T. Kamishima, H. Kazawa, and S. Akaho "A Survey and Empirical Comparison of Object Ranking Methods" In J. Fürnkranz and E. Hüllermeier, editors, Preference Learning, pp. 181–201. Springer (2010)
UNOFFICIAL VERSION

A Survey and Empirical Comparison of Object Ranking Methods

Toshihiro Kamishima[1], Hideto Kazawa[2], and Shotaro Akaho[1]

[1] 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
[2] Google Japan Inc., Cerulean Tower 6F, 26-1 Sakuragaoka-cho, Sibuya-ku, Tokyo, kazawa@google.com

Abstract. Ordered lists of objects are widely used as representational forms. Such ordered objects include Web search results or bestseller lists. In spite of their importance, methods of processing orders have received little attention. However, research concerning orders has recently become common; in particular, researchers have developed various methods for the task of *Object Ranking* to acquire functions for object sorting from example orders. Here, we give a unified view of these methods and compare their merits and demerits.

1 Introduction

We survey methods for learning to estimate orders, and empirically compare these methods. The term *Order* indicates a sorted sequence of objects according to some property. 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. In particular, several methods have been developed for learning functions used to sort objects from example orders. We call this task *Object Ranking* and emphasize its usefulness for sensory surveys[3], information retrieval, and decision making. We give a unified view of the object ranking task that are independently proposed and discuss the connection with the other types of tasks dealing with orders. We then show experimental results to reveal the pros and cons of these object ranking methods.

We formalize the object ranking task in Section 2. We survey methods in Section 3. Experimental results on artificial data and on real data are shown in Sections 4 and 5, respectively. We discuss and summarize the results in Section 6.

2 Orders and Object Ranking

This section shows basic notions regarding orders and formalizes the object ranking task.

[3] quantification of respondents' sensations or impressions

2.1 Basics about Orders

We begin by defining basic notations regarding orders. An object or entity to be sorted is denoted by $\mathbf{x}_j$. The universal object set, $\mathcal{X}^*$, consists of all possible objects. Each object $\mathbf{x}_j$ is represented by the attribute value vector $\mathbf{x}_j = [x_{j1}, x_{j2}, \ldots, x_{jk}]^\top$, where k is the number of attributes. An order is denoted by $O = \mathbf{x}_{j_a} \succ \mathbf{x}_{j_b} \succ \cdots \succ \mathbf{x}_{j_c}$. Note that the subscript j of $\mathbf{x}_j$ doesn't mean "The j-th object in this order," but that "The object is uniquely indexed by j in $\mathcal{X}^*$." An order $\mathbf{x}_1 \succ \mathbf{x}_2$ represents "$\mathbf{x}_1$ precedes $\mathbf{x}_2$." An object set $\mathcal{X}(O_i)$ or simply $\mathcal{X}_i$ is composed of all the objects in the order O_i. The length of O_i, i.e., $|\mathcal{X}_i|$, is shortly denoted by L_i. An order of all objects, i.e., O_i s.t. $\mathcal{X}(O_i) = \mathcal{X}^*$, is called a complete order; otherwise, the order is incomplete. Rank, $r(O_i, \mathbf{x}_j)$ or simply r_{ij}, is the cardinal number that indicates the position of the object $\mathbf{x}_j$ in the order O_i. For example, for $O_i{=}\mathbf{x}_1 \succ \mathbf{x}_3 \succ \mathbf{x}_2$, $r(O_i, \mathbf{x}_2)$ or r_{i2} is 3. For two orders, O_1 and O_2, consider an object pair $\mathbf{x}_a$ and $\mathbf{x}_b$ such that $\mathbf{x}_a, \mathbf{x}_b {\in} \mathcal{X}_1 {\cap} \mathcal{X}_2, a{\neq}b$. These two orders are concordant w.r.t. $\mathbf{x}_a$ and $\mathbf{x}_b$ 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 $\mathbf{x}_a, \mathbf{x}_b {\in} \mathcal{X}_1 {\cap} \mathcal{X}_2, a{\neq}b$.

We then describe the distance between two orders, O_1 and O_2, composed of the same sets of objects, i.e., $\mathcal{X}(O_1) = \mathcal{X}(O_2) \equiv \mathcal{X}$. Various kinds of distance for orders have been proposed [1]. *Spearman distance* $d_S(O_1, O_2)$ is widely used. It is defined as the sum of the squared differences between ranks:

$$d_S(O_1, O_2) = \sum_{\mathbf{x}_j \in \mathcal{X}} \big(r_{1j} - r_{2j}\big)^2. \tag{1}$$

By normalizing the range to be $[-1, 1]$, *Spearman's rank correlation* ρ is derived.

$$\rho = 1 - \frac{6 d_S(O_1, O_2)}{L^3 - L}, \tag{2}$$

where L is the length of orders, i.e., $L = |\mathcal{X}|$. This exactly equals the correlation coefficient between ranks of objects. The *Kendall distance* $d_K(O_1, O_2)$ is another widely used distance. Consider a set of object pairs, $\{(\mathbf{x}_a, \mathbf{x}_b) \in \mathcal{X} \times \mathcal{X}\}$, $a \neq b$, $\mathbf{x}_a, \mathbf{x}_b \in \mathcal{X}$, including either $(\mathbf{x}_a, \mathbf{x}_b)$ or $(\mathbf{x}_b, \mathbf{x}_a)$. The Kendall distance is defined as the number of discordant pairs between O_1 and O_2 w.r.t. $\mathbf{x}_a$ and $\mathbf{x}_b$. Formally,

$$d_K(O_1, O_2) {=} \frac{1}{2}\Big(M - \sum_{\{(\mathbf{x}_a, \mathbf{x}_b)\}} \mathrm{sgn}\big((r_{1a} {-} r_{1b})(r_{2a} {-} r_{2b})\big)\Big), \tag{3}$$

where $\mathrm{sgn}(x)$ is a sign function that takes 1 if $x{>}0$, 0 if $x{=}0$, and -1 otherwise. $M = (L-1)L/2$ is the number of all object pairs. By normalizing the range to

be $[-1, 1]$, *Kendall's rank correlation* τ is derived.

$$\begin{aligned}\tau =& 1 - \frac{2d_K(O_1, O_2)}{M} \\ =& \frac{1}{M} \sum_{\{(\mathbf{x}_a, \mathbf{x}_b)\}} \mathrm{sgn}\big((r_{1a} - r_{1b})(r_{2a} - r_{2b})\big). \end{aligned} \tag{4}$$

The computational costs for deriving ρ and τ are $O(L \log L)$ and $O(L^2)$, respectively. The values of τ and ρ are highly correlated, because the difference between two criteria is bounded by Daniels' inequality [2]:

$$-1 \leq \frac{3(L+2)}{L-2}\tau - \frac{2(L+1)}{L-2}\rho \leq 1. \tag{5}$$

Another inequality between d_K and d_S is Durbin-Stuart's inequality:

$$d_S \geq \frac{4}{3} d_K \left(1 + \frac{d_K}{L}\right).$$

In [1], you can find further description about other types of distance, such as Spearman's footrule, Cayley distance, and Ulam distance, and their characteristics.

Note that d_K is a metric, but d_S is not due to the violation of the triangular inequality condition. If two or more objects are tied, we give the same *midrank* to these objects [1]. For example, consider an order $\mathbf{x}_5 \succ \mathbf{x}_2 \sim \mathbf{x}_3$ ("$\sim$" denotes tie in rank), in which $\mathbf{x}_2$ and $\mathbf{x}_3$ are ranked at the 2nd or 3rd positions. In this case, the midrank 2.5 is assigned to both objects.

