T. Kamishima and S. Akaho, "Learning from Order Examples", Proceedings of The IEEE International Conference on Data Mining, pp.645-648 (2002)

Learning from Order Examples

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

Abstract

We advocate a new learning task that deals with orders of items, and we call this the Learning from Order Examples *(LOE) task. The aim of the task is to acquire the rule that is used for estimating the proper order of a given unordered item set. The rule is acquired from training examples that are ordered item sets. We present several solution methods for this task, and evaluate the performance and the characteristics of these methods based on the experimental results of tests using both artificial data and realistic data.*

1 Introduction

In this paper, we advocate a new learning task that deals with orders of items, and we call this the *Learning from Order Examples* (LOE) task. The aim of the LOE task is to acquire the rule that is used for estimating the proper order of a given item set. The rule is acquired from training examples that are ordered item sets.

An example of performing the LOE task would consist of completing a questionnaire survey on preference in foods. The surveyor presents several kinds of foods to each respondent and requests that he/she sort the foods according to his/her preferences. By applying the LOE learning algorithm, for example, the surveyor will be able to determine the most preferred food, or to detect the degree of influence of attributes on respondent's preferences. For such a survey, it is typical to adapt the Semantic Differential method. In this method, the respondent's preferences are measured by a scale, the extremes of which are symbolized by antonymous adjectives. For examples:

[like] 5 4 3 2 1 [dislike].

Use of such a scale assumes that all the respondents share an understanding of its range, divisions and extremes. Such an unrealistic assumption can be avoided by introducing order scales and LOE techniques.

We present related works in Section 2, and formalize the LOE task in Section 3. Several LOE solution methods are presented in Section 4, and the experimental results in Sections 5 and 6. Section 7 summarizes our conclusions.

2 Related Works

Our LOE task is relevant to the work of Cohen et al. [1]. The inputs of their task are item pairs with the precedence information; that is information about which of the items should precede the other. From this set of pairs, their original algorithm derived a preference function $\mathrm{PREF}(I^x, I^y)$ measuring the confidence that item I^x precedes item I^y. They then attempted to find the order that maximizes the following function:

$$\sum_{x,y:I^x \succ I^y} \mathrm{PREF}(I^x, I^y), \tag{1}$$

where $I^x \succ I^y$ denotes that I^x precedes I^y. The most basic difference between their study and ours is that inputs are item pairs with precedence information, whereas inputs of LOE tasks are sets of ordered items. Additionally, their goal was to obtain orders that preserved the pairwise precedence information as closely as possible, whereas ours is to estimate totally well sorted orders. These two orders are closely related, but are clearly distinguishable from one another, as indicated by an experiment described in Section 5. Further, Cohen et al. considered errors in the PREF function, not errors in final orders. We, on the other hand, explicitly examined the errors in final orders. Recently, Kazawa et al. [2] dealt with the similar problem to ours, but adopted another measures for errors in orders.

Several other previous studies have dealt with orders. Mannila and Meek [4], for example, tried to establish the structure expressed by partial orders among a given set of ordered sequential data. Sai et al. [6] investigated association rules between order variables.

3 Formalization of a LOE task

This section formally states the task of learning from order examples (LOE). This task is composed of two major

stages: a learning stage and a sorting stage. In the learning stage, the rule for sorting is acquired from a training example set. In the sorting stage, based on the acquired rule, the true order of an unordered item set is estimated.

An item I^x corresponds to an object, entity, or substance to be sorted. Items are individualized by the attribute value vector, $A(I^x)=(a^1(I^x), a^2(I^x), \ldots, a^{\#A}(I^x))$ ($\#A$ is the number of attributes). In this paper, we concentrate on the case in which all the attributes are categorical. The domain of the s-th attribute is $v^s_1, \cdots, v^s_{\#a^s}$. The universal item set, $\{I\}_{All}$, consists of all possible items. An item set, $\{I\}_i$, is a subset of $\{I\}_{All}$. The number of items in $\{I\}_i$ is denoted by $\#I_i$.

An order is a sequence of items that are sorted according to some property, such as size, preference, or price. The order of the item set $\{I\}_i = \{I^x, I^y, \ldots, I^z\}$ is denoted by $O_i{=}I^x{\succ}I^y{\succ}\cdots{\succ}I^z$. To express the order of two items, $I^x{\succ}I^y$, we use the sentence "I^x precedes I^y." We assume an unobserved order of the universal item set, and call this the absolute order, O^*_{All}.

