Nantonac Collaborative Filtering

Recommendation Based on Multiple Order Responses

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

Shotaro Akaho — (affiliation as previous author)
s.akaho@aist.go.jp, http://staff.aist.go.jp/s.akaho/

keywords: recommender system, order, ranking method, collaborative filtering

Summary

Recommender systems use collaborative filtering to recommend objects by summarizing the preference patterns of people who have similar patterns to the target user. Traditionally, these preference patterns are represented by rating scores. We developed recommendation methods using order responses instead of rating scores, and showed the advantages of using orders. However, there was a problem with this framework, in that the number of objects per user is limited, because it is generally difficult for users to sort tens or hundreds of items at the same time. To overcome this limitation, we extended this framework to allow multiple orders to be collected from each user.

1. Introduction

A recommender system suggests the objects that users are expected to prefer [Ben Schafer 01]. Collaborative filtering (CF) is an algorithm that implements this recommender system by automating the word-of-mouth paradigm. To carry out this CF, users' preference patterns must be captured. Almost all of the CF techniques adopt the Semantic Differential (SD) method [Osgood 57], in which users' preference patterns are obtained by using a scale on which extremes are represented by antonymous words. One example is a five-point-scale on which the numbers "1" and "5" indicate "don't prefer" and "prefer," respectively. To treat the acquired rating scores as interval values, unsafe assumptions must be introduced, as shown in Section 1·1.

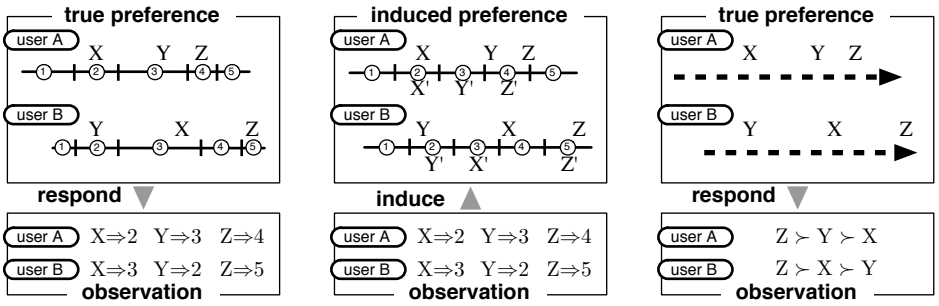
We therefore advocate a method called **Nantonac Collaborative Filtering**[*1] [Kamishima 03], which is a CF framework adopting a ranking method in which users' preference patterns are represented by **orders**. An order is an object sequence which is sorted according to the degree of users' preferences. We showed that the precision of recommendation could be improved by introducing a ranking method. In this paper, we extend this nantonac CF framework so that it can deal with multiple response orders per user.

We describe an SD-based method and our original nantonac CF method in Chapter 2. To demonstrate the advantages of adopting a ranking method, we performed preliminary experiments, which are described in Chapter 3. In Chapter 4 and Chapter 5, we present our extended nantonac CF methods and experimental results, respectively. Chapter 6 summarizes our conclusions.

1·1 Related work and Motivation

Collaborative filtering is generally performed in three steps: (1) obtaining users' preference data, (2) predicting users' preference patterns, and (3) recommending items to users. In almost all research on CF, attempts have been made to improve the prediction accuracy or computational efficiency in step (2) [Breese 98, Herlocker 99]. There have also been several studies on the problems in step (3). Herlocker et al. [Herlocker 00] pointed out that explaining the reason for making recommendations increases users' confidence in the CF system. Ziegler et al. [Ziegler 05] proposed a technique of "topic diversification" to increase the serendipity of recommendations.

*1 The word "nantonac" originates from a Japanese word, "nantonaku," which means "just somehow." For example, in Japanese, if I say "I *nantonaku* understand something," I am saying that I cannot specifically explain why I understand it, but that I somehow do.