2.2 An Object Ranking Task

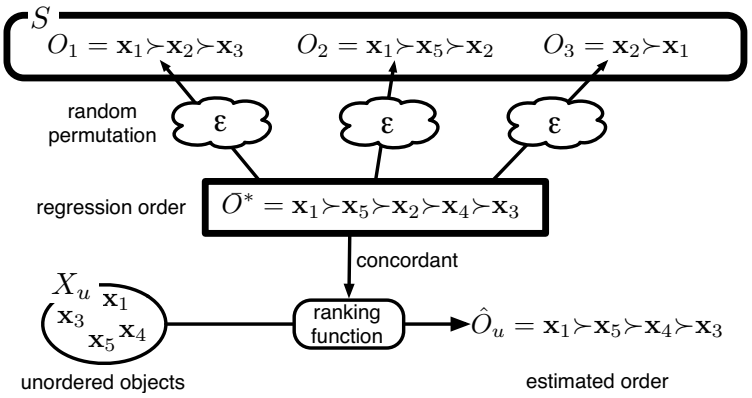


Fig. 1. The object ranking task

An *Object Ranking* task can be considered a regression or a fitting task whose target variables are orders. Further, input samples comprise not a set of vectors, but a set of orders, $S = \{O_1, \ldots, O_N\}$, where N is the number of samples. The regression curve corresponds to a regression order. Analogous to the case of a regression function, a regression order is estimated so as to be concordant not only with given sample orders in S, but also with orders that will be generated in future. This task differs from a regression in two ways. First, since the target variables are orders, the modeling methods of regression orders and errors are needed. A regression order is modeled by a ranking function, $\mathrm{ord}(\cdot)$, which represents a rule for sorting objects. Given an object set $\mathcal{X}_u$, a ranking function outputs the ideal order that consists of all objects in $\mathcal{X}_u$. Though errors of real values are modeled by an additive term of a random variable, errors in orders are modeled by a random permutation $\varepsilon(\cdot)$. That is to say, a sample order O_i is generated by $\varepsilon(\mathrm{ord}(\mathcal{X}_i))$. Second, since sample orders are generally incomplete, there may be objects not observed in given samples (e.g., x_4 in Figure 1). Such objects should be ranked under the assumption that the neighboring objects in the attribute space would be close in rank. Object ranking is also different from classification, because no ordinal relations should be defined between classes.

We say that a ranking function is *absolute* if outputs of the function are concordant with each other; otherwise, it is *relative*. That is to say, for any $\mathcal{X}_u \subseteq \mathcal{X}^*$, while an absolute ranking function outputs orders that are concordant with orders a regression order that consists of all objects in $\mathcal{X}^*$, a relative ranking function does not. Being absolute is also equivalent to the condition 3, "The independence of irrelevant alternatives," of the Arrow's impossibility theorem [3]. Concretely, given unordered sets $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ and $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_4\}$, an absolute ranking function outputs orders that are concordant w.r.t. $\mathbf{x}_1$ and $\mathbf{x}_2$ regardless of the existence of objects $\mathbf{x}_3$ or $\mathbf{x}_4$. However, a relative ranking function differently sorts $\mathbf{x}_1$ and $\mathbf{x}_2$ due to the existence of the other objects, e.g., $\mathbf{x}_1 \succ \mathbf{x}_2 \succ \mathbf{x}_3$ and $\mathbf{x}_2 \succ \mathbf{x}_4 \succ \mathbf{x}_1$. An absolute ranking function would be preferable in applications such as filtering or recommendation. For example, if one prefers an apple to an orange, he/she will always rank an apple higher than an orange when sorting any set of fruits according to degree of preference. One example application appropriate for relative ranking can be found in [4]. When summarizing multi-documents, after extracting important sentences, these sentences are ordered. In this case, appropriate order of sentences would be affected by the other extracted sentences.

Object ranking is closely related to a notion of a *Center of Orders* [1]; given sample orders S, center $\bar{O}$ is defined as the order that minimizes the sum of the distances $\sum_{O_i \in S} d(O_i, P(\bar{O}, \mathcal{X}_i))$. Note that $P(\bar{O}, \mathcal{X}_i)$ denotes the projection of $\bar{O}$ on the set $\mathcal{X}_i$, which is the order that consists of the objects in $\mathcal{X}_i$ and is concordant with $\bar{O}$. This notion is also referred by the other terms, such as aggregated ranking [5], in machine learning or information retrieval disciplines. We here adopt the term "center of orders" in statistics, because this would be the firstly given denotation. This center differs from the above regression order in the points that concordance only with *given* samples is considered and that

no description of objects, e.g., attribute values, is employed. The computation of centers is generally NP-hard, except for the cases such as employing Spearman's distance.

Object ranking is also related to *Ordinal Regression* [6, 7], which is a regression whose type of dependent variables is ordered categorical. Ordered categorical variables can take one of a finite set of predefined values, like categorical variables, and order these values additionally; for example, a domain of a variable is {"good","fair","poor"}. Ordered categories and orders are different in two points. First, while orders provide purely relative information, ordered categorical values additionally include absolute information. For example, while the category "good" means absolutely good, $\mathbf{x}_1 \succ \mathbf{x}_2$ means that $\mathbf{x}_1$ is relatively better than $\mathbf{x}_2$. Second, the number of grades that can be represented by ordered categorical variables is limited. Consider that there are four objects. Because at least two objects must be categorized into one of the three categories, {"good","fair","poor"}, the grades of these two objects are indistinguishable. However, orders can represent the differences of grades between any two objects. Because a task of object ranking is more complicated than ordinal regression, object ranking methods can be applicable to solve ordinal regression, but the converse is not true. But, generally speaking, since it is not efficient to solve too much complicated tasks, these two tasks should be carefully differentiated. For example, the computational cost of SVMs for object ranking (see Table 3) is about the square of $O(N^2\bar{L}^4)$, where $\bar{L}$ is the mean length of sample orders. The SVM specially designed for ordinal regression [8] demands less computational cost $O(N^2|\mathcal{Y}|^2)$, where $|\mathcal{Y}|$ is the number of grades.

3 Object Ranking Methods

We present five object ranking methods. Their abbreviations are given in parentheses in the section titles.

3.1 Cohen's method (Cohen)

Cohen's method [9] is designed to find the order $\hat{O}_u$ that maximizes

$$\sum_{\mathbf{x}_a \succ \mathbf{x}_b \in \hat{O}_u} \mathrm{P}[\mathbf{x}_a \succ \mathbf{x}_b | \mathbf{x}_a, \mathbf{x}_b], \tag{6}$$

where $\mathrm{P}[\mathbf{x}_a \succ \mathbf{x}_b | \mathbf{x}_a, \mathbf{x}_b]$ is the conditional probability given the attribute vectors of $\mathbf{x}_a$ and $\mathbf{x}_b$, and $\mathbf{x}_a \succ \mathbf{x}_b \in \hat{O}_u$ denotes all the ordered pairs concordant with $\hat{O}_u$. Note that O_u consists of all the objects in a given set of unordered objects, $\mathcal{X}_u$. Unfortunately, because the maximization of Equation (6) is known as a linear ordering problem [10], which is NP-hard, it is not tractable to find the optimal solution if $|\mathcal{X}_u|$ is large. Cohen et al. hence proposed a greedy algorithm that sequentially chooses the most-preceding object in Figure 2. They proposed a more elaborate approximation method too, but any strict or approximation algorithms to solve the linear ordering problem [10] can be applied for this optimization.

