total, the orders in one cluster provide $|C| \log_2(|X_i|!) \approx (|S|/|\pi|) \log_2(|X_i|!) \approx 436$ bits information. Consequently, due to the shortage of information, fully precise order means weren't derived.
Parameter 5: Closeness between order means does not affect partitioning so much. We don't know the exact reason, but one possible explanation is that order means can easily be distinguished from each other since they are very long compared to sample orders.
Parameter 6: The original k-means tends to lead to poor partitions if the sizes of clusters vary, but the k-o'means do not. We think that this is also caused by the above high distinguishablity between order means.
Parameter 7: It is hard to find partitions consisting of clusters in which intra-cluster similarities are low. The sample orders are relatively short, so even a low level of noise affects the partitioning performance a great deal.

4.3 Experiments on Preference Survey Data

We applied our k-o'means to the questionnaire survey data of preference in *sushi*. Since notion of true clusters are not appropriate for such a real data, we use the k-o'means as exploratory analysis tools. We asked each respondent to sort 10 objects (i.e., sushi) according to his/her preference. Such a sensory survey is a very suitable area for analysis based on orders. The objects were randomly selected from 100 objects according to

Table 4. Summaries of partition on *sushi* data

Attributes of Clusters	C_1	C_2
$\|C\|$: the numbers of respondents	607	418
A1 : preference to heavy tasting sushi	0.4016	−0.1352
A2 : preference to sushi users infrequently eat	−0.6429	−0.6008
A3 : preference to expensive sushi	−0.4653	−0.0463
A4 : preference to sushi fewer shops supply	−0.4488	−0.2532

their probability distribution, based on menu data from 25 sushi shops found on the WWW. For each respondent, the objects were randomly permutated to cancel the effect of the display order to their responses. The total number of respondents was 1039. We eliminated the data obtained within a response time which was either too short (shorter than 2.5 minutes) or too long (longer than 20 minutes). Consequently, 1025 data was extracted.

We use k-o'means as an exploratory tool, and divided the data into two clusters. The summary of each cluster is shown in Table 4. The results given in the table were the best in terms of Equation (7) among 20 trials. The first row of the table shows the number of respondents grouped into each of the clusters. The C_1 is a major cluster. The subsequent four rows show the rank correlation between each order mean and the sorted object list according to the specific object attributes. For example, the fourth row presents the ρ between the order mean 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 next comment on each of the object attributes. Note that attributes A1 and A2 were derived from the questionnaire survey by the SD method, and the others from the menu data.

The attribute A1 (the second row) 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 tasting. The C_1 respondents preferred heavy tasting objects much more than the C_2 respondents. The attribute A2 (the third row) shows how frequently the respondent eats the object. The positive correlation indicates a preference for objects that the respondent infrequently eats. Both respondents of C_1 and C_2 prefer the objects they usually eat. No clear difference was observed between clusters. The attribute A3 (the fourth row) is the prices of the objects. These were regularized so as to cancel the effects of sushi styles (*hand-shaped* or *rolled*), and differences of price between shops. The positive correlation indicate a preference for cheap objects. While the C_1 respondents preferred expensive objects, the C_2 respondents did not. The attribute A4 (the fifth row) 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 C_2 is rather larger than that of C_1, the difference is not statistically significant. Roughly speaking, the members of the major group C_1 prefer more heavy tasting and expensive sushi than the members of the minor group C_2.

Selecting the number of clusters is worth a mention in passing. Since notion of the optimal number of clusters depend on application of the clustering result, the number cannot be decided in general. However, it can be considered that dividing uniformly

distributed data is invalid. To check this, we tested whether the nearest pair of order means could be distinguished or not. We denote the order means of clusters by $\bar{O}_1, \ldots, \bar{O}_{|\pi|}$, and that of the entire sample set S by $\bar{O}^*$. Let ρ_{ab} be the rank correlation between $\bar{O}_a$ and $\bar{O}_b$, and ρ^*_a be the rank correlation between $\bar{O}^*$ and $\bar{O}_a$. First, we found the $\bar{O}_\alpha$ and $\bar{O}_\beta$ such that the $\rho_{\alpha\beta}$ was the maximum among all pairs of order means. To test whether the closest (i.e. the most correlated) pair of clusters, C_α and C_β, should be merged or not, we performed statistical test of the difference between two correlation coefficients, ρ^*_α and ρ^*_β. In this case, the next statistics follows the Student t-distribution with the degree of freedom $|X^*|-3$:

$$t = (\rho^*_\alpha - \rho^*_\beta)\sqrt{\frac{(|X^*|-3)(1+\rho_{\alpha\beta})}{2(1-{\rho^*_\alpha}^2-{\rho^*_\beta}^2-{\rho_{\alpha\beta}}^2+2\rho^*_\alpha\rho^*_\beta\rho_{\alpha\beta})}}.$$

If the hypothesis $\rho^*_\alpha{=}\rho^*_\beta$ is not rejected, these two clusters should be merged and the number of cluster should be decreased. When partitioning the survey data into two clusters (i.e. $k = 2$), $t = 9.039$. At the significance level of 1%, it could be concluded that these clusters should not be merged. However, since $t = 1.695$ when $k = 3$, these clusters are invalid and the closest pair of clusters should be merged. Note that any criteria for selecting the number of clusters are not almighty as pointed in [19], since the optimality depends on clustered data and the aim of clustering. For example, when adopting the k-o'means for the purpose of caching [20], the best estimation accuracy was achieved when k was larger than two.

