# Dimension Reduction for Supervised Ordering

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

## Abstract

*Ordered lists of objects are widely used as representational forms. Such ordered objects include Web search results and best-seller lists. Techniques for processing such ordinal data are being developed, particularly methods for a supervised ordering task: i.e., learning functions used to sort objects from sample orders. In this article, we propose two dimension reduction methods specifically designed to improve prediction performance in a supervised ordering task.*

## 1 Introduction

Orders are sequences of objects sorted according to some property and are widely used to represent data. For example, responses from Web search engines are lists of pages sorted according to their relevance to queries. Best-seller lists, which are item sequences sorted according to sales volume, are used on many E-commerce sites. Processing techniques for orders have immediate practical value, and so research concerning orders has become very active in recent years. In particular, several methods are being developed for learning functions used to sort objects represented by attribute vectors from example orders. We call this task **Supervised Ordering** [14] and emphasize its usefulness for sensory tests[1] [14, 17], information retrieval [4, 9, 11, 20, 23] , and recommendation [8].

Several methods have been developed for the supervised ordering task. However, when the attribute vectors that describe objects are in very high dimensional space, these supervised ordering methods are degraded in prediction performance. The main reason for this is that the number of model parameters to be learned grows in accordance with the increase of dimensionality; thus, the acquired functions might not perform well when sorting unseen objects due to over-fitting.

[1] measurement of respondents' sensations, feelings or impressions

Dimension reduction techniques are one obvious solution to the problems caused by high dimensionality. Dimension reduction is the task of mapping points originally in high dimensional space to a lower dimensional sub-space, while limiting the amount of lost information. Principal component analysis (PCA) is one of the typical techniques for dimension reduction. PCA is designed so that variations in original data are preserved as much as possible. It has been successfully used for other learning tasks but is less appropriate for a supervised ordering task. Since PCA is designed so as to preserve information regarding the objects themselves, useful information in terms of the target ordering might be lost by this approach. Therefore, in this paper, we propose **Rank Correlation Dimension Reduction (RCDR)** for dimension reduction in conjunction with supervised ordering. RCDR is designed to preserve information that is useful for mapping to the target ordering.

We show a formalization of the supervised ordering and known facts regarding orders in Section 2. We propose our RCDR methods in Section 3. Experimental results are shown in Section 4. We discuss and summarize the results in Section 5.

## 2 Supervised Ordering

To describe the known properties of orders and the supervised ordering task, some basic notations must first be designed. An object, entity, or substance to be sorted is denoted by $\mathbf{x}_j$. The universal object set, $X^*$, 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 dimensions of attribute space. The order is denoted by $O = \mathbf{x}_a \succ \cdots \succ \mathbf{x}_j \succ \cdots \succ \mathbf{x}_b$. Note that subscript $j$ of $\mathbf{x}$ doesn't mean "The $j$-th object in this order," but that "The object is uniquely indexed by $j$ in $X^*$." The order $\mathbf{x}_1 \succ \mathbf{x}_2$ represents "$\mathbf{x}_1$ precedes $\mathbf{x}_2$." An object set $X(O_i)$ or simply $X_i$ is composed of all objects in the order $O_i$. The length of $O_i$, i.e., $|X_i|$, is denoted by $L_i$. An order of all objects, i.e., $O_i$ s.t. $X(O_i)=X^*$, is called a complete order; otherwise, the order is incomplete. Rank,

$r(O_i, \mathbf{x}_j)$ 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 $\mathbf{x}_a, \mathbf{x}_b{\in}X_1{\cap}X_2, \mathbf{x}_a{\neq}\mathbf{x}_b$. 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 $\mathbf{x}_a, \mathbf{x}_b{\in}X_1{\cap}X_2, \mathbf{x}_a{\neq}\mathbf{x}_b$.

We then describe the distance between two orders, $O_1$ and $O_2$, composed of the same sets of objects, i.e., $X(O_1) = X(O_2) \equiv X$. Various kinds of distance for orders have been proposed [18]. **Spearman distance** $d_S(O_1, O_2)$ is widely used. It is defined as the sum of the squared differences between ranks:

$$d_S = \textstyle\sum_{\mathbf{x}_j \in X} \left(r_{1j} - r_{2j}\right)^2. \quad (1)$$

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

$$\rho = 1 - 6d_S(O_1, O_2)/(L^3 - L), \quad (2)$$

where $L = |X|$. 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, $\{(\mathbf{x}_a, \mathbf{x}_b) \in X \times X\},\ a \neq b,\ \mathbf{x}_a, \mathbf{x}_b \in X$, including either $(\mathbf{x}_a, \mathbf{x}_b)$ or $(\mathbf{x}_b, \mathbf{x}_a)$. The number of object pairs is $M = (L-1)L/2$. 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,

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

where $\mathrm{sgn}(x)$ is a sign function that takes 1 if $x{>}0$, 0 if $x{=}0$, and $-1$ otherwise. By normalizing the range to be $[-1, 1]$, **Kendall's rank correlation** $\tau$ is derived.

