Supervised Ordering — An Empirical Survey

Toshihiro Kamishima
National Institute of Advanced
Industrial Science and Technology (AIST)
mail@kamishima.net (http://www.kamishima.net/)

Hideto Kazawa
NTT Communication Science Laboratories,
Nippon Telegraph and Telephone Corporation
kazawa@cslab.kecl.ntt.co.jp

Shotaro Akaho
National Institute of Advanced Industrial Science and Technology (AIST)
s.akaho@aist.go.jp

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 Supervised Ordering to acquire functions for object sorting from example orders. Here, we give a unified view of these methods and our new one, and empirically survey their merits and demerits.

1 Introduction

The term **order** indicates a sequence of objects that is sorted according to some property. For example, the responses from Web search engines are lists of pages sorted according to their relevance to queries. Retail stores use bestseller lists, which are item-sequences sorted according to sales volume. Research concerning orders has recently begun. In particular, several methods have been developed for learning functions used to sort objects from example orders. We call this task **Supervised Ordering**. We have advocated an unified view of the supervised ordering task to independently proposed tasks, and have considered the connection with the other types of tasks dealing with orders. We performed experiments targeting these methods, and our our preliminary results were reported in our extended abstract [25]. As the next step, in this work our new method and one more method were added to our survey, and these were checked by using more elaborate data sets.

We formalize the supervised ordering task in Section 2, survey methods in Section 3, and summarize our findings in Section 4.

2 Supervised Ordering

This section formalizes the supervised ordering task. We begin by defining basic notations. An object, entity, or substance to be sorted is denoted by x_j. The universal object set, X^*, consists of all possible objects. Each object x_j is represented by the attribute value vector $x_j = (x_{j1}, x_{j2}, \ldots, x_{jk})$, where k is the number of attributes. The order is denoted by $O = x_{j_a} \succ x_{j_b} \succ \cdots \succ x_{j_c}$. Note that the subscript j of x doesn't mean "The j-th object in this order," but that "The object is uniquely indexed by j in X^*." The order $x_1 \succ x_2$ represents "x_1 precedes x_2." An object set $X(O_i)$ or simply X_i is composed of all the objects in the order O_i; thus $|X_i|$ is equal to the length of the order O_i. An order of all objects, i.e., O_i s.t. $X(O_i) = X^*$, is called a complete order; otherwise, the order is incomplete. Rank, $r(O_i, x_j)$, is the cardinal number that indicates the position of the object x_j in the order O_i. For example, $r(O_i, x_2), O_i = x_1 \succ x_3 \succ x_2$ is 3. For two orders, O_1 and O_2, consider an object pair x_a and x_b, such that $x_a, x_b \in X_1 \cap X_2, x_a \neq x_b$. We say that the orders O_1 and O_2 are concordant w.r.t. x_a and x_b, if two objects are placed in the same order, i.e.,

$$(r(O_1, x_a) - r(O_1, x_b))(r(O_2, x_a) - r(O_2, x_b)) \geq 0;$$

otherwise, they are discordant. O_1 and O_2 are concordant if O_1 and O_2 are concordant w.r.t. all object pairs such that $x_a, x_b \in X_1 \cap X_2, x_a \neq x_b$.

We then describe the distance between orders, $d(O_a, O_b)$. Among many distances [28], we use well-known *Spearman distance* $d_S(O_a, O_b)$; this is defined as the sum of the squared differences between ranks. By normalizing the distance range to be $[-1, 1]$, the *Spearman's rank correlation* ρ is derived.

$$\rho = 1 - 6d_S/(|X|^3 - |X|). \tag{1}$$

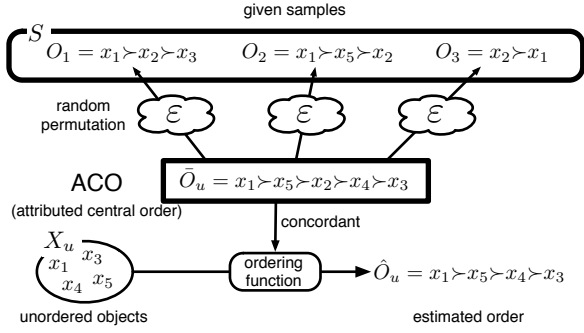This equal to the correlation coefficient of ranks of objects.

Figure 1. The supervised ordering task

We use this coefficient to evaluate the measure of prediction accuracy.