5 Conclusions

We developed a clustering technique for partitioning a set of sample orders. We showed that this method outperforms the traditional methods, and presented the characteristics of our method. By using this method, we analyzed a questionnaire survey data on preference in sushi.

The time complexity of the current algorithm is quadric in terms of $|X^*|$, thus this algorithm is difficult to deal with thousands of objects. We plan to develop the method to accommodate a much larger universal object set.

It is possible to extend our methods to hierarchical ones. We simply embedded the dissimilarity of Equation (1) and the order means to the traditional Ward method. However, the computation is very slow, since the new order mean cannot be derived from two order means of merged clusters. In addition, the performance was poor. The mean RIL on the artificial data over 10 trials is 0.894 (compare with Table 2). This is because fully precise order means cannot be derived if the sizes of clusters are small, and the sizes of clusters are very small in the early stage of the Ward method. More elaborate method would be required.

Acknowledgments. A part of this work is supported by the grant-in-aid for exploratory research (14658106) of the Japan society for the promotion of science.

References

1. Everitt, B.S.: Cluster Analysis. third edn. Edward Arnold (1993)
2. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall (1988)
3. Osgood, C.E., Suci, G.J., Tannenbaum, P.H.: The Measurement of Meaning. University of Illinois Press (1957)
4. Nakamori, Y.: Kansei data Kaiseki. Morikita Shuppan Co., Ltd. (2000) (in Japanese).
5. Cadez, I.V., Gaffney, S., Smyth, P.: A general probabilistic framework for clustering individuals and objects. In: Proc. of The 6th Int'l Conf. on Knowledge Discovery and Data Mining. (2000) 140–149
6. Ramoni, M., Sebastiani, P., Cohen, P.: Bayesian clustering by dynamics. Machine Learning **47** (2002) 91–121
7. Keogh, E.J., Pazzani, M.J.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In: Proc. of The 4th Int'l Conf. on Knowledge Discovery and Data Mining. (1998) 239–243
8. Thurstone, L.L.: A law of comparative judgment. Psychological Review **34** (1927) 273–286
9. Cohen, W.W., Schapire, R.E., Singer, Y.: Learning to order things. Journal of Artificial Intelligence Research **10** (1999) 243–270
10. Joachims, T.: Optimizing search engines using clickthrough data. In: Proc. of The 8th Int'l Conf. on Knowledge Discovery and Data Mining. (2002) 133–142
11. Kamishima, T., Akaho, S.: Learning from order examples. In: Proc. of The IEEE Int'l Conf. on Data Mining. (2002) 645–648
12. Kazawa, H., Hirao, T., Maeda, E.: Order SVM: A kernel method for order learning based on generalized order statistic. The IEICE Trans. on Information and Systems, pt. 2 **J86-D-II** (2003) 926–933 (in Japanese).
13. Mannila, H., Meek, C.: Global partial orders from sequential data. In: Proc. of The 6th Int'l Conf. on Knowledge Discovery and Data Mining. (2000) 161–168
14. Sai, Y., Yao, Y.Y., Zhong, N.: Data analysis and mining in ordered information tables. In: Proc. of The IEEE Int'l Conf. on Data Mining. (2001) 497–504
15. Kendall, M., Gibbons, J.D.: Rank Correlation Methods. fifth edn. Oxford University Press (1990)
16. Hohle, R.H.: An empirical evaluation and comparison of two models for discriminability scales. Journal of Mathematical Psychology **3** (1966) 173–183
17. Huang, Z.: Extensions to the k-means algorithm for clustering large data with categorical values. Journal of Data Mining and Knowledge Discovery **2** (1998) 283–304
18. Kamishima, T., Motoyoshi, F.: Learning from cluster examples. Machine Learning (2003) (in press).
19. Milligan, G.W., Cooper, M.C.: An examination of procedures for determining the number of clusters in a data set. Psychometrika **50** (1985) 159–179
20. Kamishima, T.: Nantonac collaborative filtering: Recommendation based on order responses. In: Proc. of The 9th Int'l Conf. on Knowledge Discovery and Data Mining. (2003)