$\hat{O}_u \leftarrow \emptyset$
for all $\mathbf{x} \in \mathcal{X}_u$ **do**
  $\mathrm{score}(x) \leftarrow \sum_{\mathbf{x}' \in \mathcal{X}_u} \mathrm{P}[\mathbf{x} \succ \mathbf{x}' | \mathbf{x}, \mathbf{x}']$
end for
while $\mathcal{X}_u \neq \emptyset$ **do**
  $\mathbf{x}_{top} \leftarrow \arg\max_{\mathbf{x}} \mathrm{score}(\mathbf{x})$
  $\hat{O}_u \leftarrow \hat{O}_u \succ \mathbf{x}_{top}, \quad \mathcal{X}_u \leftarrow \mathcal{X}_u - \{\mathbf{x}_{top}\}$
  for all $\mathbf{x} \in \mathcal{X}_u$ **do**
    $\mathrm{score}(\mathbf{x}) \leftarrow \mathrm{score}(\mathbf{x}) - \mathrm{P}[\mathbf{x} \succ \mathbf{x}_{top} | \mathbf{x}, \mathbf{x}_{top}]$
  end for
end while
return $\hat{O}_u$

Fig. 2. Cohen's greedy sorting algorithm

$\mathrm{P}[\mathbf{x}_a \succ \mathbf{x}_b | \mathbf{x}_a, \mathbf{x}_b]$ is learned by Cohen et al.'s Hedge algorithm. The Hedge is an online algorithm, which is a variant of the Winnow [11]. Pairwise preference judgments are determined based on the linear combination of subordinate ordering functions. Given one preference feedback sample, a weight for an ordering function is increased according as the β parameter and the contribution of the ordering function to the concordance with the feedback. We set the β to 0.9; the attributes $\{\mathbf{x}_{jl}, -\mathbf{x}_{jl}\}_{l=1}^{k}$ are used as ordering functions in our experiments. To use the Hedge algorithm in off-line mode, the objects in S are iteratively given as feedback, and iterations are repeated until the loss becomes stationary. Their Hedge algorithm is designed so that it takes only ordinal information of attributes into account and discards the numerical values themselves. Hence, our experimental condition in which objects are represented by nominal or numerical attributes are rather disadvantageous to this method.

3.2 RankBoost (RB)

Freund et al. proposed *RankBoost* [12, 13], that is a boosting algorithm targeting orders. Inputs of the RankBoost are the feedback function $\Phi(\mathbf{x}_a, \mathbf{x}_b)$, which implies $\mathbf{x}_b \succ \mathbf{x}_a$ if $\Phi(\mathbf{x}_a, \mathbf{x}_b) > 0$, and a set of ranking features $f_l(\mathbf{x}_i)$, which conveys partial information about the target ordering. Given these inputs, RankBoost returns the final ranking $H(\mathbf{x}_i)$ that works as a score function. First, the initial distribution is calculated by $D_1(\mathbf{x}_a, \mathbf{x}_b) = \max(\Phi(\mathbf{x}_a, \mathbf{x}_b), 0)/Z_1$, where Z_1 is a normalization coefficient. Then, for each round $t = 1, \ldots, T$, the algorithm repeats the selection of weight α_t and weak learner $h_t(\mathbf{x})$, and the update of distribution by:

$$D_{t+1}(\mathbf{x}_a, \mathbf{x}_b) = \frac{1}{Z_t} D_t(\mathbf{x}_a, \mathbf{x}_b) \exp\big(\alpha_t (h_t(\mathbf{x}_a) - h_t(\mathbf{x}_b))\big).$$

Weak learners capture some information about target orders from ranking features, and output hypotheses such that $h_t(\mathbf{x}_b) > h_t(\mathbf{x}_a)$ implies $\mathbf{x}_b \succ \mathbf{x}_a$. α_t and h_t are selected so that the normalization factor Z_t is minimized. Once these weights

and weak learners are acquired, unseen objects, $\mathbf{x} \in \mathcal{X}^*$, are sorted in descending order of $H(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$.

In our experiment, as ranking features, we adopt all terms that appear in n-order polynomials of object attributes after attribute values are transformed so as to range in $[0, 1]$. Let $\Phi(\mathbf{x}_a, \mathbf{x}_b)$ be $2\mathrm{P}[\mathbf{x}_b \succ \mathbf{x}_a] - 1$. Weak learners are set to ranking features themselves, i.e., $h(\mathbf{x}) = f_l(\mathbf{x})$. Selection method of α_t and h_t is the third option in section 3.2 in [13]. The number of rounds, T is set to 100.

3.3 SVM-based methods: Order SVM (OSVM) and Herbrich's method (SVOR)

We show two SVM-based methods: OrderSVM and SVOR. In summary, the former is designed to discriminate whether or not a given object is ranked higher than j-th, while the latter judges which of two objects precedes the other.

Since this paper concerns not categorical but ordinal rank, this method may appear to be a groundless attempt to discriminate high-ranked objects from low-ranked ones. However, we showed that the probability of finding an object sorted above a fixed rank is concordant with the true score function. Thus, if a classifier will discriminate the top j objects from the rest, its discrimination function must be concordant to some extent with probability and therefore with the true score function. This observation leads to the use of SVM as the estimator of a score function in [14].

We first show *Order SVM* [14]. To enhance the reliability of this estimation, we proposed training multiple SVMs with different threshold ranks and sorting unseen objects using the average of those SVMs. Its learning is formulated as the following optimization problem:

$$\min_{\mathbf{w},\mathbf{v}_t,b_t} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{\lambda}{2}\sum_{t=1}^{L-1}\|\mathbf{v}_t\|^2 + C\sum_{t=1}^{L-1}\sum_{i=1}^{m}\sum_{j=1}^{L}\xi_i^j(t) \tag{7}$$

$$\text{s.t. } \mathrm{sgn}[j-t]((\mathbf{w}+\mathbf{v}_t)\cdot\mathbf{x}_i^j + b_t) \geq 1 - \xi_i^j(t),\ \xi_i^j(t) \geq 0, \quad \text{forall } i, j, t,$$

where $\mathbf{x}_i^j$ is the feature vector of the j-th ranked object in the i-th order, $\{\mathbf{x}_i^j\}_{i=1\ldots m}^{j=1\ldots L}$ are the training samples, and C and λ are hyperparameters. In our experiments, we set $C = 0.1$ and $\lambda = 1$ based on preparatory experiments on a small data set. The $\mathrm{sgn}[z]$ is 1 if $z \geq 0$; otherwise, -1. The SVM that discriminates the top t objects from the rest is $f_t(\mathbf{x}) = (\mathbf{w} + \mathbf{v}_t) \cdot \mathbf{x} + b_t$. Thus, the second regularizer $\sum_t \|\mathbf{v}_t\|^2$ makes all $f_t(\mathbf{x})$ agree on the predicted orders as much as possible. The order is predicted by sorting objects according to the score $\mathbf{w} \cdot \mathbf{x}$. The dual problem of Equation (7) is similar to that of standard SVMs, and any kernel function can be used instead of the inner products between feature vectors [14]. This method performs discriminations whether an object is ranked higher than t for each $t = 1, \ldots, L-1$. The number of failures in these discriminations is equivalent to the absolute difference between object ranks in the estimated order and sample order. These differences are then summed up

over all objects in orders. This sum can be considered as representing Spearman footrule [1], which is absolute distance between two orders. Therefore, this method aims minimizing the sum of distances in Spearman footrule between an estimated order and each sample order.

We refer to the other SVM-based method as *Support Vector Ordinal Regression* (SVOR) [15] since its formulation is very similar to standard SVMs and the work on it appears to be inspired by that of past ordinal regression works [16]. This method was independently developed as *Ranking SVM* by Joachims [17] and was proposed in [18].

SVOR discriminates correctly ordered pairs from incorrectly ordered pairs. In contrast to the Cohen method in which precedences are independently learned from pairs and there is no guarantee that transitivity holds among the learned preferences, SVOR uses a single score function for learning and thus avoids the intractability problem of the sorting process shown in [9].