A **Supervised Ordering** task (Figure 1) 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, $S = \{O_1, \dots, O_N\}$, where N is the number of samples. The regression curve corresponds to an **Attributed Central Order** (ACO). Analogous to the case of a regression function, an ACO is estimated so as to be concordant not only with given sample orders in S, but also with orders that will be generated. This task differs from a regression in two ways. First, since the target variables are orders, the modeling methods of an ACO and errors are needed. An ACO is modeled by a ordering function, $\mathrm{ord}(X_u)$: Given an unordered object set X_u, $\mathrm{ord}(X_u)$ outputs the estimated order $\hat{O}_u$, such that it is composed of X_u and is concordant with the ACO. 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 $\varepsilon(\mathrm{ord}(X_i))$. Second, since samples 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. Supervised ordering is also different from classification, because orders can be structured using symmetric groups, but classes cannot. We say that a ordering function is *absolute* if outputs of the function are concordant with each other; otherwise, it is *relative*. This is equivalent to the condition 3, "The independence of irrelevant alternatives," of the Arrow's impossibility theorem [5]. For example, if unordered sets $\{x_1, x_2, x_3\}$ and $\{x_1, x_2, x_4\}$ are given to the absolute ordering function, then the function outputs orders that are concordant w.r.t. x_1 and x_2 regardless of objects x_3 or x_4. An absolute ordering function implies that the corresponding ACO is independent of the contents of an input unordered set.

A supervised ordering task is closely related to a notion of a **central order** [28]; given sample orders S, central or-

der $\bar{O}$ is defined as the order that minimizes the sum of the distances $\sum_{O_i \in S} d(O_i, \bar{O})$, and it differs from the above ACO in that concordance only with *given* samples is considered, and objects are represented not by attributes, but by unique identifiers. The derivation task of central orders is generally NP-hard. Many methods for this task have been developed, and these can be categorized as four types [10]: **Thurstonian** [33, 30], in which objects are sorted according to real score values, **Paired Comparison** [6, 27], based on the ordinal judgment between object pairs, **Distance Based** [27], depending on the distance from a modal order, and **Multistage** [31, 11], in which objects are sequentially arranged top to end. Supervised ordering methods are commonly designed by incorporating a way to deal with attributes into these ordering models. Supervised ordering methods adopting a Thurstonian model differs from a standard regression in a point that true score function doesn't always be needed to be estimated, since any monotonically transformed scores lead the same resultant orders.

Supervised ordering is also related to **Ordinal Regression** [1, 16, 2, 29, 12, 7, 9, 32, 3, 18], which is a regression whose a type of response 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, $x_1 \succ x_2$ means that x_1 is relatively better than 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.

3 Methods

We present five supervised ordering methods. Their abbreviations are given in parentheses in the section titles.

3.1 Cohen's method (Cohen)

Cohen's method [8] is designed to find the order $\hat{O}_u$ that maximizes

$$\sum_{x_a \succ x_b \in \hat{O}_u} \Pr[x_a \succ x_b | x_a, x_b], \qquad (2)$$

where $\Pr[x_a \succ x_b | x_a, x_b]$ is the conditional probability given the attribute values of x_a and x_b, and $x_a \succ x_b \in \hat{O}_u$ denotes all the ordered pairs concordant with $\hat{O}_u$. Unfortunately, the maximization of Equation (2) is known as a

linear ordering problem [15], and it is NP-hard. It is not tractable to find the optimal solution if $|X_u|$ is large. Cohen et al. hence proposed a greedy algorithm that sequentially chooses the most-preceding object.

$\Pr[x_a \succ x_b | x_a, x_b]$ is learned by Cohen et al.'s original Hedge algorithm. We set the β parameter to 0.9; the attributes $\{x_{jl}, -x_{jl}\}_{l=1}^k$ are used as ordering functions f in their paper. To use the Hedge algorithm in off-line mode, the same S is 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, experiments proved this method rather disadvantageous.