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

The computational costs for deriving $\rho$ and $\tau$ are $O(L \log L)$ and $O(L^2)$, respectively. The values of $\tau$ and $\rho$ are highly correlated, because the difference between two criteria is bounded by Daniels' inequality [16]:

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

A **Supervised Ordering** task (Figure 1) can be considered as a regression or a fitting task whose target variables are orders. The input data is a set of sample orders,

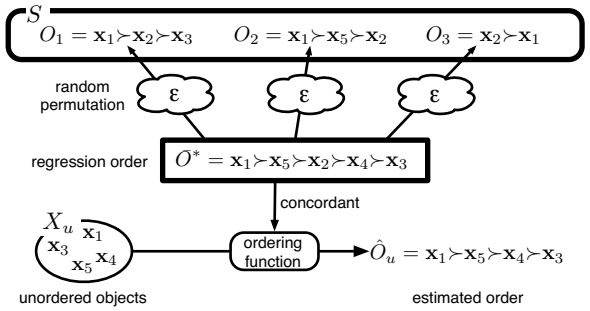



**Figure 1. A supervised ordering task**

$S = \{O_1, \ldots, O_N\}$, where $N$ is the number of samples. These samples give information about which objects should be ranked higher, and consist of objects represented by attribute vectors. No other side information, such as preference scores, is provided. The regression curve corresponds to a regression order. Analogous to a standard regression, 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. A regression order is modeled by an ordering function: Given an unordered object set, the ordering function outputs the estimated order, such that it is composed of the given unordered set and is concordant with the regression order. Supervised ordering also differs from classification because orders can be structured using symmetric groups, while classes cannot.

A supervised ordering task is closely related to a notion of a **central order** [18]: Given sample orders $S$, the central order $\bar{O}(S)$ is defined as the order that minimizes the sum of the distances $\sum_{O_i \in S} d(O_i, \bar{O})$. It differs from the above regression order in that concordance only with *given* samples is considered, and objects are represented not by attributes, but by unique identifiers. In a supervised ordering case, there may be objects not observed in given samples. (e.g., $\mathbf{x}_4$ in Figure 1) Such objects should be ranked under the assumption that the neighboring objects in the attribute space would be close in rank.

Supervised ordering is also related to **ordinal regression** [19], which is a regression in which response variables are ordered categorical. Similar to categorical variables, ordered categorical variables can take one of a finite set of predefined values, and these values are ordered additionally; for example, the domain of a variable may be {"good", "fair", "poor"}. Ordered categories and orders differ 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 lim-

ited. Consider a set of 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 differences of grades between any two objects. As pointed out in [3], supervised ordering is a more general problem than ordinal regression; thus, supervised ordering methods can be applied to ordinal regression tasks. Generally speaking, one should not try to solve a more general problem than is required. For example, an ordinal regression task can be solved by using an SVM designed for supervised ordering as in [10]. However, an SVM specialized for ordinal regression is more efficient [21]. Therefore, the two tasks, supervised ordering and ordinal regression, have to be carefully distinguished.

### 2.1 Methods and Applications of Supervised Ordering

Several methods have been developed for supervised ordering. In [14], these methods are surveyed and their pros and cons are shown. Below, we briefly show these methods. Cohen et al. [4] proposed a method adopting the paired comparison approach. Training examples are first decomposed into ordered pairs, and the algorithm learns probability functions in which one object precedes the other. Then, unordered objects are sorted so as to maximize the objective function that is a sum of these probability functions. RankBoost [8] tries to find a score function that is a linear combination of weak hypotheses. Weak hypotheses provide some partial information about the target order to learn. By using a boosting technique, weights and weak hypotheses are chosen so that scores are concordant with given sample orders. Unordered objects can be sorted according to the learned scores. Order SVM [15] learns near-parallel hyperplanes in the attribute vector space; the hyperplanes separate higher-ranked objects from lower-ranked ones. In the sorting stage, objects are ordered along the direction perpendicular to the hyperplanes. Support Vector Ordinal Regression (SVOR) [9] was proposed by Herbrich et al. In the learning stage, SVOR finds an optimal direction such that along this direction the minimum margin between a pair of objects is large. This method is independently proposed as the Ranking SVM by Joachims [11]. An active learning extension of this method is proposed by Yu [23].

We turn to our Expected Rank Regression (ERR) 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, an unseen complete order $O_i^*$ is generated. $(|X^*|-L_i)$ objects are then selected uniformly at random, and these are eliminated from $O_i^*$; then, the $O_i$ is observed. According to [1], under this assumption, the conditional expectation of ranks of the object $\mathbf{x}_j \in X_i$ in the unseen complete order given $O_i$ is

$$E[\hat{r}(O_i^*, \mathbf{x}_j)|O_i] \propto r(O_i, \mathbf{x}_j)/(L_i + 1). \quad (5)$$

These expected ranks are calculated for all objects in each $O_i \in S$. Next, weights of regression function $f(\mathbf{x}_j)$ are estimated by applying a standard regression method. Samples for regression consist of the attribute vectors of objects, $\mathbf{x}_j$, and their corresponding expected ranks, $r(O_i, \mathbf{x}_j)/(L_i+1)$; thus, the number of samples is $\sum_{O_i \in S} L_i$. Once parameters of $f(\mathbf{x}_j)$ are learned, the order $\hat{O}_u$ can be estimated by sorting the objects $\mathbf{x}_j \in X_u$ according to the values of $f(\mathbf{x}_j)$.

Next, we show some examples of applications. In [11, 20], a supervised ordering method is used to exploit relevance feedback data in a document retrieval task. The authors proposed an elegant technique to implicitly obtain users' feedback information about preference in retrieved documents. Assume that retrieved documents are listed by sorting the degree of relevance to the given query. If the user selected the third document $\mathbf{x}_c$, this action implies that the user prefers this document to the first, $\mathbf{x}_a$, or the second, $\mathbf{x}_b$, because he/she checks the sorted documents sequentially from the top of the list. So, relevance feedback data, $\mathbf{x}_c \succ \mathbf{x}_a$ and $\mathbf{x}_c \succ \mathbf{x}_b$, can be implicitly obtained. Further, documents are represented by features, such as similarity measures to the query words, types of documents, or the ranks in lists generated without exploiting feedback information. From these feedback data and features of documents, a supervised ordering method makes it possible to learn functions for sorting documents according to the degree of user's preference as well as the documents' relevance to the user's query.