SVOR's learning is formulated as the following optimization problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\sum_{1\le j<l\le L}\xi_i^{jl} \tag{8}$$
$$\text{s.t. } \mathbf{w}\cdot(\mathbf{x}_i^j - \mathbf{x}_i^l)\ge 1-\xi_i^{jl}, \quad \xi_i^{jl}\ge 0, \quad \text{for } {}^\forall i, j < l,$$

where the same notations as OSVM are used for $\mathbf{x}_i^j$, m, L, and C (In our experiments, $C = 1$). SVOR tries to find the direction $\mathbf{w}$ along which sample objects are ordered so that the narrowest separation between samples is maximal. The prediction of orders is done by sorting objects according to the score $\mathbf{w} \cdot \mathbf{x}$. As in the case of OSVM, the dual problem of Equation (8) can be written using only the inner products of $\mathbf{x}$; thus we can use any kernel function in SVOR, as well. This method is designed so that for each object pair in orders, the discrimination whether one object precedes the other is performed. The number of failures in these discriminations is equivalent to Kendall distance. Therefore, this method aims minimizing the sum of distances in Kendall distance between an estimated order and each sample order.

3.4 Expected Rank Regression (ERR)

We turn to our *Expected Rank Regression method* [19]. In this method, after expected ranks of objects are derived, the function to estimate these expected ranks is learned using a standard regression technique.

To derive expected ranks, assume that orders $O_i \in S$ are generated as follows: First, a complete sample order O_i^*, which cannot be observed and consists of all objects in $\mathcal{X}^*$, is generated. $|\mathcal{X}^*| - |O_i|$ objects are then selected uniformly at random, and these are eliminated from O_i^*; then, the O_i is observed. Under this assumption, a theoretical result in order statistics [20] shows that the conditional expectation of rank of the object $\mathbf{x}_j \in \mathcal{X}_i$ in the unobserved complete order O_i^* given the sample order O_i is

$$\mathrm{E}[r(O_i^*, \mathbf{x}_j)|O_i] \propto \frac{r(O_i, \mathbf{x}_j)}{|O_i| + 1}. \tag{9}$$

These expected ranks are calculated for all the objects in all the orders in a sample set S. Next, standard regression techniques are applied to derive weights of regression function, $f(\mathbf{x}_j)$, which predicts the above expectation of ranks. Samples for regression consist of the attribute vectors of objects, $\mathbf{x}_j$, and their corresponding expected ranks, $r(O_i, \mathbf{x}_j)/(|O_i|+1)$; thus the number of samples becomes $\sum_{O_i \in S} |\mathcal{X}(O_i)|$. Once weights of $f(\mathbf{x}_j)$ are learned, the order $\hat{O}_u$ can be estimated by sorting the objects $\mathbf{x}_j \in \mathcal{X}_u$ according to the corresponding values of $f(\mathbf{x}_j)$.

Below, we describe a rationale why this ERR works: We assume that sample orders are generated according to the Thurstone's model (case V) [21]. In this model, sample orders are generated by sorting objects $\mathbf{x} \in \mathcal{X}_u$ in the ascending order of the scores, $\mathbf{x}$, $f^*(\mathbf{x})$, that follow the following normal distribution:

$$f^*(\mathbf{x}) \sim \mathcal{N}(\mu(\mathbf{x}), \sigma^2), \tag{10}$$

where $\mu(\mathbf{x})$ and σ are mean and standard deviation, respectively. A complete regression order, $\bar{O}^*$, is determined by sorting all objects in $\mathcal{X}^*$ in the ascending order of the mean scores $\mu(\mathbf{x})$, and random permutation is caused by the additive noise in scores of objects; the above complete sample orders, $O_i^*, i = 1, \ldots, |S|$, are consequently generated. Next, we consider the probability that a pair of adjacent objects in $\bar{O}^*$ permutes. We assume that such probabilities are equal over all adjacent pairs, because there is no prior information that one pair is more frequently permuted than the other pair. This assumption makes the differences of the mean scores between these adjacent pairs, $|\mu(\mathbf{x}_i) - \mu(\mathbf{x}_j)|$, to be constant. Further, these mean can be replaced with rank in $\bar{O}^*$ without loss of generality, because an order derived based on scores is invariant for any linear transformation in scores. We then learn a function to predict rank $r(\hat{O}^*, \mathbf{x})$ for any object $\mathbf{x}$. Expectations of these ranks over random elimination of objects can be obtained by equation (9), and noise in scores follows normal distribution as in 10. Consequently, we can estimate ranks in $\bar{O}^*$ by applying standard regression to samples, $(\mathbf{x}_j, r(O_i, \mathbf{x}_j)/(|O_i|+1))$.

In our experiments, n-order polynomials are adopted as a class of regression functions. No regularization terms were adopted in regression.

4 Experiments on Artificial Data

To reveal the characteristics of object ranking methods, we applied these methods to artificial data.

4.1 Experimental Conditions

Artificial data were generated in three steps. First, we generated two types of vectors: numerical (`num`) and binary (`bin`). Each numerical vector consists of $5(\equiv k)$ attributes, which follow the normal distribution, $N(0, 1)$. Binary vectors are composed of $15(\equiv k)$ attributes, and are randomly generated so that every

object is represented by different value vectors. Note that, when the methods designed for numeric attributes, binary values are converted to the number, 0 or 1. Second, true regression orders are determined based on these attribute values. Objects are sorted according to the values of the function:

$$\text{utility}(\mathbf{x}_j)=(1+\sum_{l=1}^{k} w_l x_{jl})^{\text{dim}},$$

where w_l are random weights that follow the normal distribution, $\mathcal{N}(0,1)$. We tested two settings: linear (dim=1) and non-linear (dim=2 or 3), denoted by `li` and `nl`, respectively. Finally, N sample orders $O_i \in S$ were generated by randomly eliminating objects from the true regression orders.

As an error measure, we used Spearman's ρ (Equation (2)) between the estimated order and the above true regression order. If ρ is 1, the two orders are completely concordant; if it is -1, one order is the reverse of the other. We will show the means of ρ over a 10-fold cross validation for 10 different weight sets of utility(x_j). Regardless of the length of training orders, the size of the unordered set, $|\mathcal{X}_u|$, is set to 10, because errors cannot be precisely measured if orders are too short.

4.2 Experimental Results

As the basic experimental condition, we chose N=300, L_i=5, and $|\mathcal{X}^*|$=1000. Under this condition, the probability that one object in $\mathcal{X}_u$ is unobserved in the training samples seems rather low (25.8%). However, the probability that the object pairs in $\mathcal{X}_u$ become unobserved, which is intrinsic to ordinal relations, is fully high (99.5%). Therefore, we consider that this data set is well suited to evaluate the generalization ability. Note that algorithm parameter settings described in Section 3 are tuned for this basic condition under a noisy condition described later. By default, we used this basic condition in the following experiments. The other settings were: For Cohen, we adopt their Hedge and their greedy search algorithm. Note that we also applied the exhaustive search to derive the optimal orders that maximizes equation (6), but the results were rather inferior to that by the greedy search in Figure 2. For RB, ranking features are all terms of a second-order polynomial. For SVOR and OSVM, Gaussian kernels with $\sigma = 1$ were used. For ERR, the second-order polynomials was used as class of regression functions.

Table 1(a) shows the means of ρ under this basic setting. Each row corresponds to each of the four data sets described in Section 4.1, and each column corresponds to each method in Section 3. The rank of each method is shown in brackets. Except for the difference between RB and SVOR of `nl`/`bin`, the difference between each method and the next-ranked one is statistically significant at the level of 1% when using a paired t-test and a Bonferroni multiple comparison.