3.2 RankBoost (RB)

Freund et al. proposed **RankBoost** [13, 14], which is a boosting algorithm targeting orders. Inputs of RankBoost are the feedback function $\Phi(x_a, x_b)$, where $\Phi(x_a, x_b) > 0$ implies $x_b \succ x_a$, and ranking features $f_l(x_i)$, which gives partial information about target ordering. Given these inputs, RankBoost returns the final ranking $H(x_i)$, which works as a Thurstonian score function. First, the initial distribution is calculated by $D_1(x_a, x_b) = \max(\Phi(x_a, x_b), 0)/Z_1$, where Z_1 is a normalization factor. Then, for each round $t = 1, \ldots, T$, the algorithm repeats the selection of weight α_t and weak learner $h_t(x)$, and the update of distribution by:

$$D_{t+1}(x_a, x_b) = \frac{1}{Z_t} D_t(x_a, x_b) \exp\big(\alpha_t(h_t(x_a) - h_t(x_b))\big).$$

Weak learners acquire some information about target orders from ranking features, and $h_t(x_b) > h_t(x_a)$ implies $x_b \succ x_a$. 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 $x_j \in X_u$ are sorted in descending order of $H(x_j) = \sum_{t=1}^T \alpha_t h_t(x_j)$.

In our experiment, we adopt terms of n-order polynomials of object attributes as ranking features. Attribute values are transformed so as to be $[0, 1]$. Let $\Phi(x_a, x_b)$ be $2 \times \Pr[x_b \succ x_a] - 1$. Because attributes are binary or numerical, weak learners are set not to threshold function, $h_t(x) = [\![f_l(x) > \theta]\!]$, but to the ranking features itself, $h_t(x) = f_l(x)$. Selection method of α_t and h_t is the third option in section 3.2 in [14]. 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, **Order SVM** [26] and SVOR [17]. 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 [26].

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:

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

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

$$\xi_i^j(t) \geq 0 \quad \text{for } ^\forall i, j, t, \quad (3)$$

where $\mathbf{x}_i^j$ is the feature vector of the j-th ranked object in the i-th ranking, $\{\mathbf{x}_i^j\}_{i=1\ldots m}^{j=1\ldots L}$ are the training samples, and C and λ are hyperparameters (In this paper, $C = 0.1$ and $\lambda = 1$). The $\operatorname{sgn}[z]$ is 1 if $z \geq 0$; otherwise, -1. The SVM that discriminates the top t objects from the rest is $f_t(\mathbf{x}) = (\mathbf{w}+\mathbf{v}_t) \mathbf{x} + b_t$. 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 Thurstonian scores, $\mathbf{w} \cdot \mathbf{x}$. The dual problem of Equation (3) is similar to that of standard SVMs, and any kernel function can be used instead of the inner products between feature vectors [26].

We refer to the other SVM-based method as **Support Vector Ordinal Regression** (SVOR) 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 [19]. This method was independently developed as Ranking SVM by Joachims [20].

SVOR discriminates correctly ordered pairs from incorrectly ordered pairs, and uses a Thurstonian score function. In contrast to the Cohen method in which preferences 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 [8].

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

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \sum_{1 \leq j < l \leq L} \xi_i^{jl}$$

$$\text{s.t. } \mathbf{w} \cdot (\mathbf{x}_i^j - \mathbf{x}_i^l) \geq 1 - \xi_i^{jl}, \quad \xi_i^{jl} \geq 0 \text{ for } ^\forall i, j < l, \quad (4)$$

where the same notations as OSVM are used for $\mathbf{x}_i^j$, m, L, and C (In this paper, $C = 1$). SVOR tries to find the direction $\mathbf{w}$ along which sample objects are ordered so that the narrowest separation between samples is maximal. The estimated orders are predicted by sorting objects according to the Thurstonian scores, $\mathbf{w} \cdot \mathbf{x}$. As in the case of OSVM, the dual problem of Equation (4) can be written using only the inner products of $\mathbf{x}$. Thus we can use any kernel function in SVOR, as well.

3.4 Expected Rank Regression (ERR)

We turn to our new **Expected Rank Regression** (ERR) method, which is an improved version of the regression-based method in [23]. 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. $|X^*| - |X_i|$ objects are then selected uniformly at random, and these are eliminated from O_i^*; then, the O_i is observed. Under this assumption, the conditional expectation of ranks of the object $x_j \in X_i$ in the unseen complete order given O_i is proportional to: [4]

$$E[\hat{r}(O_i^*, x_j)|O_i] \propto r(O_i, x_j)/(|X_i| + 1). \qquad (5)$$

These expected ranks are calculated for all objects in each $O_i \in S$. Next, weights of regression function $f(x_j)$ are estimated by applying a common regression method. Samples for regression consist of the attribute vectors of objects, x_j, and their corresponding expected ranks, $r(O_i, x_j)/(|X_i| + 1)$; thus the number of samples is $\sum_{O_i \in S} |X(O_i)|$. In this paper, n-order polynomials are adopted as a class of regression functions. Once weights of $f(x_j)$ are learned, the order $\hat{O}_u$ can be estimated by sorting the objects $x_j \in X_u$ according to the Thurstonian scores of $f(x_j)$.