The example is a 2-tuple, $(\{I\}_i, O^*_i)$; an item set and its true order. In a noiseless case, the true order is consistent with the absolute order. In a realistic situation, however, true orders may be affected by noises, e.g., swapping of item positions or changes of attribute values. An example set, EX, consists of $\#EX$ examples, as follows:

$$EX = \{(\{I\}_1, O^*_1), (\{I\}_2, O^*_2), \ldots, (\{I\}_{\#EX}, O^*_{\#EX})\}.$$

Note that there are items included in $\{I\}_{All}$ that do not appear in any examples.

The aim of the LOE task is to acquire the rule from the above training example set. The acquired rule is then used for estimating the true order of an unordered item set. We denote an unordered item set by $\{I\}_U$, and its estimated order by $\hat{O}_U$. Note that attribute value vectors of items in the unordered set are known.

In order to directly evaluate the errors in orders, we adopt the *Spearman's Rank Correlation Coefficient* or the "ρ" [3]. The ρ is the correlation between ranks of items. The rank, $r(O, x)$, is the cardinal number that indicates the position of I^x in the order O. For example, for the order $O{=}I^3{\succ}I^1{\succ}I^2$, the $r(O, 3){=}1$ and the $r(O, 2){=}3$. If no tie in rank is allowed, the ρ between two orders, O^1_i and O^2_i, can be simply calculated as follows:

$$\rho = 1 - \frac{6 \times \sum_{I^x \in \{I\}} (r(O^1, x) - r(O^2, x))^2}{(\#I)^3 - \#I}.$$

The ρ becomes 1 if two orders are coincident, and -1 if one order is a reverse of the other order.

4 Methods

We describe two classification-based and one regression-based solution methods for the LOE task.

4.1 The LOE Methods Based on Classification Techniques

This method is similar to that of Cohen et al. The examples are decomposed into a set of item pairs, and the preference function is derived from these pairs. The unordered items are sorted based on this function.

In the learning stage, from the item set $\{I\}$ in the example $(\{I\}, O^*)$, all the item pairs, (I^x, I^y), are extracted such that I^x precedes I^y in the order O^*. For example, from the order $O^* = I^3{\succ}I^1{\succ}I^2$, three item pairs, (I^3, I^1),(I^3, I^2), and (I^1, I^2), are extracted. Such pairs are extracted from all $\#EX$ examples, and these are collected into the set P.

Then the preference function, $\mathrm{PREF}(I^x, I^y)$, is derived from the set P. This function, when given the attribute values of $A(I^x)$ and $A(I^y)$, outputs the confidence that I^x precedes I^y in the absolute order. To derive this preference function, we adopt the technique of the naive Bayesian classifier [5], as follows:

$$\mathrm{PREF}(I^x, I^y) = \mathtt{Pr}[I^x{\succ}I^y | A(I^x), A(I^y)]$$
$$= \frac{\mathtt{Pr}[A(I^x), A(I^y) | I^x{\succ}I^y]}{\mathtt{Pr}[A(I^x), A(I^y) | I^x{\succ}I^y] + \mathtt{Pr}[A(I^x), A(I^y) | I^y{\succ}I^x]},$$

$$\mathtt{Pr}[A(I^x), A(I^y) | I^x{\succ}I^y] \approx \prod_{s=1}^{\#A} \mathtt{Pr}[a^s(I^x), a^s(I^y) | I^x{\succ}I^y].$$

Note that $\mathtt{Pr}[I^x{\succ}I^y] = \mathtt{Pr}[I^y{\succ}I^x] = 0.5$ is assumed. As the probability $\mathtt{Pr}[a^s(I^x), a^s(I^y) | I^x{\succ}I^y]$, we adopt the following Bayesian estimator with Dirichlet prior in order that the probability keeps non-zero:

$$\frac{\#(a^s(I^x), a^s(I^y)) + 1/(\#a^s)^2}{\#P + 1},$$

where $\#(a^s(I^x), a^s(I^y))$ is the number of all the pairs (I^z, I^w) such that $a^s(I^x){=}a^s(I^z)$ and $a^s(I^y){=}a^s(I^w)$, and $\#P$ is the number of pairs in P.

In the sorting stage, by using $\mathrm{PREF}(I^x, I^y)$, the true order of $\{I\}_U$ is estimated. We examined the SumClass and ProductClass strategies, as follows.

SumClass (SC): The following greedy algorithm is designed so as to maximize the Equation (1); that is, the target function of Cohen et al.

1) $\hat{O}^{(0)} := \emptyset$, $\{I\}^{(0)} := \{I\}_U$, $t := 0$
2) $I^x := \arg\max_x \sum_{y: I^y \in \{I\}^{(t)}, x \neq y} \mathrm{PREF}(I^x, I^y)$
3) $\hat{O}^{(t+1)} := \hat{O}^{(t)}{\succ}I^x$, $\{I\}^{(t+1)} := \{I\}^{(t)} - I^x$
4) if $\{I\}^{(t+1)} = \emptyset$ then output $\hat{O}^{(t+1)}$ as $\hat{O}_U$
else $t := t + 1$, goto step 2

Simply speaking, this algorithm chooses, one by one, the most-preceding item. Note that this algorithm becomes equivalent to the greedy method of Cohen et al. if $\mathrm{PREF}(I^x, I^y){=}1{-}\mathrm{PREF}(I^y, I^x)$ is satisfied.

