Filling-in Missing Objects in Orders

Toshihiro Kamishima 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/) s.akaho@aist.go.jp

Abstract

Filling-in techniques are important, since missing values frequently appear in real data. Such techniques have been established for categorical or numerical values. Though lists of ordered objects are widely used as representational forms (e.g., Web search results, best-seller lists), filling-in techniques for orders have received little attention. We therefore propose a simple but effective technique to fill-in missing objects in orders. We built this technique into our collaborative filtering system.

1 Introduction

We developed a technique to fill-in missing objects in orders, and built this technique into our collaborative filtering system based on order responses.

An *order* is a sorted sequence of objects, in which the only meaningful determination is which object precedes or succeeds the others. Such orders are widely used as representational forms. For example, Web search engines return page lists sorted according to their relevance to queries. Further, best-seller lists, which are item-sequences sorted according to the volume of sales, are used on many E-commerce sites. In spite of their importance, the methods of processing orders have received little attention. Filling-in missing objects in orders is one such processing task. This task is important, since missing values are frequently observed in real data. Specifically, given a set of sample orders, we developed a method of determining the rank of an object that doesn't appear in one order among the samples, based on the summaries of samples. This is an analogy for filling-in missing numerical values by means of samples.

We were motivated to develop our filling-in technique to perform better recommendation in collaborative filtering (*CF* for short). CF is a framework for recommending items based on the other users' preference patterns [2, 8]. Almost all CF methods adopt the Semantic Differential (SD) method [7] to measure users' preferences. In this method, the users expose their preference by using, for example, a five-point-scale on which *1* and *5* indicate *don't prefer* and *prefer*, respectively. One alternative is a ranking method. Users' preference patterns are obtained in the form of response orders, which are lists of objects sorted according to the degrees of the users' preferences. We previously called a CF framework incorporating this ranking method, *Nantonac Collaborative Filtering*[1] [4]; using it, more appropriate recommendations could be performed. However, it was not as advantageous if the length of response orders was short, because it becomes difficult to evaluate similarities between users' preferences. Such short responses can be easily collected using Joachims' procedure [3]. We therefore wanted to improve recommendations in such a condition by introducing a filling-in technique.

We describe filling-in methods for orders in Section 2 and a nantonac CF task in Section 3. Section 4 and 5 show our experimental results and our conclusions, respectively.

2 Filling-in Missing Objects in Orders

We first describe our basic notations. x_j denotes an object, entity, or substance to be sorted. The universal object set, X^*, consists of all possible objects. The order is denoted by $O{=}x_1{\succ}x_2{\succ}\cdots{\succ}x_3$. The meaning of the order, $x_1{\succ}x_2$, is "x_1 precedes x_2." The object set 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. $X_i = X^*$, is called a complete order; otherwise, it is an incomplete order. The 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.,

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

[1] The word *nantonac* originates from Japanese, *nantonaku*, which means "unable to explain specifically, but I think such and such is the case."

Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)
0-7695-2142-8/04 $ 20.00 IEEE

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 $x_a, x_b \in X_1 \cap X_2, x_a \neq x_b$. The distance, $d(O_a, O_b)$, is defined between two orders consisting of the same objects, that is, $X_a = X_b (\equiv X)$. *Spearman's distance* $d_S(O_a, O_b)$ [5] is a typical dissimilarity; that is defined as the sum of the squared differences between ranks. We adopted Spearman's distance, because its statistical properties have been well studied and its computational complexity is relatively small. By normalizing the distance range to be $[-1, 1]$, the *Spearman's rank correlation* ρ is defined as

$$\rho = 1 - 6 \times d_S / (|X|^3 - |X|). \quad (1)$$

We then describe the task of filling-in missing objects. To calculate the distance between two orders, we use

$$O_a = x_1 \succ x_3 \succ x_6 \text{ and } O_b = x_5 \succ x_3 \succ x_2 \succ x_6. \quad (2)$$

Distance is defined between two orders consisting of the same objects, but X_a and X_b differ; that is to say, both orders include missing objects. The missing objects for O_a are $\tilde{X}_a = \{x | \, x \notin O_a \wedge x \in X_b\} = \{x_2, x_5\}$, and those for O_b are $\{x_1\}$. Hence, it is not possible to directly calculate the distance. One way to overcome this difficulty is to ignore the missing objects and to calculate the distance over common objects, i.e., $X_a \cap X_b$. For example, by ignoring missing objects, both O_a and O_b are converted into $x_3 \succ x_6$; accordingly, the Spearman's distance $d_S = 0$. But since useful information might be contained in these ignored objects, ignoring them would lessen the precision or the confidence in the calculation of distances. Moreover, if $X_a \cap X_b = \emptyset$, the distance cannot be obtained. If samples of orders are available, the ranks of such missing objects can be filled-in based on the summary statistics of the samples. Such techniques would be highly beneficial to the derivation of more appropriate distances. Note that, in this paper, we assume that objects are uniformly missing. For example, top-3-orders do not satisfy this assumption, because only the top three objects are observed and objects at the bottom portions of the order are always missing.