Defects of the Cohen would be due to the fact that only the ordinal information of attributes is considered, as described in Section 3.1. The ERR methods were inferior in `bin` cases, but were superior in `num` cases. The performance with

Table 1. Basic Results: $|\mathcal{X}^*|$=1000, L_i=10, N=300

(a) under noiseless conditions

	Cohen	RB	SVOR	OSVM	ERR
li/num	0.860 [5]	0.959 [2]	0.914 [3]	0.886 [4]	0.982 [1]
li/bin	0.966 [2]	0.978 [1]	0.885 [4]	0.868 [5]	0.895 [3]
nl/num	0.682 [5]	0.763 [4]	0.911 [2]	0.878 [3]	0.935 [1]
nl/bin	0.786 [5]	0.875 [1]	0.866 [2]	0.842 [3]	0.830 [4]

(b) under noisy conditions

	Cohen	RB	SVOR	OSVM	ERR
nl/num	0.652 [5]	0.719 [4]	0.818 [1]	0.797 [3]	0.813 [2]
nl/bin	0.764 [5]	0.842 [1]	0.817 [2]	0.809 [3]	0.796 [4]

nl/bin data was unstable because the weights of a regression function have to be determined based on two points, 0 and 1. The SVM-based method could avoid this problem by adopting the regularization property. The two SVM-based methods, OSVM and SVOR, also bear a resemblance to each other. The RB was rather inferior for the nl case, but it could be improved by increasing the number of rounds T. For example, when we tried the number of iterations $T = 1000$ for basic data under a noisy condition in Table 1(b), the ρ improved from 0.720 to 0.765. However, it was too slow compared with other methods, so we had to set $T = 100$ when performing our experiments.

Table 1(a) shows the results under a noiseless condition; that is to say, all the sample orders are perfectly concordant with the corresponding regression order. To test the robustness of the methods against the noise in the orders, we permuted two randomly selected pairs of adjacent objects in the original sample orders. By changing the number of times that objects are permuted, the noise level could be controlled. The order noise level is measured by the probability that the ρ between the original order and the permuted one is smaller than the ρ between the original order and a random one. This probability can be computed by using the statistical property of Spearman's ρ. We generated four types of data whose noise levels were 0%∼10%. Note that the 0% level noise is equivalent to the noiseless case.

Figure 3 shows the means of ρ in accordance with the order noise level for the nl/num data. In accordance with the increase of noise, the empirical ρ (between the estimated order and the permuted sample order) drastically became worse, whereas true ρ (between the estimated order and the noiseless order that cannot be observed in non-artificial data) did not decrease to a significant degree. For example, at the 10% noise level, the empirical ρ by the ERR for the nl/num data is 0.409, while the true ρ is 0.904. We then examined the robustness of these methods against noise in attribute values. For the numerical attributes, the α% level of noise is obtained by multiplying the true values by the random factors that follow $N(1, \alpha/100)$. Note that sample orders were calculated based on noiseless attribute values. Figure 4 shows the means of ρ in accordance with the level

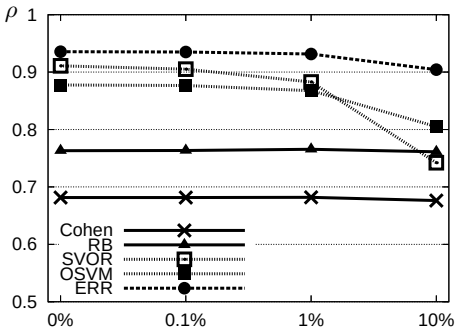


Fig. 3. Variation in the order noise

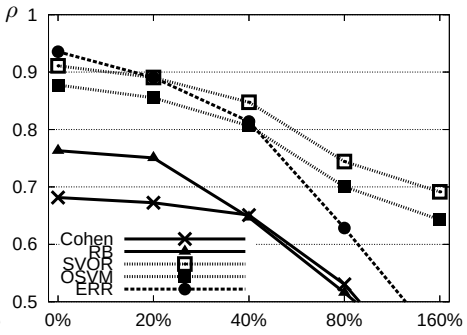
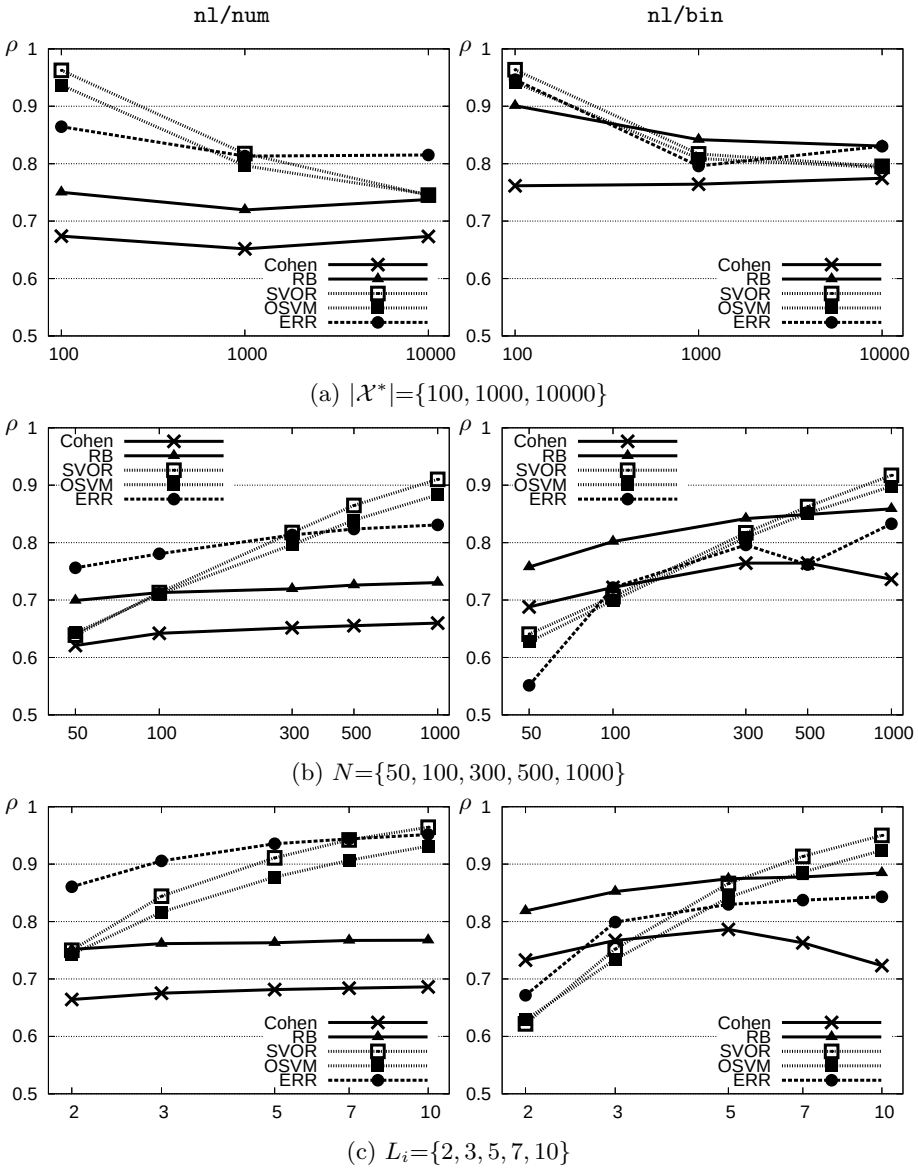


Fig. 4. Variation in the attribute noise

of attribute noise for the `nl/num` data. We generated five types of data whose α values were set to 0%∼160%. The results shown in Figures 3 and 4 indicate a clear contrast. The SVM-based methods were robust against attribute noise, but not against order noise. Conversely, the other methods were robust against order noise, but not against attribute noise. This could be explained as follows: The SVM-based methods are sensitive to order noise because the exchanged ordered pairs tend to become support vectors, while perturbation of attribute values does not affect the support vectors as much. Inversely, the non-SVM-based methods can learn correctly if correct orders constitute the majority of the sample orders; thus, these methods are robust against order noise. However, any perturbation in attribute values affects their performance.