4 Discussions and Conclusions

We applied these supervised ordering methods to artificial and real data sets. We could not show these results in detail due to the lack of space; thus, we here summarize the results. Detailed experimental results can be found in the publication list page at our Web site [21].

In the first and second columns of Table 1, we summarize computational complexities of learning and sorting time. We assume that the number of ordered pairs and of objects in S are approximated by $N|\bar{X}|^2$ and $N|\bar{X}|$, respectively ($|\bar{X}|$ 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 $N|\bar{X}|^2$, if the number

of iterations T is fixed. However, if T is adaptively chosen according to $N|\bar{X}|^2$, 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, they are practically sub-linear. Generally, in practical use, the learning time of the SVM-based methods is slow, that of Cohen and RB is intermediate in speed, and that of ERR is much faster. In terms of time for sorting of $x_j \in X$, the Cohen greedy requires $O(|X|^2)$, while the others perform more quickly, $O(|X| \log |X|)$.

Finally, we can summarize the pros and cons of each method. Our new ERR method was practically the fastest without sacrificing its prediction performance. Therefore, algorithm parameters can be tuned in relatively short times. Even for the cases where ERR performed poorly, we observed that it could be improved by re-tuning. In this method, the uniform distribution of the object observation is assumed, but our experimental results demonstrated robustness against the violation of this assumption. A demerit of this method is quadratic computation time in terms of the number of attributes, k.

The most prominent merit of Cohen is that it is 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 only ordinal information among attribute values. We observed that performance could be improved by adopting the naive Bayes, which is designed to use categorical or numerical information in attribute values. The Cohen suffers from the problem of relative ordering. An absolute ordering 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 his/her preference. As described in Section 2, the supervised ordering task is related to ordering models. The Cohen method adopts paired comparison, while the others are Thurstonian. Accordingly, though absolute ordering functions can be acquired by any of the methods other than Cohen, Cohen learns relative functions.

The unique property of the RB is the 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 types, the other algorithms cannot be used. Our experimental results for RB were rather inferior, but we observed that they could be improved by adaptively increasing the number of rounds, T. Due to the slow convergence, we had to stop iterations after the end of the drastic error drop at the beginning stage. However, it takes the same or more computation time as the SVM-based methods until complete convergence. Furthermore, it should be also noted

Table 1. Computational complexities

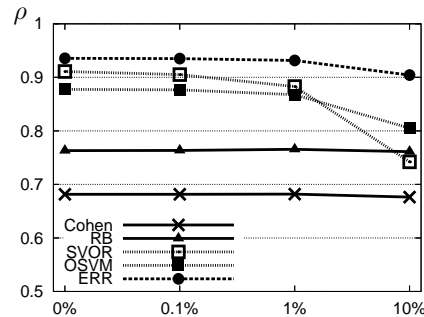|	Learning	Sorting
Cohen	$N\|\bar{X}\|^2 k$	$\|X\|^2$
RB	$N\|\bar{X}\|^2 k$	$\|X\|\log\|X\|$
SVOR	$N^2\|\bar{X}\|^4 k$	$\|X\|\log\|X\|$
OSVM	$N^2\|\bar{X}\|^4 k$	$\|X\|\log\|X\|$
ERR	$N\|\bar{X}\|k^2$	$\|X\|\log\|X\|$

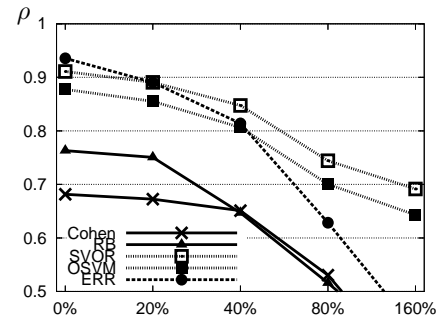**Figure 2. Order noise results**

Figure 3. Attribute noise results

that too many rounds T can cause over-fitting.