2.1 Traditional Filling-in Methods for Orders

In literature on psychological statistics, these missing values are commonly processed by considering a set of orders instead of a single order. We describe the notion of an *Incomplete Order Set* (IOS)[2] [5], which is defined as a set of orders that are concordant with the given incomplete order. Formally, let O be the order consisting of the object set X, and $\tilde{X}$ be the set of missing objects. An IOS is defined as

$$\text{ios}(O, \tilde{X}) = \{O'_i | O'_i \text{ is concordant with } O, X'_i = X \cup \tilde{X}\}.$$

[2] In the cited book, this notion is referred to by the term *incomplete ranking*, but we have adopted IOS to insist that this is a set of orders.

This idea is not fit for large-scale data sets because the size of the set is $|X'|!/|X|!$, which grows exponentially in accordance with $|X'|$. Furthermore, there are some difficulties in defining the distances between the two sets of orders. One possible definition is to adopt the arithmetic mean of the distances between orders in each of the two sets. However, this is not distance because $d(\text{ios}_a, \text{ios}_a)$ may not be 0.

To avoid the above difficulties in IOS, we proposed the idea of *default rank* [4]. The idea is to rank missing objects into the middle of the filled-in orders, since such ranks can be considered as being neutral. However, default ranks were not found to be effective. We had thought that the middle ranks in orders would represent neutral values, but this was not the case. For example, suppose that there is an object ranked at the lowest in almost all the sample orders. If this object was ranked at the middle, the filled-in order would indicate that the object is ranked relatively high.

2.2 A Default Order

We propose a new idea, *default orders*. In the case of numerical or nominal variables, missing values can be replaced with the summary statistics of samples, for example, the means or the modes. By analogy, we try to fill the ranks of missing objects by using the centers of orders in sample orders, S. The central order $\bar{O}_S$ [5] is defined as

$$\bar{O}_S = \arg\min_O \textstyle\sum_{O_i \in S} d(O_i, O).$$

Note that $\bar{O}_S$ is composed of objects $\bar{X}_S = \cup_{O_i \in S} X_i$. Except for a few special cases, deriving the strict central orders is not tractable. Hence, we employ the Thurstone's paired comparison method, which is based on the model of the Thurstone's law of comparative judgment [9]. This model sorts objects, x_j, according to their utilities, which follow the normal distribution $N(\mu_j, \sigma)$. By applying the least square method to this model [6], the μ'_j (a linear transformation of the μ_j) is derived as

$$\mu'_j = \tfrac{1}{|\bar{X}|} \textstyle\sum_{x \in \bar{X}_S} \Phi^{-1}\big(\Pr[x_j \succ x]\big), \quad (3)$$

where $\Phi(\cdot)$ is the normal distribution function. The probability that the object x_j precedes x, $\Pr[x_j \succ x]$, can be estimated by simply counting the ordered pairs appearing in the sample set. We approximate the central order by sorting objects according to the corresponding μ'_j.

A default order is an order that is concordant with a central order and is composed of missing objects. For example, supposing the orders in Equation (2) are given, and let the central order of samples be $x_1 \succ x_5 \succ x_2 \succ x_3 \succ x_4 \succ x_6$. The missing objects for O_a and for O_b are $\{x_2, x_5\}$ and $\{x_1\}$, respectively. Accordingly, the default order for O_a is $\tilde{O}_a = x_5 \succ x_2$. Similarly, that for O_b is $\tilde{O}_b = x_1$. We propose to fill-in the ranks of missing objects by using these default

orders. For this purpose, the observed order and its default order have to be merged. By definition, no objects are commonly included in both orders; thus, the traditional merging methods for orders cannot be applied. We hence propose a new merging method based on order statistics.