The noiseless setting of Table 1(a) is unrealistic, because real data generally include noise. We therefore investigated the behavior of object ranking methods under more realistic *noisy* conditions. According to the above results, the relative superiority of the prediction performance among the methods heavily depended on types of noise. That is to say, while the non-SVM-based methods were superior for data with more order noises, the SVM-based ones were superior for data with more attribute noises. Instead of comparing the relative superiority of methods, we investigated the patterns of the changes of the relative predictive performance in accordance with the variation of data properties. To check these, noise levels were selected so that ERR and SVOR gave roughly equal performance. In both the `nl/num` and the `nl/bin` data sets, order noise levels were set to 1%. While the attribute noise level of the `nl/num` was 40%, binary values were flipped with a probability of 1% for the `nl/bin`. We however used noiseless data when testing the variation in the predictive performance according to the length of sample orders (Figure 5(c)), because the shorter sample orders were more seriously influenced by order noise. Along with fixing the algorithm parameter settings, we tested the changes of the prediction performance according to variation in the number of objects $|\mathcal{X}^*|$, the number of sample orders N, and the length of orders L_i.



(a) $|\mathcal{X}^*|$=$\{100, 1000, 10000\}$

(b) N=$\{50, 100, 300, 500, 1000\}$

(c) L_i=$\{2, 3, 5, 7, 10\}$

Fig. 5. Variation in the number of objects $|\mathcal{X}^*|$, the number of sample orders N, and the length of sample orders L_i

Table 1(b) shows the results under this noisy condition for the basic data. Except between SVOR and ERR of `nl/num`, the difference between each method and the next-ranked one is statistically significant at the level of 1% when using a paired t-test and a Bonferroni multiple comparison. Figure 5 shows the means of ρ in accordance with the variations in the other properties of samples sets. The results for the `nl/num` and the `nl/bin` data are shown in each column. Rows (a), (b), and (c) show results when the number of objects $|\mathcal{X}^*|$, the number of sample orders N, and the length of orders L_i were varied, respectively.

In terms of Figure 5(a), the probability that an object in test orders has not been included in training samples decreases in accordance with the increase of $|\mathcal{X}^*|$; accordingly, a greater generalization ability is required. SVM-based methods were better if $|\mathcal{X}^*|$ was small, but their performance dropped for larger $|\mathcal{X}^*|$. Adoption of soft-margin parameter C tuning for $|\mathcal{X}^*|$ was required in order for the SVM-based methods to work well. The non-SVM-based methods results were rather flat. This would be because the number of model parameters to determine is fewer in these methods than in the SVM-based ones.

Turning to Figure 5(b) and (c), the Cohen method performed more poorly for the larger L_i. This would be because the paired comparison model used in the Cohen method assumes independence among ordered pairs. For small N or L_i, the performance of the SVM-based methods was inferior to those of the others. However, the performance was improved in accordance with the increase of N or L_i. This might be because the SVM-based methods are over-fitted when the sample set is so small that the learned functions are not sparse. We also expected that this observation arises from the strength of model biases. Hence, we further checked the performance by changing the parameters of the methods, but we could not find a simple relation between the number of parameters to learn and the number of observed samples.

5 Experiments Using Real Data

We applied the methods described in Section 3 to real data from the following questionnaire surveys. The first data set was a survey of preferences in sushi (Japanese food), and is denoted as SUSHI [4] [22, 19]. In this data set, $N = 500$, $L_i = 10$, and $|\mathcal{X}^*| = 100$. Objects are represented by 12 binary and 4 numerical attributes. By using the k-o'means clustering method [23, 24], we generated two sample orders whose ordinal variances were broad and narrow. The probabilities that objects were selected in O_i were not uniform, as assumed in an ERR method. Objects were selected independently with probabilities that range from 3.2% to 0.13% from $\mathcal{X}^*$. The second data set was a questionnaire survey of news articles sorted according to their significance, and is denoted as NEWS. These news articles were obtained from "CD-Mainichi-Newspapers 2003." In this data set, $N = 4000$, $L_i = 7$, and $|\mathcal{X}^*| = 11872$. The variance among sample orders was slightly broader than the tight SUSHI data. Articles were represented by

[4] This data set can be downloaded from `http://www.kamishima.net/sushi/`

keyword and document frequencies, and these 22297 elements were compressed to 20 attributes using latent semantic indexing (a contribution ratio is about 0.9). Additionally, we used 8 binary attributes to represent article categories. For both data sets, the N or L_i were varied by eliminating sample orders or objects. Orders became difficult to estimate as N and/or L_i decreased. Errors were measured by the empirical ρ between the sample order and the estimated one. The algorithm's parameters were set to the practical setting in Section 4.2. Ideally, these parameters should be tuned by using cross validation, but some methods were too slow to perform such fine tuning. Therefore, these experiments reveal not the relative superiority among methods but the changes in performance along with the variation in N and/or L_i.

Table 2. Results on real data sets

	N:L_i	Cohen	RB	SVOR	OSVM	ERR
SUSHI	500:10(b)	0.364 [5]	0.384 [4]	0.393 [3]	0.400 [1]	0.397 [2]
	100:5(b)	0.354 [2]	0.356 [1]	0.284 [4]	0.315 [3]	0.271 [5]
	100:2(b)	0.337 [1]	0.281 [2]	0.115 [4]	0.208 [3]	0.010 [5]
	500:10(n)	0.543 [5]	0.583 [4]	0.719 [1]	0.708 [2]	0.705 [3]
	100:5(n)	0.548 [5]	0.612 [4]	0.646 [2]	0.655 [1]	0.617 [3]
	100:2(n)	0.577 [1]	0.542 [2]	0.522 [4]	0.540 [3]	0.421 [5]
NEWS	4000:7	−0.008 [5]	0.350 [3]	0.244 [4]	0.366 [2]	0.386 [1]
	1000:5	−0.009 [5]	0.340 [3]	0.362 [1]	0.353 [2]	0.312 [4]
	1000:2	−0.009 [5]	0.338 [3]	0.349 [1]	0.344 [2]	0.149 [4]

In Table 2, we show the means of ρ. The column labeled N:L_i represents the number and the length of sample orders, and the letter, "b" or "n" denotes the types of variance, broad or narrow, respectively. In the SUSHI case, the differences among methods were less clear than those in artificial data. Though we expected that the SVM would work well for a tight data set, the variance in sample orders was less affected. We could say that this is due to the fitness of the SUSHI data to a linear model; we observed that the other method worked well when using a linear model. ERR showed good performance for large N or L_i, but poorer results for small N or L_i. This is due to too complicated model for regression, because the mean ρ increased to 0.249 by adopting a simpler linear regression model for 100:2:(b). Inversely, RB was better for small N or L_i. This is due to the setting of $T = 100$, the number of rounds. When $T = 300$, we observed that performance for large N or L_i improved, but it was depressed for small N or L_i because of over-fitting. In the case of NEWS, sample orders were tight, but the correlation between sample orders and attributes were remarkably weak. Thus, all methods performed poorly. For such weak attributes, Cohen performed very poorly, even though we tuned β parameters. Again, ERR was better for large N or L_i, but was poorer for small N or L_i. However, this was due to the model complexity as in the above SUSHI case. In summary, the performances of

four methods other than Cohen were roughly equal, when dealing with these real data.