ProductClass (PC): This strategy is the same as the SumClass strategy, except for this criterion of optimality. As the criterion, Cohen et al. adopted Equation (1), (i.e., the sum of the PREF's values), but they did not present any theoretical reason for adopting this sum. We therefore test the product of the PREF values, because this value represents the likelihood of precedence events under the independence assumption. Though these events are not in fact independent, we consider that on a theoretical basis, this criterion has an advantage over that of Cohen et al.

The algorithm is the same as the SumClass strategy, excepting for step 2, which is as follows:

2) $I^x := \arg\max_x \prod_{y:I^y \in \{I\}^{(t)}, x \neq y} \mathrm{PREF}(I^x, I^y)$

4.2 The LOE Methods Based on Regression Techniques

All the orders in the training example are integrated into one total order. By using regression techniques, the evaluation function used for estimating an item's rank is derived. Any unordered items are sorted according to the value of this function. We use the abbreviation "R" for this method.

At the Learning stage, all items that appear in the training example set are collected into one item set, $\{I\}_C$. Next, the system finds the combined order for the $\{I\}_C$ that is as consistent with the orders in the training examples as possible. To derive this combined order, O_C, make the set of item pairs P in the previous section, and calculate the following preference function:

$$\mathrm{PREF}'(I^x,I^y) = \mathtt{Pr}[I^x \succ I^y] = \frac{\#(I^x,I^y)+0.5}{\#(I^x,I^y)+\#(I^y,I^x)+1}$$

where $\#(I^x, I^y)$ is the number of the item pairs, (I^x, I^y), in P. By using the strategy ProductClass and the above function PREF', the combined order is derived. Note that the function PREF' is different from the previous PREF with regard to the dependence on attribute values.

From the order O_C, we then acquire the ranking function, RANK, that measures the tendency of precedence. This function is derived by a linear regression technique in which dummy variables are adopted, also known as the *Type I quantification method*. One categorical attribute, $a^s(I)$, is represented by $\#a^s-1$ dummy variables. The first attribute value, v_1^s, is transformed into all the 0 dummy variables, and the other values, v_t^s, are transformed into dummy variables, the $(t-1)$-th element of which is 1 and the other elements of which are 0. For example, assume the attribute $a^s(I)$ can take 3 values. The values v_1^s and v_3^s are transformed into the dummy variables $(0,0)$ and $(0,1)$, respectively. All the variables in $A(I)$ are transformed into dummy variables, and these are concatenated into one vector, $d(A(I))$. For each element I in $\{I\}_C$, the $d(A(I))$ is derived. These vectors are combined into the matrix D whose i-th row is the $d(A(I^x))$ such that the rank of I^x is i in the order O_C. By using the following X, the function $\mathrm{RANK}(A(I^x))$ is defined as $X^T d(A(I^x))$:

$$X^T = (D^T D)^{-1} D^T (1, \ldots, \#I_C)^T.$$

At a sorting stage, the true order of an unordered item set is estimated by sorting the values of $\mathrm{RANK}(d(A(I)))$.

Table 1. The means of ρs

	ALL	3	5	10
SC	0.808	0.667	0.825	0.932
PC	0.808	0.667	0.825	0.932
R	0.802	0.617	0.837	0.950

5 Experiments on Artificial Data

We apply the three methods described in the previous section to artificial data in order to analyze the characteristics of these methods.

We prepared 9 types of universal item sets. The absolute order for these item sets was decided based on the linear weight function. For each type of universal item set, we randomly generated 10 different weights. Accordingly, 90 2-tuples of a universal item set and its absolute order were generated. Furthermore, from each of these tuples, we generated 9 example sets: $\#I$ (the numbers of items) was 3, 5, or 10, and the $\#EX$ (the numbers of examples) was 10, 30, or 50, respectively. In total, 810 example sets were generated. All the data sets were noiseless; that is, all the true orders were consistent with the absolute order.

As the testing procedure, we adopted a leave-one-out (LVO) test; that is, a $\#EX$-fold cross-validation test. As an error measure, we adopted the ρ between the estimated order and the absolute order.

