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 [10]. 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. Two orders, O_1 and O_2, are concordant if ordinal relations are consistent between any object pairs commonly contained in these two orders; otherwise, they are discordant.

A **Supervised Ordering** 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 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

Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)
1550-4786/05 $20.00 © 2005 IEEE

IEEE COMPUTER SOCIETY

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. 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*. 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** [12]; given sample orders S, central order $\bar{O}$ 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 [4]: **Thurstonian**, in which objects are sorted according to real score values, **Paired Comparison**, based on the ordinal judgment between object pairs, **Distance Based**, depending on the distance from a modal order, and **Multistage**, 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 is also related to **Ordinal Regression** [1], 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.

Cohen's method (Cohen) [3] is designed to find the order $\hat{O}_u$ that maximizes the objective function

$$\sum_{x_a \succ x_b \in O_u} \Pr[x_a \succ x_b | x_a, x_b], \quad (1)$$

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 $x_a \succ x_b \in O_u$ denotes all the ordered pairs concordant with O_u. Because maximization of Equation (1) is known as a linear ordering problem, which 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.

Freund et al. proposed **RankBoost** (RB) [5], 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 selections of a weight α_t and a weak learner $h_t(x)$, and the updates of distribution by:

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

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. After learning, unseen objects $x_j \in X_u$ are sorted in descending order of $H(x_j) = \sum_{t=1}^{T} \alpha_t h_t(x_j)$, where T is the number of rounds.

We show two SVM-based methods, **Order SVM** (OSVM) [11] and SVOR. 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.

To enhance the reliability of this estimation, our OSVM trains 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:

$$\min_{\mathbf{w}, \mathbf{v}_t, b_t} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{\lambda}{2} \sum_{t=1}^{L-1} \|\mathbf{v}_t\|^2 + C \sum_{t=1}^{L-1} \sum_{i=1}^{m} \sum_{j=1}^{L} \xi_i^j(t)$$
$$\text{s.t. } \mathrm{sgn}[j-t]((\mathbf{w}+\mathbf{v}_t)\cdot \mathbf{x}_i^j + b_t) \geq 1 - \xi_i^j(t),$$
$$\xi_i^j(t) \geq 0 \quad \text{for } {}^{\forall} i, j, t, \quad (2)$$

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

IEEE COMPUTER SOCIETY

C and λ are hyperparameters. 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 Thurstonian scores, $\mathbf{w}\cdot\mathbf{x}$. The dual problem of Equation (2) is similar to that of standard SVMs, and any kernel function can be used instead of the inner products between feature vectors.

We refer to the other SVM-based method as **Support Vector Ordinal Regression** (SVOR) [6] since its formulation is very similar to standard SVMs and the work on it appears to be inspired by their past ordinal regression work. This method was independently developed as **Ranking SVM** by Joachims [7]. SVOR discriminates correctly ordered pairs from incorrectly ordered pairs, and uses a Thurstonian score function. SVOR's learning is formulated as the following optimization problem:

$$\min_{\mathbf{w}} \tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\sum_{1\leq j<l\leq L}\xi_i^{jl}$$
$$\text{s.t. } \mathbf{w}\cdot(\mathbf{x}_i^j-\mathbf{x}_i^l)\geq 1-\xi_i^{jl}, \quad \xi_i^{jl}\geq 0 \text{ for } {}^{\forall}i,j<l, \quad (3)$$

where the same notations as OSVM are used for $\mathbf{x}_i^j$, m, L, and C. 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 (3) can be written using only the inner products of $\mathbf{x}$. Thus we can use any kernel function in SVOR, as well.

We turn to our new **Expected Rank Regression** (ERR) method, which is an improved version of the regression-based method in [9]. 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, we 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: [2]

$$E[r(O_i^*, x_j)|O_i] \propto r(O_i, x_j)/(|X_i|+1). \quad (4)$$

These expected ranks are calculated for all objects in each $O_i \in S$. Next, weights of the 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 [8].

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.

1550-4786/05 $20.00 © 2005 IEEE

IEEE COMPUTER SOCIETY

Table 1. Computational complexities

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

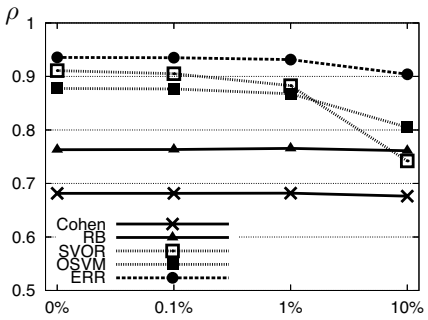

Figure 1. Order noise results

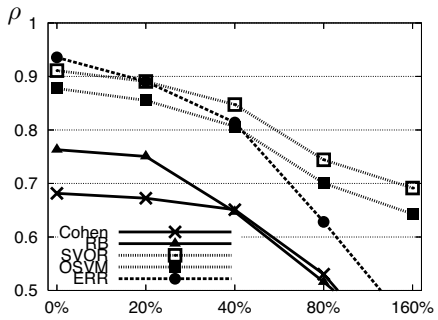

Figure 2. Attribute noise results

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 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 1 and 2 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.

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] A. Agresti. *Categorical Data Analysis*. John Wiley & Sons, 1996.

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

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

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

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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].

[11] 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.

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

IEEE COMPUTER SOCIETY