(a) Observations measured by an SD method (b) Degrees of preference induced from the rating scores (c) Observations measured by a ranking method

Fig. 1 Mappings between the degree of preference and observations

The data collection step (1) should be examined together with the other steps, but this important issue has been almost completely ignored. Users' preference data are collected implicitly or explicitly [Ben Schafer 01]. In the case of implicit collection, the degrees of preference are expected to be based on the users' activity, such as browsing time or purchasing actions. Another way to collect preference data is to explicitly ask users for this information. In this case, to our knowledge, a semantic differential (SD) method [Osgood 57] has been adopted in all CF systems. However, this SD method has the following defects. In Figure 1(a), we intuitively show the process by which ratings are captured using the SD method. The degrees of preference and observed rating scores are shown in the upper and lower panels of the figure, respectively. We here discuss the mappings from degrees to observation. For example, if the degree of preference for the object X is in the interval corresponding to the score 2, the user A will respond with a rating score of 2. When measuring a physical quantity like length or weight, the mapping can be defined based on an objective and invariant criterion such as the speed of light or the kilogram prototype. However, it is not possible to quantitatively measure the degrees of preference, impression, or sensation, and thus each user of necessity uses his/her own mapping based on a subjective criterion. Therefore, the mappings become different between users A and B, and thus the observed ratings are not compatible. Accordingly, to overcome the above incompatibility of rating scales, two assumptions are generally introduced [Nakamori 00]: (1) the total lengths of all the scales are equal, and (2) all intervals within the same scale are equal. In this case, a change in the mappings is forced, and thus there may be a deviation between the true degrees of preference and the degrees of preference induced from the observations. For example, in Figure 1(b), the degrees of preference for the objects X, Y, and Z deviate to X', Y', and Z', respectively.

To avoid this problem, we advocated **Nantonac Collaborative Filtering** [Kamishima 03], in which the degree of users' preference is measured by using a ranking method. In this ranking method, a user responds by ordering objects according to his/her preference. Because the degrees of preference are relatively measured, as shown in Figure 1(c), it is not necessary to make the above unsafe assumptions in order to render the obtained observations compatible.

One might think that the outcomes of the ranking method are equivalent to orders to which rating scores are converted: For example, in Figure 1(a), the user A's ratings for the objects X and Y are 2 and 3, respectively. The order $Y \succ X$ can be obtained by converting these ratings. However, such converted orders are different from the ones obtained by the ranking method. As pointed out in [Luaces 04], a trained expert, e.g., a wine taster, can maintain a consistent mapping throughout a given session, and users' mappings generally change for each response. In the ranking method, this is not problematic, because objects are sorted simultaneously.

However, this ranking method has a defect, in that the length of order responses are limited, because it is impractical for users to sort tens or hundreds of objects simultaneously. With the previous nantonac CF framework, the number of responses was limited to one per user, and thus the total amount of preference data obtained was severely limited. To overcome this problem, we describe how we extended this nantonac CF framework so that multiple orders could be collected from each user. We propose several methods to deal with these multiple order responses and examine the performance of these methods.

2. Collaborative Filtering

We will now describe methods for SD-based CF and nantonac CF.

2·1 SD-based Collaborative Filtering

We will first describe the collaborative filtering method developed as part of the GroupLens project [Resnick 94], which adopts a SD method to measure users' preference. Collaborative filtering is a task to predict the preferences of a particular user (an active user) based on the preference data collected on other users (a user database). Formally, the task is defined as follows: Let X^* be a set of all objects $\{x_1, \ldots, x_{L^*}\}$, where L^* is the total number of objects. Let $Y^* = \{y_1, \ldots, y_{M^*}\}$ be a set of all users, where M^* is the total number of users. s_{ij} denotes the rating score given by the user $y_i \in Y^*$ to the object $x_j \in X^*$. The score represents the preference of the user, and takes one of a fixed finite set, for example, $\{1,2,3,4,5\}$. Note that the subscript j of s_{ij} doesn't means that "The j-th object is rated by user y_i," but that "The object is uniquely indexed by j in X^*." X_i denotes a set of objects the user y_i rated, and L_i is defined as $|X_i|$. $S_i = \{s_{ij} | x_j \in X_i\}$ denotes a set of rating scores assigned to the objects $x_j \in X_i$ by the user y_i. The user database, $D_S = \{S_i | y_i \in Y_S\}$, consists of score sets rated by the user in the user set $Y_S \subset Y^*$. Users in Y_S are called sample users. $y_a \in Y^* \backslash Y_S$ is an active user for whom the system will make recommendations. Let S_a be the set of scores rated by an active user, and X_a be the set of objects that an active user has already rated. Given the S_a and the D_S, the task of collaborative filtering is estimating the objects that the active user is expected to rate high. Such objects are then recommended to the active user.