Like a standard SVM, the SVOR and OSVM are advantageous if the number of attributes, k, is large. We tested their robustness against order and attribute noises. Order noise is the permutation in sample orders, while attribute noise is the perturbation of attribute values. Figure 2 and 3 show the depression of estimation accuracies in accordance with the increase of order and attribute noise levels, respectively. The performance measure, Spearman's ρ, indicates that the larger is the more accurate prediction. The two SVM-based methods were robust against attribute noise, but not against order noise. This is because the interchanged ordered pairs tend to become support vectors, but the perturbation of attribute values does not affect the support vectors so much. Conversely, the non-SVM-based methods can learn correctly if correct orders constitute the majority of sample orders; thus, these are robust against order noise. However, any perturbation in attribute values always affects their performance. Hence, while the SVM-based methods are preferable for the less-order-noise condition, the other methods are suitable for the less-attribute-noise condition. The most serious demerit of SVM-based methods is their 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.

In our next study, we intended to improve upon these methods and apply them to practical problems, such as content-based filtering.

References

[1] A. Agresti. *Categorical Data Analysis*. John Wiley & Sons, second edition, 1996.

[2] J. A. Anderson. Regression and ordered categorical variables. *Journal of The Royal Statistical Society (B)*, 46(1):1–30, 1984.

[3] J. A. Anderson and P. R. Philips. Regression, discrimination and measurement models for ordered categorical variables. *Journal of The Royal Statistical Society (C)*, 30(1):22–31, 1981.

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

[5] K. J. Arrow. *Social Choice and Individual Values*. Yale University Press, second edition, 1963.

[6] B. Babington Smith. Discussion on professor ross's paper. *Journal of The Royal Statistical Society (B)*, 12:53–56, 1950. (A. S. C. Ross, "Philological Probability Problems", pp. 19–41).

[7] W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041, 2005.

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

[9] K. Crammer and Y. Singer. Pranking with ranking. In *Advances in Neural Information Processing Systems 13*, pages 641–647, 2002.

[10] D. E. Critchlow, M. A. Fligner, and J. S. Verducci. Probability models on rankings. *Journal of Mathematical Psychology*, 35:294–318, 1991.

[11] M. A. Fligner and J. S. Verducci. Multistage ranking models. *Journal of The American Statistical Association*, 83:892–901, 1988.

[12] E. Frank and M. Hall. A simple approach to ordinal classification. In *Proc. of the 12th European Conf. on Machine Learning*, pages 145–156, 2001. [LNAI 2167].

[13] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. In *Proc. of The 15th Int'l Conf. on Machine Learning*, pages 170–178, 1998.

[14] 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.

[15] M. Grötschel, M. Jünger, and G. Reinelt. A cutting plane algorithm for the linear ordering problem. *Operations Research*, 32(6):1195–1220, 1984.

[16] E. F. Harrington. Online ranking/collaborative filtering using the perceptron algorithm. In *Proc. of The 20th Int'l Conf. on Machine Learning*, pages 250–257, 2003.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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

[22] T. Kamishima. Nantonac collaborative filtering: Recommendation based on order responses. In *Proc. of The 9th Int'l Conf. on Knowledge Discovery and Data Mining*, pages 583–588, 2003.

[23] T. Kamishima and S. Akaho. Learning from order examples. In *Proc. of The 2nd IEEE Int'l Conf. on Data Mining*, pages 645–648, 2002.

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

[25] T. Kamishima, H. Kazawa, and S. Akaho. Estimating attributed central orders — an empirical comparison. In *Proc. of the 15th European Conference on Machine Learning*, pages 563–565, 2004. [LNAI 3201].

[26] 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.

[27] C. L. Mallows. Non-null ranking models. I. *Biometrika*, 44:114–130, 1957.

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

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

[30] F. Mosteller. Remarks on the method of paired comparisons: I — the least squares solution assuming equal standard deviations and equal correlations. *Psychometrika*, 16(1):3–9, 1951.

[31] R. L. Plackett. The analysis of permutations. *Journal of The Royal Statistical Society (C)*, 24(2):193–202, 1975.

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

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

A   Experiments Using Artificial Data

To reveal the characteristics of supervised ordering methods, we applied these methods to artificial data.

A.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. Second, ACOs are determined based on these attribute values. Objects are sorted according to the values of the function: $\text{utility}(x_j){=}(1 + \sum_{l=1}^{k} w_l x_{jl})^{\dim}$, where w_l are random weights. 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 ACOs.