In [17, 2], supervised ordering methods are used for sensory tests to examine which product features affect the value of the products. Metasearch engines are constructed in [4, 8]. Supervised ordering can be used to make content-based recommendation. Users' relative preference data are first obtained. Based on these preference orders and features of items, items can be sorted according to the degree of users' preference by applying a supervised ordering method. Finally, highly ranked items are recommended to users.

## 3 Rank Correlation Dimension Reduction

In the previous section, we defined a supervised learning task. Here, we show a dimension reduction technique specially designed for these supervised ordering methods.

To obtain satisfactory results when using data mining or machine learning algorithms, it is important to apply pre-processing methods, such as feature selection, dealing

with missing values, or dimension reduction. Appropriate pre-processing of data can improve prediction performance, and can occasionally reduce computational and/or memory costs. Some pre-processing techniques for mining or learning methods dealing with orders have been proposed. Bahamonde et al. [2] applied wrapper-type feature selection to a supervised ordering task. Slotta et al. [22] performed feature selection for classification of orders. In [6, 5], rank statistics were used for selecting informative genes from microarray data. To measure the similarities between orders, Kamishima and Akaho proposed a method to fill in missing objects in orders [13]. To our knowledge, however, dimension reduction techniques specially designed for a supervised ordering task have not yet been developed.

Similar to other types of learning tasks, such as classification or regression, dimension reduction techniques will be beneficial for supervised ordering tasks, in particular, if the number of attributes, $K$, is very large. With reduced dimensions, the generalization ability can be improved. Because the number of model parameters to be learned grows in accordance with $K$, the acquired functions might not perform well when sorting unseen objects due to over-fitting. In particular, if there are many non-informative attributes or if complex models are used, the problem of over-fitting will be alleviated by reducing dimensions.

To reduce the number of dimensions before performing supervised ordering, one might assume that reduction techniques used for other learning tasks can be used. However, this is not the case. Principal component analysis (PCA) is one of typical techniques for dimension reduction. PCA is designed so that information about data in original attribute vector space is preserved as much as possible. This approach is less appropriate for a supervised ordering task. Specifically, because a supervised ordering task must find a mapping from attribute vectors to the target ordering, it is not sufficient to preserve information only in source vectors. On the other hand, Diaconis' spectral analysis [7] for orders is another possibility. This is a technique to decompose distributions of orders into sub-components. For example, first-order components represent the frequency that the object $\mathbf{x}_j$ is $l$-th ranked, while second-order components represent the frequency that objects $\mathbf{x}_j$ and $\mathbf{x}_k$ are $l$-th and $m$-th ranked, respectively. However, our goal is not to find decomposition in an ordinal space, but to find a sub-space in an attribute vector space.

From the above discussion, it should be clear that we had to develop reduction techniques that preserve information about mappings from attribute vectors to the target ordering. This is analogous to Fisher's discriminant analysis, which is a dimension reduction technique to preserve information about a mapping from an attribute vector to target classes.

Additionally, the computational cost for reducing dimensions should not be much higher than that for supervised ordering methods. Computational complexities of supervised ordering methods in the learning stage are summarized in Table 1. We assume that the number of ordered pairs and objects in $S$ are approximated by $N\bar{L}^2$ and $N\bar{L}$, respectively ($\bar{L}$ 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 method or the RankBoost is linear in terms of $N\bar{L}^2$, if the number of iterations is constant. However, in practical use, the number of iterations should be increased adaptively. In the experiment in [8], the number of iterations was linearly increased in accordance with the number of ordered pairs, $N\bar{L}^2$. Therefore, their time complexities approach $N^2\bar{L}^4k$. When dimension reduction methods require much higher computational costs than those in Table 1, the reduction of dimensions greatly lessens scalability.

**Table 1. Computational complexities of supervised ordering algorithms**

| Cohen | RankBoost | SVOR | Order SVM | ERR |
|---|---|---|---|---|
| $N\bar{L}^2K$ | $N\bar{L}^2K$ | $N^2\bar{L}^4K$ | $N^2\bar{L}^4K$ | $N\bar{L}K^2$ |

NOTE: $\bar{L}$: the mean length of sample orders, $N$: the number of samples, and $K$: the dimension of attribute vectors. The number of ordered pairs and objects in $S$ are approximated by $N\bar{L}^2$ and $N\bar{L}$, respectively. The SVM's learning time is assumed to be quadratic in the number of training samples. The learning complexities of Cohen's method or the RankBoost are as above if the number of iterations is constant. However, in practical use, because the number of iterations should be increased adaptively in accordance with the number of ordered pairs, their time complexities approach $N^2\bar{L}^4k$.

Taking into account what is mentioned above, our dimension reduction methods should satisfy two requirements.

1. It must be designed so as to preserve information about mappings from object attributes to targeting orders.
2. The computational complexity for dimension reduction should not be much larger than that for supervised ordering algorithms.

To fill these requirements, we propose **Rank Correlation Dimension Reduction (RCDR)**. Given a basis that consists of $l$ vectors, the next $l+1$ vector is selected so as to preserve as much information about target ordering as possible. By repeating this procedure, we obtain the final sub-space.

First, we outline our RCDR method. Let $\mathbf{w}^{(l)}$ be the $l$-th vector of a basis. The sub-space spanned by the basis, $\{\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(l)}\}$, is called the $l$-th sub-space. We represent this sub-space by the matrix, $W^{(l)} = [\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(l)}]$.