Consider the case that the observed order O and its default order $\tilde{O}$ are merged into the filled order O'. These three orders respectively consist of object sets, X, $\tilde{X}$, and X'. By definition, $X' = \tilde{X} \cup X$ and $X \cap \tilde{X} = \emptyset$. Instead of directly modeling this merging process, we consider the division process. Because we assumed that objects are uniformly missing, suppose that $|O|$ of objects are uniformly sampled from objects in the X' without replacement. These are then sorted so as to be concordant with the O', so that O is obtained. In this case, for the i-th object $x_{i:O}$ in O, the expectation of ranks in O' becomes

$$\mathrm{E}[r(O', x_{i:O})] = i \times \frac{|O'|+1}{|O|+1},$$

according to [1]. Similarly, for the j-th object in $\tilde{O}$, the expectation is $j(|O'|+1)/(|\tilde{O}|+1)$. We assigned these expectations of rank to the objects in X and $\tilde{X}$; then O' is formed by sorting according these expectations. In the example of Equation (2), $O_a{=}x_1{\succ}x_3{\succ}x_6$ and its default order $\tilde{O}_a = x_5{\succ}x_2$ are merged. To the second object x_3 in O_a, the expected rank $2 \times (5{+}1)/(3{+}1) = 3$ is assigned. These expectations are assigned to all the remaining objects in a similar way. Consequently, by sorting objects according to these expectations, we obtain the order $O'_a{=}x_1{\succ}x_5{\succ}x_3{\succ}x_2{\succ}x_6$. Similarly, $O'_b{=}x_5{\succ}x_3{\succ}x_1{\succ}x_2{\succ}x_6$. This filling-in technique is very simple; thus, its computational complexity is small, $O(\max(|X'|, |\tilde{X}|\log|\tilde{X}|))$, if the central order is calculated in advance. Hence this filling-in method can be applied to large-scale data.

3 Nantonac CF with Filling-in Objects

We built the above filling-in technique based on default orders into our nantonac CF method.

We first describe a *nantonac collaborative filtering* task [4]. The aim of a CF task is to predict the preferences of a particular user (an active user) based on the preference data collected on other users (a user DB). The system shows a set of objects, X_i, to the user i, and the user sorts these objects according to his/her preferences. The sorted sequences are denoted by $O_i{=}x_1{\succ}x_2{\succ}\cdots{\succ}x_{|X_i|}$. The user DB, $D_S = \{O_1, \ldots, O_{|D_S|}\}$, is a set of all O_i. Sample users are people who provided orders in the DB. Let O_a be the order sorted by the active user, and the order is composed of objects in X_a. Given O_a and D_S, the goal of a nantonac CF task is to estimate which of the objects are preferred by the active user.

A simple correlation method (SCR) [4] is a basic method for performing a nantonac CF task. In this method, objects are recommended to active users through almost the same process as that used in the GroupLens [8]. Rating scores are simply substituted by ranks $r(O_i, x_j)$, which is the rank of object j in the sample order of the user i. The system estimates the active user's preferences for the object j by the function:

$$\hat{r}_{aj} = \frac{\sum_{i\in I_j} R_{ai}\left(r(O_i,x_j)-\bar{r}_i\right)}{\sum_{i\in I_j}|R_{ai}|}, \qquad (4)$$

where $\bar{r}_i$ denotes the mean ranks over $X_{ai}{=}X_a{\cap}X_i$. I_j is a set of indices of sample users who evaluated the object j; i.e., $\{i|\, O_i{\in}D_S \text{ s.t. } x_j{\in}X_i\}$. R_{ai} is a Pearson correlation between the ranks of the active user and the user i concerning objects in X_{ai}. The objects are sorted in ascending order of estimated preferences, and highly ranked objects are recommended. Note that the objects missing in the other order are ignored, but the ranks are not renumbered. For example, for $O_a{=}x_1{\succ}x_2{\succ}x_3$ and $O_i{=}x_3{\succ}x_1$, the object x_2 in O_a is missing in O_i, so x_2 is ignored. However, the rank of x_3 remains $r(O_a, x_3){=}3$, not 2. Hence, this correlation is different from Spearman's ρ.

In accordance with the decrease of $|X_i|$ relative to $|X^*|$, the frequency of the event $X_a{\cap}X_i{=}\emptyset$ increases. Because R_{ai} is always 0 in such cases, the similarities between users can no longer be precisely measured, and an inappropriate recommendation will be made. Improving recommendations for short response orders, especially those with a length of two, is an important objective. One of the obstacles to performing CF is that users often take the trouble of representing their preferences. To overcome this obstacle, Joachims proposed a method for collecting preference orders of length two [3].