As an error measure, we used Spearman's ρ (Equation (1)) between the estimated order and the above ACO. If ρ is 1, the two orders are completely concordant; if it is -1, one order is the reverse of the other. Note that similar results were observed when using another measure, such as Kendall's τ, top-3-precision, or positive/negative accuracy. We will show the means of ρ over a 10-fold cross validation for 10 different weight sets of $\text{utility}(x_j)$. Regardless of the length of training orders, the size of the unordered set, X_u, is set to 10, because errors cannot be precisely measured if orders are too short.

A.2   Experimental Results

As the basic experimental condition, we chose $N{=}300$, $|X_i|{=}5$, and $|X^*|{=}1000$, based on the observation of preliminary results in [25]. Under this condition, the probability that one object in X_u is unseen in the training samples seems rather low (25.8%). However, the probability that the object pairs in X_u become unseen, which is intrinsic to ordinal relations, is very 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: Cohen – Hedge and Greedy search; RB – ranking features are terms of a second-order polynomial. SVOR&OSVM – Gaussian kernel with $\sigma = 1$; ERR– regression function as a second-order polynomial. We consider that these settings would be first tried, and that no method presented an extremely advantageous model.

Table 2(a) shows the means of ρ under this basic setting. Each row corresponds to each of the four data sets described in Section A.1, and each column corresponds to each method in Section 3. The rank of each method is shown in brackets. Except between RB and SVOR of nl/bin, the difference between each method and the next-

Table 2. Basic Results: $|X^*|{=}1000$, $|X_i|{=}10$, $N{=}300$

(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]

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 regression methods were inferior in bin cases, but were superior in num cases. The performance with 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 $T = 1000$ for basic data under a noisy condition in Table 2(b), the ρ improved from 0.720 to 0.765. However, it was so slow that we had to set $T = 100$ when performing our experiments.

Table 2(a) shows the results under a noiseless condition; that is to say, all the sample orders are perfectly concordant with the corresponding ACO. To test the robustness of the methods against the noise in the orders, we permutated two randomly selected pairs of adjacent objects in the original sample orders. By changing the number of times that objects are permutated, the noise level could be controlled. The order noise level is measured by the probability that the ρ between the original order and the permutated one is smaller than the ρ between the original order and a random one. 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 2 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 permutated one) drastically became worse, whereas true ρ (between the estimated order and the non-permutated one) 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 $\alpha\%$ 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 3 shows the means of ρ in accordance with the level of attribute noise for the nl/num data. We generated five types of data whose α values were set to 0%~160%. As pointed out in our preliminary work [25], the results shown in Figures 2 and 3 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 2(a) is unrealistic, because real data generally include noise. We therefore investigated the behavior of supervised ordering 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. No method was almighty. 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,
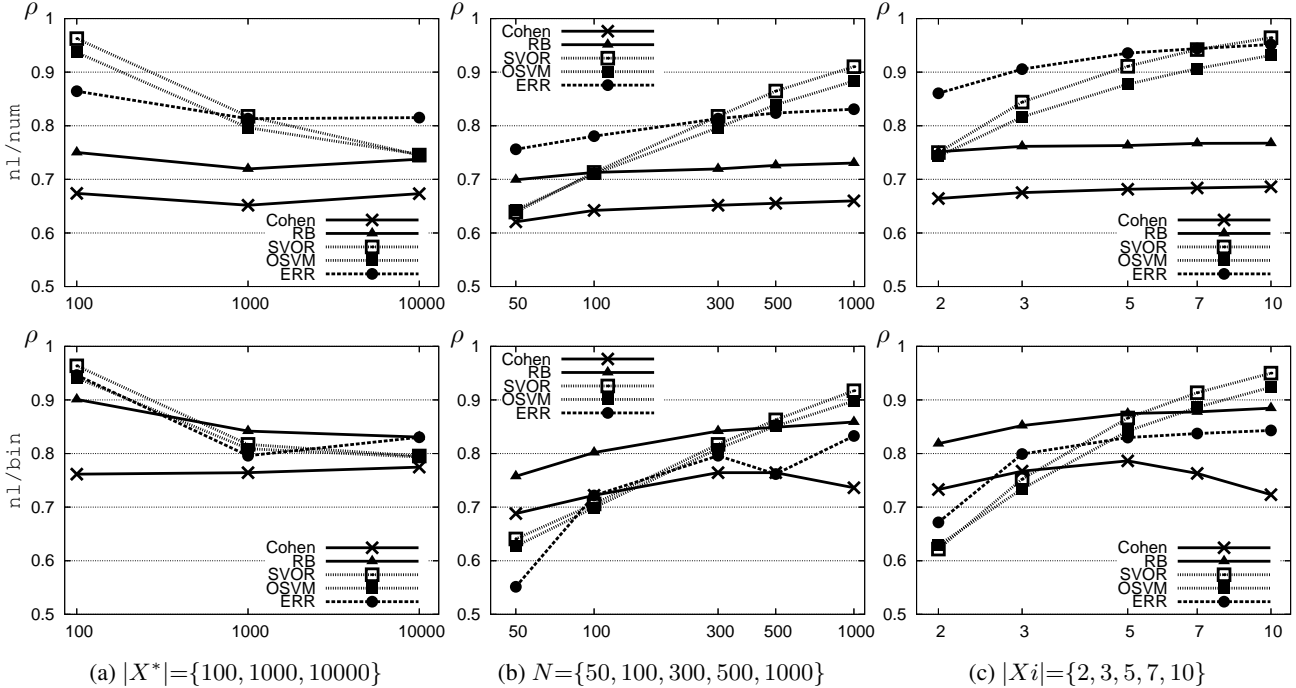
Figure 4. Variation in the number of objects $|X^*|$, the number of sample orders N, and the length of sample orders $|X_i|$