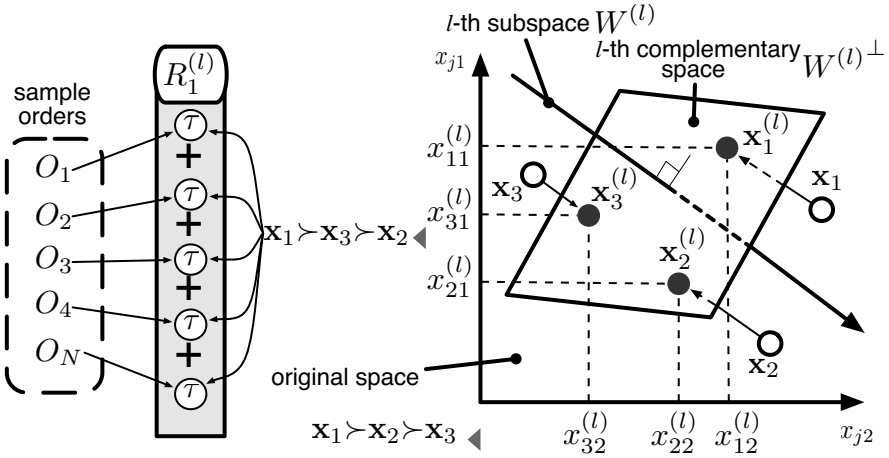



**Figure 2. An outline of rank correlation dimension reduction method**

Let $W^{(l)\perp}$ be the complementary space of the $W^{(l)}$, that is spanned by $(K-l)$ vectors which are orthogonal to all vectors in the basis, $\{\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(l)}\}$. We are given sample orders $S$ and attribute vectors, $\{\mathbf{x}_j\}$, and the basis of the $l$-th sub-space. This condition is depicted in Figure 2. The objects in the original $K$-dimensional spaces (marked by "$\circ$" in Figure 2) are projected to the complementary space $W^{(l)\perp}$ of the $l$-th sub-space. The projected objects (marked by "$\bullet$" in Figure 2) are denoted by $\mathbf{x}_j^{(l)}$, and $\mathbf{x}_j^{(0)} \equiv \mathbf{x}_j$. By this projection, we can eliminate information about the target ordering contained in the sub-space $W^{(l)}$. For each $k = 1, \ldots, K$, objects are sorted in descending order of the $k$-th attribute values of the objects projected to $W^{(l)\perp}$. In Figure 2, examples of those orders are $\mathbf{x}_1 \succ \mathbf{x}_3 \succ \mathbf{x}_2$ in the first attribute and $\mathbf{x}_1 \succ \mathbf{x}_2 \succ \mathbf{x}_3$ in the second attribute. The rank correlations between each of these orders and each sample order are calculated. Then, the sum of these rank correlations are denoted by $R_k^{(l)}$ (strict definition will be given later). This $R_k^{(l)}$ represents the concordance between the target ordering and the $k$-th attribute values of the objects projected on the $l$-th complementary space. A new vector, $\mathbf{w}^{(l+1)}$, is chosen so that each element of this vector, $w_k^{(l+1)}$, is as proportional to the corresponding concordance, $R_k^{(l)}$, as possible.

Now, we formally describe our RCDR. Let $\mathbf{w}^{(l)} = [w_1^{(l)}, w_2^{(l)}, \ldots, w_K^{(l)}]^\top$ be the $l$-th vector of a basis. These vectors are orthonormal to each other, i.e., ${\mathbf{w}^{(l)}}^\top \mathbf{w}^{(m)}=0$, $l \neq m$ and $\|\mathbf{w}^{(l)}\|=1$. The dimension of the final sub-space is denoted by $K'$. We are given a set of sample orders $S = \{O_1, \ldots, O_N\}$, the basis of the $l$-th sub-space, $W^{(l)}$, and the objects $\{\mathbf{x}_j | \mathbf{x}_j \in X_S\}$, $X_S \equiv \cup_{O_i \in S} X_i$. From these, we derive the $(l+1)$-th vector, $\mathbf{w}^{(l+1)}$, as follows. First, we define $R_1^{(l)}, \ldots, R_K^{(l)}$ as the concordances between sample orders and the attribute values of the objects projected on the complementary space, $W^{(l)\perp}$. Let us focus on the sample order $O_i$ and the $k$-th attribute values of objects. Because the goal of a supervised ordering task is to estimate the orders of objects, the relative ordering of attribute values is more important than the attribute values themselves. We therefore sort the $k$-th attribute values $x_{jk}^{(l)}$ of all objects $\mathbf{x}_j \in X(O_i)$ in descending order, where $x_{jk}^{(l)}$ denotes the $k$-th attribute value of the object, $\mathbf{x}_j^{(l)}$ projected on the $l$-th complementary space. Note that the projected objects are represented as $[x_{j1}^{(l)}, \ldots, x_{jK}^{(l)}]^\top$ on the coordinates of the original space. The resultant order is denoted by $O(X_i, x_{jk}^{(l)})$. Because both this $O(X_i, x_{jk}^{(l)})$ and the sample order $O_i$ consist of the same set of objects, the concordance between these two orders can be measured by Kendall's $\tau$. Such rank correlations are calculated between the $k$-th attribute values and each of sample orders in $S$, and these correlations are summed up:

$$R_k^{(l)} = \sum_{O_i \in S} \tau(O_i, O(X_i, x_{jk}^{(l)})). \tag{6}$$

We use this sum as a measure of the concordance between the $k$-th attribute values of objects and the target ordering. Next, to fill the first requirement of the RCDR, the $(l+1)$-th vector is chosen so that the above concordance is preserved as much as possible. Let us consider the vector,

$$\mathbf{R}^{(l)} = [R_1^{(l)}, \ldots, R_K^{(l)}]^\top.$$

Because the elements of this vector are the concordances between attribute values and the target ordering, this vector would point in the direction that preserves information about the target ordering in the attribute space. Therefore, we choose the vector $\mathbf{w}^{(l+1)}$ so that it maximizes the cosine between $\mathbf{w}^{(l+1)}$ and $\mathbf{R}^{(l)}$ in the complementary space,