Table 1 shows the means of the ρs. The column labeled "ALL" shows the mean over all 810 example sets, and the columns labeled 3, 5, and 10 show the means over the example sets composed of the item sets whose sizes were 3, 5, and 10, respectively. Overall, in accordance with increase in the size of the item sets, the more proper orders are estimated. For the ρ between two random orders lengths of which are $\#I$, it is known that

$$\rho\sqrt{(\#I-2)/(1-\rho^2)}$$

follows the Student t-distribution with degree of freedom $\#I-2$. Based on this fact, we could check whether, on average, proper orders were estimated or not. All three methods can produce the proper orders when $\#I$=10 at a significance level of 1%. In short, all three methods work well when item sets are large.

Table 2. The t-values between ρs

	ALL	3	5	10
$SC-PC$	-0.1472	-0.2709	0.3806	-1.5912
$SC-R$	1.4430	$\overline{4.4143}$	-2.2272	$\underline{-8.5784}$
$PC-R$	1.4626	$\overline{4.4254}$	$\underline{-2.3547}$	$\underline{-8.5023}$

We next applied paired t-tests to determine the differences between methods. The α–β row of Table 2 shows the t-values for the α method's ρ minus the β method's ρ. The overline (underline) indicates that method α (β) is superior at the significance level of 1%. The columns are the same those in Table 1. The SC and the PC methods were approximately equal in terms of accuracy. However, we consider that the PC method is preferable because it has a theoretical advantage. Although the R method is significantly inferior when item sets are small, the R method surpasses the other two methods as the sizes of the item sets grow. From another experimental result not presented here due to lack of space, it seems that this phenomenon is due to the degree of transitive consistency among the precedence events.

Additional experimental results demonstrated the performance of the SC and the PC methods. These methods adopt a greedy search to find the order that maximizes the sum or the product of the PREF values; thus, the optimal solution may not be acquired. The degree to which the performance by this solution, compared with the optimal solution, is examined. By applying the paired t-test to the optimal ρ minus the greedy ρ, we obtain a t-value of -2.7915 for the SC and -2.9306 for the PC. Surprisingly, this means that the solution by the optimal search is significantly worse than that by the the greedy search. Namely, an effort to preserve the pairwise precedence information does not lead to an order minimizing the rank correlation. This result enhances the distinction between our LOE task and that of Cohen et al. as described in Section 2.

6 Experiments on Realistic Data

We applied the methods in Section 4 to more realistic data in the form of answers to a small questionnaire. We surveyed subjects on their preferences of *sushi* (Japanese food). We asked 52 people to sort 10 types of sushi according to his/her preference. Each item (i.e., a specific type of sushi) is described by five attributes, each of which can take three to five attribute values.

We first generated all possible combinations (i.e., $2^5-1=31$ combinations) of attributes. For each combination and each of the three methods, we applied the LVO test and derived the means of the ρs between the true orders and the estimated orders. For each method, we selected the best combinations of attributes based on these means of the ρs. The means by using the best combinations are as follows:

SC	PC	R
0.451	0.454	0.455

For the orders estimated by all of the three methods, we observed a correlation between the estimated and the true orders at a significance level of 10%, on average.

The t-values for the differences between the ρs derived by two different methods are as follows:

SC$-$PC	SC$-$R	PC$-$R
-0.1807	-0.3098	-0.1002

We found no statistically significant differences among the three methods. An advantage of the R method was observed in the results on artificial data when the size of item sets are 10, but no such advantage was observed in this result. This is because sufficient examples are available relative to the size of the universal item set, and any method can therefore successfully derive high-quality orders. This was confirmed by the fact that all three methods estimated similar orders.

Finally, it should be noted that by observing the estimated orders and other related data, we can perform further analysis. For example, the survey answers revealed that the most preferred type of sushi is *toro* (fatty tuna). Since the ρ between the estimated order by the R method and my (the first author's) preference order is 0.842, it can be concluded that I have a highly typical tendency in terms of preference of sushi type.

7 Conclusion

We proposed a new learning task and presented its solution methods. We intend to develop the LOE technique so that it can be used to estimate order while directly minimizing the ρ error.

References

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

[2] H. Kazawa, T. Hirao, and E. Maeda. Ranking SVM and its application to sentence selection. In *Proc. of 2002 Workshop on Information-Based Induction Sciences*, 2002. (in Japanese).

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

[4] H. Mannila and C. Meek. Global partial orders from sequential data. In *Proc. of The 6th Int'l Conf. on Knowledge Discovery and Data Mining*, pages 161–168, 2000.

[5] T. M. Mitchell. *Machine Learning*. The McGraw-Hill Companies, 1997.

[6] Y. Sai, Y. Y. Yao, and N. Zhong. Data analysis and mining in ordered information tables. In *Proc. of the IEEE Int'l Conf. on Data Mining*, pages 497–504, 2001.