6 Discussion and Conclusions

We first discuss the relation between object ranking methods and the probabilistic generative models for orders (see appendix A). These models can be categorized as four types [1, 25]:

- *Thurstonian*: Objects are sorted according to the corresponding score.
- *Paired comparison*: Objects are compared in pairwise, and all objects are sorted based on these comparison results.
- *Distance-based*: Orders are generated with the probability that is determined based on the distance from a modal order.
- *Multistage*: Objects are sequentially arranged top to end.

Object ranking methods are commonly designed by incorporating a way to deal with attributes into these models. Error or loss in orders are designed based on these generative models. In Cohen, because the precedence between object pairs is firstly determined based on the learned model, we consider that this method adopts a paired comparison model. Regarding RB and SVM methods, a modal order is determined by sorting the outputs of a score function. Loss functions between a modal order and sample orders are defined based on the discordance between them. Therefore, these methods can be considered to be related to the distance-based model. While RB and SVOR adopt Kendall distance, OSVM adopts Spearman footrule. Finally, ERR is based on the Thurstonian model as described in section 3.4.

Table 3. Computational complexities

	Cohen	RB	SVOR	OSVM	ERR
Learn	$N\bar{L}^2k$	$N\bar{L}^2k$	$N^2\bar{L}^4k$	$N^2\bar{L}^4k$	$N\bar{L}k^2$
Sort	L^2	$L\log L$	$L\log L$	$L\log L$	$L\log L$

We next summarize computational complexities of learning and soring time in the first and second rows of Table 3. We assume that the number of ordered pairs and of objects in S are approximated by $N\bar{L}^2$ and $N\bar{L}$, respectively, where $\bar{L}$ is the mean length of the sample orders. The SVM's learning time is assumed to be quadratic in the number of training samples. The learning time of Cohen's Hedge algorithm or the RB is linear in terms of the $N\bar{L}^2$, if the number of iteration T is fixed. However, if T is adaptively chosen according to $N\bar{L}^2$, their time complexity becomes super-linear. In terms of the number of attributes k, the SVM-based methods depend on the number of non-zero attribute values; thus it is practically sub-linear. Standard regression used in ERR can be computed

faster, if attributes are sparse. Generally, in practical use, the learning time of the SVM-based methods is slow, that of Cohen and RB is intermediate, and that of ERR are much faster. In terms of time for sorting of $\mathbf{x}_j \in \mathcal{X}$, the Cohen greedy requires $O(L^2)$ while the others perform more quickly, $O(L \log L)$.

We finally summarize the pros and cons of each method. Our new ERR method was practically the fastest without sacrificing its prediction performance. This quickness make it to try many parameter settings in relatively short times. Even for the result where ERR were poor, e.g., SUSHI:100:2(b), we observed that it could be improved by re-tuning. In this method, the uniform distribution of the object observation is assumed, but SUSHI result demonstrated the robustness against the violation of this assumption to an extent. However, this method requires quadric time in terms of k, if attribute vectors are dense.

The most prominent merit of Cohen is to be an on-line method. For on-line learning purposes, the other methods cannot be used. Though the Cohen performed rather poorly in our experiments, this is because the Hedge algorithm is designed to take into account as its attributes only ordinal information. We observed that the performance could be improved by using other classifier, such as the naive Bayes, instead of the Hedge, because this method was designed to deal with categorical or numerical attributes. Further, our experiments were absolute ranking task (see section 2.2), but the Cohen acquires relative ranking function, because this method adopts a paired comparison model.

The unique property of the RB is rich options of weak learners. Because of this property, various types of attributes can be used. If objects are represented by vectors whose attribute types are mixtures of ordinal and numerical/categorical, the other algorithm cannot be used. Our experimental results of RB were rather inferior, but we observed that this could be considerably improved by adaptively increasing T. Due to too slow convergence, we had to stop iterations after the end of the drastic error drops at the beginning stage. However, it takes as same or more computation time as the SVM-based methods until complete convergence, and it should be also noted that too large T would cause over-fitting.

Like a standard SVM, the SVOR and OSVM are advantageous if k is large. The our experimental results demonstrated that the two SVM-based methods and the others are robust against different types of noise. Hence, for data in which orders are permuted, the non-SVM-based methods are preferable, while for data whose attributes are disturbed, the SVM-based methods are preferable. The demerit of the SVM-based methods are slowness. The learning complexity of the two SVM-based methods is the same, but the OSVM is practically slower. However, it was more robust against order noise than SVOR.

A Probabilistic Generative Models for Orders

We briefly summarize the probabilistic generative models for orders. Readers that needs further information should refer a textbook [1] or a survey paper [25]. As described before, these models can be categorized into four types: Thurstonian, paired comparison, distance-based, and multistage. All these are generative

model for complete orders, in which all objects are included. We sequentially introduce these four types.

As seen in the name *Thurstonian*, this type of models are firstly proposed by Thurstone [21] and are also called by order statistics models. In this model, objects, $\mathbf{x}_i \in \mathcal{X}^*$, are sorted according to their corresponding scores. This score is a real value probabilistic function that is defined as

$$\mathrm{score}(\mathbf{x}_i) = \mu(\mathbf{x}_i) + \epsilon(\mathbf{x}_i),$$

where $\mu(\mathbf{x}_i)$ is the mean score for the object $\mathbf{x}_i$ and $\epsilon(\mathbf{x}_i)$ follows the distribution with zero mean. In original Thursone's model, the normal distribution is used as $\epsilon(\mathbf{x}_i)$. Especially, the "case V", in which the variance of the normal distribution is constant over all objects, is widely used. In this case, the probability the object $\mathbf{x}_i$ precedes the $\mathbf{x}_j$ is simply computed by [26]:

$$\mathrm{P}[\mathbf{x}_i \succ \mathbf{x}_j] = \mathrm{P}[\mathrm{score}(\mathbf{x}_i) > \mathrm{score}(\mathbf{x}_j)] = \Phi\left(\frac{\mu(\mathbf{x}_i) - \mu(\mathbf{x}_j)}{\sqrt{2}\sigma}\right),$$

where $\Phi(\cdot)$ is the normal c.d.f. and σ^2 is its variance.

In a *paired comparison model*, which object precedes the other is determined for each pair of objects. Accordingly, $L(L-1)/2$ ordered pairs are generated, where L is the total number of objects, i.e. $L = |\mathcal{X}^*|$. If this set of ordered pairs is cyclic, i.e., $\mathbf{x}_i \succ \mathbf{x}_j$, $\mathbf{x}_j \succ \mathbf{x}_k$, and $\mathbf{x}_k \succ \mathbf{x}_i$, then this set is discarded, and all pairs are re-generated; otherwise, a total order is uniquely determined. The saturated model having $L(L-1)/2$ parameters, which represent the probabilities that one object precedes the other, is called by *Babington Smith model*. Babington Smith firstly showed the moments of this model in [27]. After that, some models with fewer parameters have been proposed. The next model having L parameters is called *Bradley-Terry model* [28]:

$$\mathrm{P}[\mathbf{x}_i \succ \mathbf{x}_j] = \frac{f(\mathbf{x}_i)}{f(\mathbf{x}_i) + f(\mathbf{x}_j)},$$

where $f(\cdot)$ is positive real function.

In *distance-based model*, orders, O, follows the following distribution

$$\mathrm{P}[O] = \frac{1}{Z(\lambda)} \exp\Big(-\lambda d(O_0, O)\Big),$$

where λ is a concentration parameter, which is non-negative real value, the O_0 is a modal ranking, at which this distribution is peaked, $d(\cdot,\cdot)$ denotes the distance between orders, and $Z(\lambda)$ is a normalization factor. Especially, this model is called *Mallows ϕ-model* and *Mallows θ-model* if Kendall and Spearman distance are adopted, respectively. This is because these are the special cases of the *Mallows model* [29], which is a kind of a paired comparison model. If i-th and j-th ranked objects in a generated order are $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, Mallows model is defined as

$$\mathrm{P}[\mathbf{x}_i \succ \mathbf{x}_j] = \frac{\theta^{i-j}\phi^{-1}}{\theta^{i-j}\phi^{-1} + \theta^{j-i}\phi},$$

where θ and ϕ are positive real parameters. If $\theta = 1$ (resp. $\phi = 1$), this model becomes Mallows ϕ-model (resp. Mallows θ-model).