**Input:**
$S = \{O_1, \ldots, O_N\}$: a sample order set
$\mathbf{x}_j \in X_S \equiv \cup_{O_i \in S} X_i$: attribute value vectors
$K'$: the dimension of sub-space
**Algorithm:**
1 $\mathbf{x}_j^{(0)} \equiv \mathbf{x}_j$
2 for $l$ in $0, \ldots, (K'-1)$
3 compute $\mathbf{R}^{(l)}$ s.t. $R_k^{(l)} = \sum_{O_i \in S} \tau(O_i, O(X_i, x_{jk}^{(l)}))$
4 if $l > 0$ then
$W^{(l)} = [\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(l)}]$, $\mathbf{R}^{(l)\perp} = (I - W^{(l)} W^{(l)\top}) \mathbf{R}^{(l)}$
else $\mathbf{R}^{(l)\perp} = \mathbf{R}^{(l)}$
5 $\mathbf{w}^{(l+1)} = \mathbf{R}^{(l)\perp} / \|\mathbf{R}^{(l)\perp}\|$
6 for $\mathbf{x}_j$ in $X_S$
7 $\mathbf{x}_j^{(l+1)} = \mathbf{x}_j^{(l)} - \mathbf{w}^{(l+1)} \mathbf{w}^{(l+1)\top} \mathbf{x}_j^{(l)}$
8 return $W^{(K')} = [\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(K')}]$

**Figure 3. Kendall rank correlation dimension reduction**

$W^{(l)\perp}$. Further, the vector $\mathbf{R}^{(l)}$ is constant, and $\mathbf{w}^{(l+1)} = 1$; thus, the maximization of this cosine is equivalent to the maximization between the dot product between $\mathbf{R}^{(l)}$ and $\mathbf{w}^{(l+1)}$. This optimization problem is formalized as follows:

$$\mathbf{w}^{(l+1)} = \arg\max_{\mathbf{w}} \mathbf{w}^\top \mathbf{R}^{(l)}, \tag{7}$$
$$\text{subject to:} \|\mathbf{w}^{(l+1)}\| = 1, \mathbf{w}^{(l+1)\top} \mathbf{w}^{(m)} = 0,\ m = 1, \ldots, l.$$

Note that one might think that $\mathbf{w}^{(l)}$ becomes a zero vector, if $l \geq 2$, but this is not the case. When the performing standard regression and Pearson's correlation is maximized, $\mathbf{w}^{(l)}$ would be a zero vector for $l \geq 2$. This is because zero Pearson's correlation implies such orthogonality in the attribute space. However, because rank correlation doesn't imply orthogonality, $\mathbf{w}^{(l)}$ is generally a non-zero vector even if $l \geq 2$.

Next, we solve Equation (7). The derivation of $\mathbf{w}^{(l+1)}$ can be easily shown by the following procedure: Calculate the vector of the correlations sums, $\mathbf{R}^{(l)}$, project this vector to the $l$-th complementary space, and normalize the projected vector. Once a new vector is derived, objects in the $l$-th complementary space, $\mathbf{x}^{(l)}$, are mapped to the new complementary space, and iteratively the next vector can be computed. This algorithm is shown in Figure 3. $\mathbf{R}^{(l)}$ is computed in line 3, projected to the current complementary space in line 4, and normalized in line 5 so that its norm is one. In lines 6 and 7, the objects in the current complementary space are projected to the new complementary space. Because the concordance is measured by Kendall's $\tau$, we call this method **Kendall RCDR**. The computational complexities of lines, 3, 4, 5, and 6-7 are $O(N\bar{L}^2 K)$, $O(KK')$, $O(K)$ and $O(N\bar{L}K)$, respectively; thus, the complexity per one iteration is $O(N\bar{L}^2 K)$ (generally $N\bar{L}^2 \gg K'$), and the total complexity is $O(N\bar{L}^2 KK')$. As noted before, because the complexity of Cohen's method and RankBoost practically approaches $O(N^2 \bar{L}^4 K)$, our Kendall RCDR is faster than supervised ordering methods except for ERR (see Table 1). To further save time complexity, we replace Kendall's $\tau$ in line 3 of the algorithm by Spearman's $\rho$, because $\rho$ and $\tau$ are highly correlated. We call this method **Spearman RCDR**. Because its time complexity is $O(NKK'\bar{L} \log \bar{L})$, this method becomes faster than the ERR method if $K' \log \bar{L} < K$. Therefore, our RCDR methods satisfy the second requirement. Note that the Kendall RCDR is faster than the Spearman RCDR in the special case: $L_i = 2, O_i \in S$. Joachims et al. proposed a method to implicitly collect sample orders whose lengths are two [11]. The Kendall RCDR is useful in such cases.

**Table 2. Vectors of a Basis derived by our RCDRs and the PCA**

the first vector

| method | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| KRCDR | 0.70 | 0.64 | 0.31 | −0.06 | −0.06 | 0.146 |
| SRCDR | 0.70 | 0.64 | 0.32 | −0.06 | −0.06 | 0.173 |
| PCA | 0.02 | −0.74 | 0.54 | −0.39 | 0.00 | 0.393 |

the second vector

| method | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| KRCDR | −0.27 | −0.17 | 0.93 | −0.13 | −0.13 | 0.007 |
| SRCDR | −0.30 | −0.15 | 0.94 | −0.05 | −0.05 | 0.007 |
| PCA | −0.06 | −0.18 | 0.39 | 0.90 | 0.00 | 0.213 |

NOTE: The first to fifth columns of each table show the components of vectors, $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$. In the last columns, values of $\|\mathbf{R}^{(l)}\|/N$ are shown for the RCDRs and the contribution ratio is shown for the PCA.

## 4 Experiments

After showing a simple example of our RCDR methods, we describe the experimental results for real data sets.