In the case of CF adopting the SD method, Breese et al. [2] proposed to fill-in missing scores and to calculate the correlation between users' responses over $X_a{\cup}X_i$, not $X_a{\cap}X_i$. We introduce this idea into the nantonac CF method. The procedures are the same as the original SCR except for filling-in missing objects of O_a and O_i. The central order $\bar{O}_S$ over the user DB is derived in advance. Before calculating the correlation R_{ai}, missing objects in the response orders are filled-in by using $\bar{O}_S$ as the default order. The Spearman's rank correlations between filled orders are used as R_{ai}. Note that filled orders are used only for calculating R_{ai}. The $\hat{r}_{aj}$ of Equation (4) is derived from the R_{ai} between filled orders and the original rank $r(O_i, x_j)$.

4 Experiments

To test the efficiency of using default orders, the above CF methods were applied to preference data in sushi. Data collection and experimental procedures were the same as

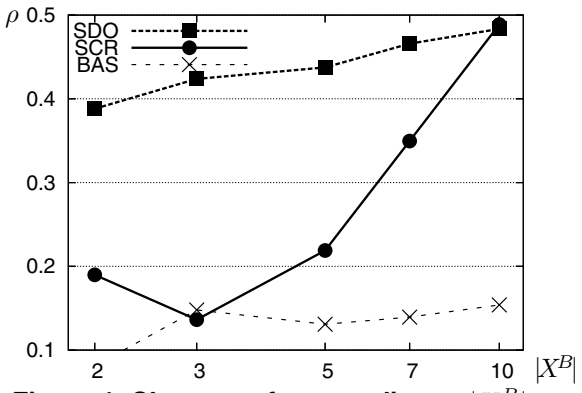

Figure 1. Changes of ρ according to $|X^B|$

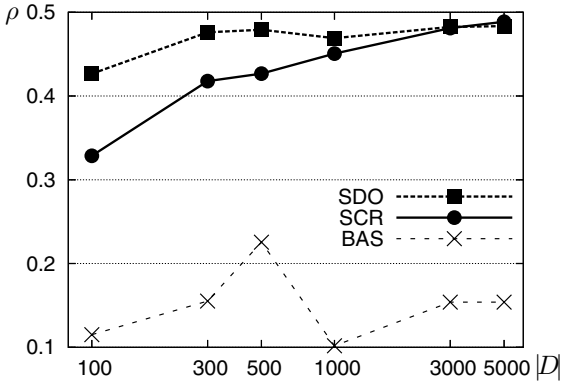

Figure 2. Changes of ρ according to $|D|$

those of [4], except that the data size was expanded to 5000. The quality of the recommendation was measured by means of the Spearman's ρ (Equation (1)) between the estimated and the true preference order. Note that a larger ρ indicates a better estimation.

We compared the nantonac CF methods using default orders (SDO) with two baselines. One was the original simple correlation method (SCR) in [4]. The other was a non-personalized recommendation (BAS); for all active users, the objects were sorted so as to be concordant with the central orders of the user DB. That is to say, the central orders were treated as a best-seller list, and popular objects were recommended.

Figure 1 shows the changes according to the lengths of the response orders $|X^B|$ when fixing $|D|$=5000 (the size of data sets). The SDO method was clearly superior to the SCR, and the differences were statistically significant, if $|X^B| \leq 7$. The shorter the response orders were, the more inappropriately the similarities between users were evaluated in the case of the SCR. Therefore, the SDO was remarkably effective for the shorter orders relative to the SCR. Figure 2 shows the changes according to the sizes of the data sets $|D|$ when fixing $|X^B|$=10. The SDO was superior to the SCR if $|D| \leq 500$, and the differences were significant if $|D|$=300, 500. If sample sizes decrease, the number of sample users that rank the objects commonly ranked by the active user ranked decrease. The SCR therefore becomes ineffective. However, since such sample users disappear, the SDO could make better recommendation.

We then observed results for the other baseline, the BAS method. If all users had a shared preference, the central order itself would have provided better recommendations. However, this non-personalized method was apparently worse than the SDO method. This indicates that the advantage of the SDO was not due to the specific characteristics of the users' sharing common preferences, but arose from the ability of the SDO method to provide well-personalized recommendations.

5 Conclusions

In this paper, we have proposed the notion of default orders to fill-in missing objects in orders. The performance of nantonac CF was improved by using default orders.

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.

References

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

[2] J. S. Breese, D. Heckerman, and C. Kadie. Emprical analysis of predictive algorithms for collaborative filtering. In *Uncertainty in Artificial Intelligence 14*, pages 43–52, 1998.

[3] 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.

[4] 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.

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

[6] 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.

[7] C. E. Osgood, G. J. Suci, and P. H. Tannenbaum. *The Measurement of Meaning*. University of Illinois Press, 1957.

[8] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: An open architecture for collaborative filtering of Netnews. In *Proc. of The Conf. on Computer Supported Cooperative Work*, pages 175–186, 1994.

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