Finally, in *multistage model*, objects are sequentially arranged top to end. *Plackett-Luce model* [30], which is a kind of a multistage model, generates an order, $O = \mathbf{x}_{i_1} \succ \mathbf{x}_{i_2} \succ \cdots \mathbf{x}_{i_L}$, with the following procedure. The top ranked object, $\mathbf{x}_{i_1}$, is chosen with the probability:

$$\frac{f(\mathbf{x}_{i_1})}{\sum_{\mathbf{x} \in \mathcal{X}^*} f(\mathbf{x})},$$

the second ranked object, $\mathbf{x}_{i_2}$, is chosen with the probability:

$$\frac{f(\mathbf{x}_{i_2})}{\sum_{\mathbf{x} \in \mathcal{X}^*, \mathbf{x} \neq \mathbf{x}_{i_1}} f(\mathbf{x})},$$

and these procedures are iterated $L - 1$ times. Orders generated by this model satisfies *Luce's choice axiom*: Roughly speaking, the probability that an object is top-ranked equals to the product of the probability that the object is top-ranked in any subset and the probability that the subset contains the object. More formally, $\mathrm{P}_{\mathcal{X}}[\mathbf{x}]$ denotes the probability that an object $\mathbf{x}$ is top ranked among the object set, $\mathcal{X}$, and $\mathrm{P}_{\mathcal{X}}[\mathcal{X}']$, $\mathcal{X}' \subset \mathcal{X}$, denotes the probability that the top object in $\mathcal{X}$ is contained in $\mathcal{X}'$. A set of choice probabilities is said to satisfy Luce's choice axiom if $\forall \mathcal{X} \subset \mathcal{X}^*$ with at least two objects, $\mathbf{x}_i$ and $\mathbf{x}_j$, satisfy:

- If $\mathrm{P}_{\{\mathbf{x}_i, \mathbf{x}_j\}}[\mathbf{x}_i] > 0 \; \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ then $\forall \mathbf{x}_k \in \mathcal{X}' \subset \mathcal{X}, \mathrm{P}_{\mathcal{X}}[\mathbf{x}_k] = \mathrm{P}_{\mathcal{X}'}[\mathbf{x}_k] \mathrm{P}_{\mathcal{X}}[\mathcal{X}']$,
- If $\mathrm{P}_{\{\mathbf{x}_i, \mathbf{x}_j\}}[\mathbf{x}_i] = 0 \; \exists \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ then if $\mathbf{x}_j \in \mathcal{X}$, $\mathbf{x}_j \neq \mathbf{x}_i$, $\mathrm{P}_{\mathcal{X}}[\mathbf{x}_j] = \mathrm{P}_{\mathcal{X} \setminus \{\mathbf{x}_i\}}[\mathbf{x}_j]$.

This Plackett-Luce model is equivalent to the above Thurstonian model, if $\epsilon(\cdot)$ follows Gumbel distribution, whose c.d.f. is $1 - \exp(-\exp(x))$.

Acknowledgments: A part of this work is supported by the grant-in-aid 14658106 and 16700157 of the Japan society for the promotion of science. Thanks are due to the Mainichi Newspapers for permission to use the articles.

References

1. Marden, J.I.: Analyzing and Modeling Rank Data. Volume 64 of Monographs on Statistics and Applied Probability. Chapman & Hall (1995)
2. Kendall, M., Gibbons, J.D.: Rank Correlation Methods. fifth edn. Oxford University Press (1990)
3. Arrow, K.J.: Social Choice and Individual Values. second edn. Yale University Press (1963)
4. Bollegala, D., Okazaki, N., Ishizuka, M.: A machine learning approach to sentence ordering for multidocument summarization and its evaluation. In: Proc. of the Natural Language Processiong society of Japan. (2005)
5. 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. (2001) 613–622
6. Agresti, A.: Categorical Data Analysis. second edn. John Wiley & Sons (1996)

7. McCullagh, P.: Regression models for ordinal data. Journal of The Royal Statistical Society (B) **42**(2) (1980) 109–142
8. Shashua, A., Levin, A.: Ranking with large margin principle: Two approaches. In: Advances in Neural Information Processing Systems 15. (2003) 961–968
9. Cohen, W.W., Schapire, R.E., Singer, Y.: Learning to order things. Journal of Artificial Intelligence Research **10** (1999) 243–270
10. Grötschel, M., Jünger, M., Reinelt, G.: A cutting plane algorithm for the linear ordering problem. Operations Research **32**(6) (1984) 1195–1220
11. Littlestone, N.: Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. Machine Learning **2** (1988) 285–318
12. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. In: Proc. of The 15th Int'l Conf. on Machine Learning. (1998) 170–178
13. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research **4** (2003) 933–969
14. Kazawa, H., Hirao, T., Maeda, E.: Order SVM: a kernel method for order learning based on generalized order statistics. Systems and Computers in Japan **36**(1) (2005) 35–43
15. Herbrich, R., Graepel, T., Bollmann-Sdorra, P., Obermayer, K.: Learning preference relations for information retrieval. In: ICML-98 Workshop: Text Categorization and Machine Learning. (1998) 80–84
16. Herbrich, R., Graepel, T., Obermayer, K.: Support vector learning for ordinal regression. In: Proc. of the 9th Int'l Conf. on Artificial Neural Networks. (1999) 97–102
17. 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
18. 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: Proc. of the 15th European Conf. on Machine Learning. (2004) 286–297 [LNAI 3201].
19. Kamishima, T., Kazawa, H., Akaho, S.: Supervised ordering — an empirical survey. In: Proc. of The 5th IEEE Int'l Conf. on Data Mining. (2005) 673–676
20. Arnold, B.C., Balakrishnan, N., Nagaraja, H.N.: A First Course in Order Statistics. John Wiley & Sons, Inc. (1992)
21. Thurstone, L.L.: A law of comparative judgment. Psychological Review **34** (1927) 273–286
22. 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) 583–588
23. Kamishima, T., Fujiki, J.: Clustering orders. In: Proc. of The 6th Int'l Conf. on Discovery Science. (2003) 194–207 [LNAI 2843].
24. Kamishima, T., Akaho, S.: Efficient clustering for orders. In Zighed, D.A., Tsumoto, S., Ras, Z.W., Hacid, H., eds.: Mining Complex Data. Volume 165 of Studies in Computational Intelligence. Springer (2009) 261–280
25. Critchlow, D.E., Fligner, M.A., Verducci, J.S.: Probability models on rankings. Journal of Mathematical Psychology **35** (1991) 294–318
26. Mosteller, F.: Remarks on the method of paired comparisons: I — the least squares solution assuming equal standard deviations and equal correlations. Psychometrika **16**(1) (1951) 3–9

27. Babington Smith, B.: Discussion on professor ross's paper. Journal of The Royal Statistical Society (B) **12** (1950) 53–56 (A. S. C. Ross, "Philological Probability Problems", pp. 19–41).
28. Bradley, R.A., Terry, M.E.: Rank analysis of incomplete block designs — i. the method of paired comparisons. Biometrika **39** (1952) 324–345
29. Mallows, C.L.: Non-null ranking models. I. Biometrika **44** (1957) 114–130
30. Plackett, R.L.: The analysis of permutations. Journal of The Royal Statistical Society (C) **24**(2) (1975) 193–202

Index