### 4.1 A Preliminary Experiment

To show what is produced by our two RCDR methods, we present a simple example using artificial data. We give the ideal weight vector by $\mathbf{w}^* = [1, 1, 0.5, 0, 0]$, and set the dimensions of the original space as $K = 5$ and the number of objects as $|X^*| = 1000$. For each object $\mathbf{x}_j \in X^*$, the first to the fourth attribute values are randomly generated according to the normal distribution, $N(0, 1)$, while the fifth

value is equal to the fourth. We generated 300 sample orders as follows: Five objects were selected uniformly at random from $X^*$; then these objects were sorted in descending order of ${\mathbf{w}^*}^\top \mathbf{x}_j$. We applied Kendall RCDR, Spearman RCDR, and PCA to this data set. The first and second vectors are shown in the upper and lower parts of Table 2, respectively. In each row, we show vectors derived by Kendall RCDR, Spearman RCDR, and PCA. The first to the fifth columns show the elements of vectors. In the last column, the norm lengths of the sum vector of rank correlations per sample order, $\|\mathbf{R}^{(l)}\|/N$, are shown for the RCDR cases, and the contribution ratios are shown for the PCA cases.

Let's look at the first vector. The vectors derived by the two RCDR methods show resemblance. This indicates that one can use the faster RCDR method; concretely, Spearman RCDR is better except for the case $L_i = 2$. Because the fourth and the fifth elements of the $\mathbf{w}^*$ are zero, no information useful for the target ordering is represented in these axes. In our RCDR cases, the fourth and the fifth weights of vectors are almost zero; thus, these useless axes can be ignored. In the PCA case, the fourth weight is far from zero, because no information about the target ordering is taken into account. The PCA merely ignores axes that are correlated in attribute space, such as in the fifth element. Further, because variances in all dimensions are equal, the contribution ratio is not so large, even if the target ordering is decided by a linear function.

We turn to the second component. In the RCDR cases, the correlation vector size $\|\mathbf{R}^{(2)}\|/N$ is much smaller than $\|\mathbf{R}^{(1)}\|/N$; this means that the second vector is far less informative than the first, because the target ordering is generated by a linear function in this example. In the PCA case, the contribution ratio indicates that useful information still remains in this vector. Note that it is not guaranteed that the $\|\mathbf{R}^{(l)}\|/N$ decreases in accordance with the increase of $l$, and vectors with bigger $\|\mathbf{R}^{(l)}\|/N$ don't always contribute to predicting the target ordering. However, we empirically observed that if $\|\mathbf{R}^{(l)}\|/N$ is very small, the corresponding vector is not informative. We believe that $\|\mathbf{R}^{(l)}\|/N$ can be used as an index for the importance of vectors.