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. However, when testing the variation in the predictive performance according to the length of sample orders (Figure 4(c)), we used noiseless data, because the shorter sample orders were more seriously influenced by order noise. Algorithm parameters were tuned for these noisy basic data. Along with fixing the algorithm parameter settings, we tested the changes of the prediction performance according to variation in the number of objects $|X^*|$, the number of sample orders N, and the length of orders $|X_i|$.

Table 2(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 4 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 row. Columns (a), (b), and (c) show results when the number of objects $|X^*|$, the number of sample orders N, and the length of orders $|X_i|$ were varied, respectively. To check performance in practical use, we used *noisy* data sets.

In terms of Figure 4(a), the probability that an object in test orders has not been included in training samples de-

creases in accordance with the increase of $|X^*|$; accordingly, a greater generalization ability is required. SVM-based methods ware better if $|X^*|$ was small, but their performance dropped for larger $|X^*|$. Adoption of soft-margin parameter C tuning for $|X^*|$ 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 variables to determine is fewer in these methods than in the SVM-based ones.

Trurning to Figure 4(b) and (c), the Cohen method performed more poorly for the larger $|X_i|$. This would be because the paired comparison model used in the Cohen method assumes independence among ordered pairs. For small N or $|X_i|$, 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 $|X_i|$. 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 failed to find a simple relation between the number of variables to learn and the number of observed samples.

Table 3. Results on real data sets

	$N:|X_i|$	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]

B 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 [22]. In this data set, $N = 500$, $|X_i| = 10$, and $|X^*| = 100$. Objects are represented by 12 binary and 4 numerical attributes. By using the k-o'means clustering method [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. 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, $N = 4000$, $|X_i| = 7$, and $|X^*| = 11872$. The variance among sample orders was slightly broader than the tight SUSHI data. Articles were represented by keyword and document frequencies, and these were compressed to 20 attributes using latent semantic indexing. Additionally, we used 8 binary attributes to represent article categories. For both data sets, the N or $|X_i|$ were varied by eliminating sample orders or objects. Orders became difficult to estimate as N and/or $|X_i|$ decreased. Errors were measured by the empirical ρ.

In Table 3, we show the means of ρ. The column labeled $N:|X_i|$ 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 $|X_i|$, but poorer results for small N or $|X_i|$. This is due to a 2-order polynomial model, because the mean ρ became 0.249 for 100:2:(b) by adopting a linear model. Inversely, RB was better for small N or $|X_i|$. This is due to the setting of $T =$ 100, the number of rounds. When $T = 300$, we observed that performance for large N or $|X_i|$ improved, but it was depressed for small N or $|X_i|$ 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 tried several β parameters. Again, ERR was better for large N or $|X_i|$, but was poorer for small N or $|X_i|$. However, this was due to the advantage of a linear model 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. However, because their performances were sensitive to their parameters, we consider the most important problem for supervised ordering method research is to set helpful guidelines for these parameters.