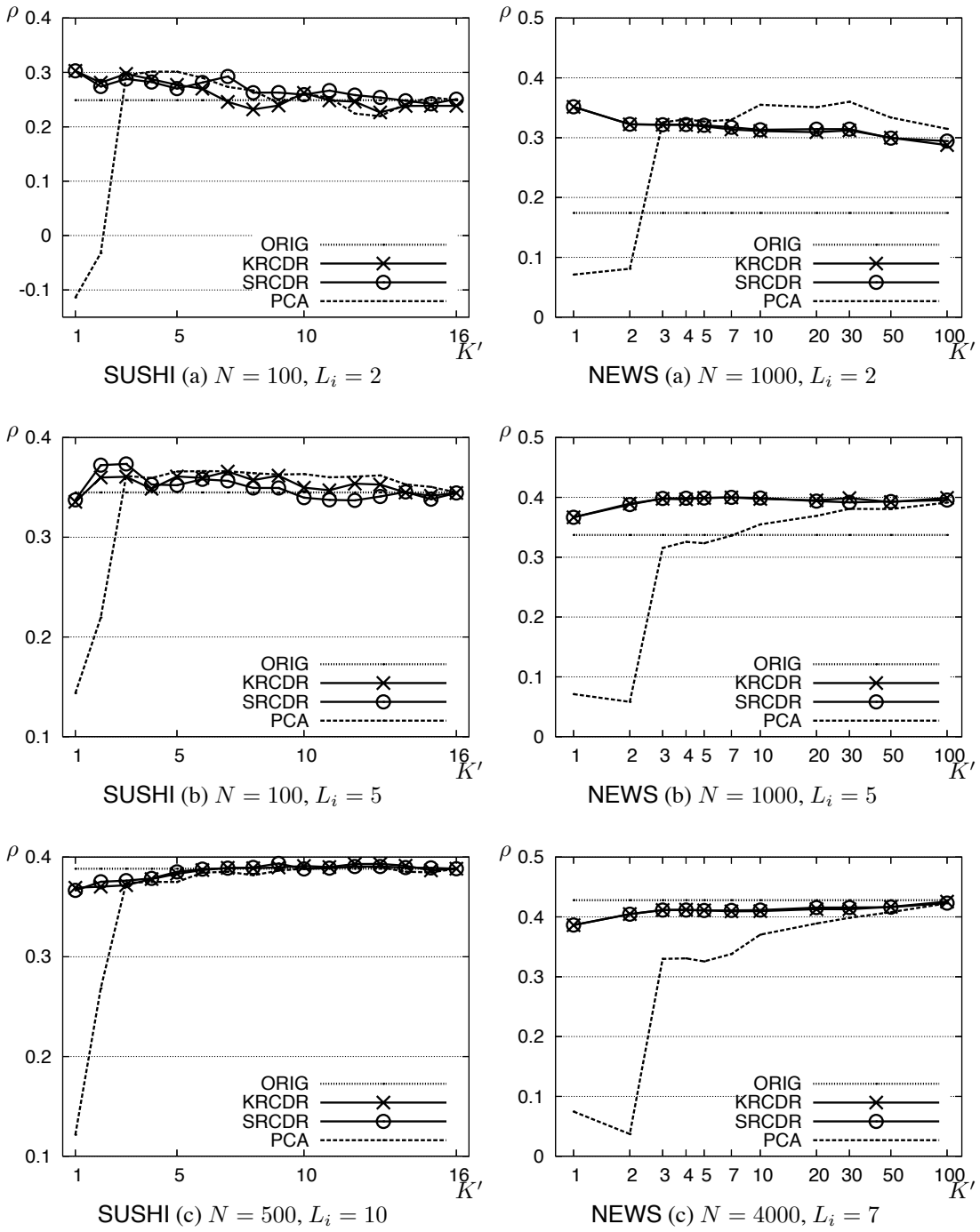
### 4.2 Experiments on Real Data Sets

We applied the methods described in Section 3 to real data from questionnaire surveys [14]. The first data set was a survey of preferences in sushi (Japanese food), and is denoted by SUSHI. In this data set, $N = 500$, $L_i = 10$, and $|X^*| = 100$. Objects are represented by 12 binary and 4 numerical attributes. The second data set was a questionnaire survey of news article titles sorted according to their significance, and is denoted by NEWS. These news articles were obtained from "CD-Mainichi-Newspapers 2003." In this data set, $N = 4000$, $L_i = 7$, and $|X^*| = 11872$. Titles were represented by 0-1 vectors indicating whether a specified keyword appears in the title. Among 18381 keywords, we selected 595 keywords that were observed 30 or more times. Additionally, we used 8 binary attributes to represent article categories; thus, the number of attributes was 603 in total.

To evaluate the usefulness of our dimension reduction methods, we applied the ERR supervised ordering method [14] to these two data sets. As a family of fitting functions, a linear model was adopted. Sample order sets were partitioned into testing and training sets. The ordering function was learned from training a sample order set with original attributes or reduced attributes. After learning, prediction performance was measured by the mean of $\rho$ between an order in a testing set, $O_t$, and the corresponding estimated order, $\hat{O}_t$. The larger $\rho$ was, the better the prediction performance was. The number of folds in cross-validation was ten for SUSHI and five for NEWS. In the left and right parts of the Figure 4, we show the variation of mean $\rho$ in accordance with the dimensions of the reduced space, $K'$, for SUSHI and NEWS, respectively. For both data sets, $N$ or $L_i$ was varied by eliminating sample orders or objects; the results for these sets are shown in each sub-figure. $N$ and/or $L_i$ increased from the sub-figure (a) to (c); thus, orders became the most difficult to estimate in the sub-figure (a) case. The curves labeled by "KRCDR", "SRCDR", and "PCA" show the mean $\rho$ derived by ERR after applying Kendall RCDR, Spearman RCDR, and PCA, respectively. The label "ORIG" indicates that no reduction method was used, and original attribute vectors were adopted.

From these figure, the following conclusions can be drawn. First, the two RCDR methods show resemblance; thus, the faster method can be used for dimension reduction. Second, both RCDRs performed better in prediction than PCA. The difference was particularly clear when the number of dimensions $K'$ was small. This means that RCDR successfully preserved information useful for estimating target orders. Therefore, we can say that RCDR is more effective than PCA when carrying out a supervised ordering task. Third, our RCDR technique could improve the prediction performance. The curves labeled "SRCDR"/"KRCDR" were compared with those labeled "ORIG." Surprisingly, the reduced vectors could lead to better prediction than the original vectors, even through some information might be lost by dimension reduction. We think that this is because the models used for ordering were simplified while useful information was preserved. This can be confirmed by the fact that the improvements were prominent when $N$ and/or $L_i$ were small. The simpler model could produce better generalization ability for a limited number of samples. Therefore, our reduction technique is useful for improving prediction performance.

We then compared the results in Figure 4 with the experi-




NOTE: The concordances between sample orders and estimated orders were measured by Spearman's $\rho$. These charts show variation of $\rho$ in accordance with the number of dimensions $K'$. $N$ is the size of a data set, and $L_i$ is the length of sample orders. The curves labeled "ORIG" show the result derived without application of dimension reduction. The curves labeled "KRCDR", "SRCDR", and "PCA" show the estimation results after reducing dimensions by the corresponding method.

**Figure 4. Comparison of dimension reduction methods on** SUSHI **(left) and** NEWS **(right) data sets**

**Table 3. Experimental results on real data sets in [14]**

| | $N$:$\lvert X_i \rvert$ | Cohen | RankBoost | SVOR | Order SVM | ERR |
|---|---|---|---|---|---|---|
| SUSHI | 500:10 | 0.364 [5] | 0.384 [4] | 0.393 [3] | 0.400 [1] | 0.397 [2] |
| | 100:5 | 0.354 [2] | 0.356 [1] | 0.284 [4] | 0.315 [3] | 0.271 [5] |
| | 100:2 | 0.337 [1] | 0.281 [2] | 0.115 [4] | 0.208 [3] | 0.010 [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] |

NOTE: This table shows the means of $\rho$. The rank of each method is shown in brackets. In this experiment, the same sets of attributes were adopted for the SUSHI, while slightly different attributes were used for NEWS. PCA was applied to compress keyword vectors, which were weighted by corresponding document frequencies. These 20 compressed attributes were used together with 8 additional binary attributes representing categories. Detailed experimental conditions are described in the extended version of [14] in our homepage [12].

mental results in [14]. The results in [14] were copied to Table 3. Rather different attributes were used, as described in the note of the table. Note that when we applied the SVOR method together with the large attribute sets used in Figure 4, the prediction accuracy was degraded. In this experiment, second-order polynomials were used for fitting functions for the ERR case; thus, the results differ from those in Figure 4. When observing these two results, for all NEWS and SUSHI-100:5 data sets, a linear ERR with RCDR could make a better prediction than all supervised ordering methods in Table 3, which adopted non-linear models for ordering. Further, our proposed method was the second best for the SUSHI-100:2 set, and the third best for the SUSHI-500:10 set. Therefore, we can say that our RCDR methods could successfully represent information about *non-linear* relations between attribute values and target ordering in orthonormal subspace.

Finally, we can exploit the components of vectors for qualitative analysis. We obtained the first vector, $\mathbf{w}^{(1)}$, derived from the SUSHI-500:10 data set by applying our Kendall RCDR method. The components of the vectors, $w_1^{(1)}, \ldots, w_K^{(K)}$, were sorted in descending order of their absolute values, $|w_k^{(1)}|$. The top 5 components were as follows:

| | |
|---|---|
| $w_{13}^{(1)} = 0.5951$ | the frequency the user eats |
| $w_{15}^{(1)} = 0.4278$ | how many restaurants supply the sushi |
| $w_{1}^{(1)} = 0.4237$ | red fish (e.g., fatty tuna) |
| $w_{14}^{(1)} = 0.2822$ | inexpensiveness |
| $w_{12}^{(1)} = -0.2317$ | lightness or non-oiliness in tasting |

From these components, we can say that "users primary prefer sushi that they frequently eat and that supplied in many sushi restaurants."

## 5 Discussion and Conclusion

In this paper, we proposed a dimension reduction technique specialized for a supervised ordering task. The method was designed so as to preserve information about a relation from object attribute vectors to the target ordering. For this purpose, we developed Kendall RCDR and Spearman RCDR. We then applied these methods to real data sets. From the experimental results, we arrived at the following conclusions. First, the RCDR methods outperform PCA when carrying out a supervised ordering task. Second, by using the RCDR technique, performance in prediction can be improved, especially when training samples are not adequate. Finally, our two RCDR methods are comparable in prediction performance. Therefore, the faster method should be used; concretely, Spearman RCDR is better except for the condition where $L_i = 2$.

Intuitively speaking, in the $l$-th iteration of the RCDR, the algorithm finds the vector that is most relevant to target ordering. After that, by mapping attribute vectors to the new sub-space, components in attributes related to this vector are subtracted. At this time, it might be effective to subtract the explained component in the target ordering from sample orders. We will try such improvement by using a technique like Diaconis' spectral analysis [7].

**Acknowledgments:** This work is supported by the grants-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] B. C. Arnold, N. Balakrishnan, and H. N. Nagaraja. *A First Course in Order Statistics*. John Wiley & Sons, Inc., 1992.

[2] A. Bahamonde, G. F. Bayón, J. D. J. R. Quevedo, O. Luaces, J. J. del Coz, J. Alonso, and F. Goyache. Feature subset selection for learning preferences: A case study. In *Proc.*

*of The 21st Int'l Conf. on Machine Learning*, pages 49–56, 2004.

[3] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proc. of The 22nd Int'l Conf. on Machine Learning*, pages 89–96, 2005.

[4] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

[5] L. Deng, J. Pei, J. Ma, and D.-L. Lee. A rank sum test method for informative gene discovery. In *Proc. of The 10th Int'l Conf. on Knowledge Discovery and Data Mining*, pages 410–419, 2004.

[6] M. Dettling and P. Bühlmann. Supervised clustering of genes. *Genome Biology*, 3(12):research0069.1–0069.15, 2002.

[7] P. Diaconis. A generalization of spectral analysis with application to ranked data. *The Annals of Statistics*, 17(3):949–979, 1989.

[8] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[9] R. Herbrich, T. Graepel, P. Bollmann-Sdorra, and K. Obermayer. Learning preference relations for information retrieval. In *ICML-98 Workshop: Text Categorization and Machine Learning*, pages 80–84, 1998.

[10] R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *Proc. of the 9th Int'l Conf. on Artificial Neural Networks*, pages 97–102, 1999.

[11] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. of The 8th Int'l Conf. on Knowledge Discovery and Data Mining*, pages 133–142, 2002.

[12] T. Kamishima. Homepage. http://www.kamishima.net/.

[13] T. Kamishima and S. Akaho. Filling-in missing objects in orders. In *Proc. of The 4th IEEE Int'l Conf. on Data Mining*, pages 423–426, 2004.

[14] T. Kamishima, H. Kazawa, and S. Akaho. Supervised ordering — an empirical survey. In *Proc. of The 5th IEEE Int'l Conf. on Data Mining*, pages 673–676, 2005.

[15] H. Kazawa, T. Hirao, and E. Maeda. Order SVM: a kernel method for order learning based on generalized order statistics. *Systems and Computers in Japan*, 36(1):35–43, 2005.

[16] M. Kendall and J. D. Gibbons. *Rank Correlation Methods*. Oxford University Press, fifth edition, 1990.

[17] O. Luaces, G. F. Bayón, J. R. Quevedo, J. Díez, J. J. del Coz, and A. Bahamonde. Analyzing sensory data using non-linear preference learning with feature subset selection. In *Proc. of the 15th European Conf. on Machine Learning*, pages 286–297, 2004. [LNAI 3201].

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

[19] P. McCullagh. Regression models for ordinal data. *Journal of The Royal Statistical Society (B)*, 42(2):109–142, 1980.

[20] F. Radlinski and T. Joachims. Query chains: Learning to rank from implicit feedback. In *Proc. of The 11th Int'l Conf. on Knowledge Discovery and Data Mining*, pages 239–248, 2005.

[21] A. Shashua and A. Levin. Ranking with large margin principle: Two approaches. In *Advances in Neural Information Processing Systems 15*, pages 961–968, 2003.

[22] D. J. Slotta, J. P. Vergara, N. Ramakrishnan, and L. S. Heath. Algorithms for feature selection in rank-order spaces. Technical Report TR-05-08, Computer Science, Virginia Tech., 2005.

[23] H. Yu. SVM selective sampling for ranking with application to data retrieval. In *Proc. of The 11th Int'l Conf. on Knowledge Discovery and Data Mining*, pages 354–363, 2005.