The GroupLens' estimation method works as follows. First, the similarity between the active user and the sample user y_i is measured by the correlation:

$$R_{ai} = \frac{\sum_{x_j \in X_{a \cap i}} \left(s_{aj} - \bar{s}_a\right)\left(s_{ij} - \bar{s}_i\right)}{\sqrt{\sum_{x_j \in X_{a \cap i}} \left(s_{aj} - \bar{s}_a\right)^2} \sqrt{\sum_{x_j \in X_{a \cap i}} \left(s_{ij} - \bar{s}_i\right)^2}}. \quad (1)$$

$X_{a \cap i} = X_a \cap X_i$ is a set of objects that are rated by both an active user y_a and a sample user y_i. $\bar{s}_i$ is the mean score over the objects in X_i. The expected score $\hat{s}_{aj}$ of the object x_j for the active user is

$$\hat{s}_{aj} = \tilde{s}_a + \frac{\sum_{y_i \in Y_j} R_{ai}\left(s_{ij} - \tilde{s}_i\right)}{\sum_{y_i \in Y_j} |R_{ai}|}. \quad (2)$$

$\bar{s}_i$ is slightly different from $\bar{s}_i$. This $\bar{s}_i$ is the mean score not over $X_{a \cap i}$, but over X_i. Y_j is a set of users who rated the object x_j, i.e., $\{y_i | s_{ij} \in S_i, S_i \in D_S\}$. The system recommends the objects for which the expected score $\hat{s}_{aj}$ is high. Note that in paper [Resnick 94] it was stated that, "All the summations and averages in the formula are computed only over those articles that Ken and Lee both rated." However, in equation (2), we use $\tilde{s}_i$ instead of $\bar{s}_i$, because we obtained a slightly better experimental result.

2·2 Nantonac Collaborative Filtering

We here show **Nantonac Collaborative Filtering** [Kamishima 03], which is the CF framework incorporating the ranking method. To show the effectiveness of adopting a ranking method, we applied the same method as with the GroupLens', except for the users' response orders. Formally, the nantonac CF is stated as follows.

In the case of nantonac, the system shows a set of objects, X_i, to the user y_i, and the user sorts them according to the degree of his/her preference. The sorted order is denoted by $O_i = x_a \succ \cdots \succ x_j \succ \cdots \succ x_b$. The order $x_1 \succ x_2$ represents "x_1 is preferred to x_2." The length of order L_i is equivalent to the size of X_i. The subscript j of x means that "The object is uniquely indexed by j in X^*." Rank, $r(O_i, x_j)$, or simply r_{ij} is the cardinal number that indicates the position of the object x_j in the order O_i. The user DB is a set of orders sorted by all the sample users, $D_O = \{O_i | y_i \in Y_S\}$. Let X_a be a set of objects sorted by the active user, and O_a be his/her order response. Given the O_a and D_O, the task of nantonac CF is to estimate the preference pattern of the active user. We call this original problem the **Single-Single Nantonac CF (SS-NCF)** to differentiate it from the extended versions in Chapter 4. This is because both the active user and sample users provide a single order response per user.

Applying the GroupLens' method to this nantonac CF problem is very simple: the rating scores s_{ij} in equation (1) are replaced with ranks r_{ij}:

$$R_{ai} = \frac{\sum_{x_j \in X_{a \cap i}} \left(r_{aj} - \bar{r}_a\right)\left(r_{ij} - \bar{r}_i\right)}{\sqrt{\sum_{x_j \in X_{a \cap i}} \left(r_{aj} - \bar{r}_a\right)^2} \sqrt{\sum_{x_j \in X_{a \cap i}} \left(r_{ij} - \bar{r}_i\right)^2}}, \quad (3)$$

and equation (2) becomes

$$\hat{r}_{aj} = \tilde{r}_a + \frac{\sum_{y_i \in Y_j} R_{ai}(r_{ij} - \tilde{r}_i)}{\sum_{y_i \in Y_j} |R_{ai}|}, \quad (4)$$

where $\bar{r}_i$ and $\tilde{r}_i$ are the mean rank over the objects in $X_{a\cap i}$ and X_i, respectively. A set of candidate objects are sorted in ascending order of the corresponding $\hat{r}_{aj}$, and the objects ranked highly are recommended.

It is worth mentioning why the rating scores can be replaced with the ranks in response orders. According to [Arnold 92], assuming that the objects in X^* are missed uniformly at random from a hidden order O^*_h, incomplete orders O_i are observed. The conditional expectation of ranks of the object $x_j \in X_i$ in the order O^*_h given O_i is

$$\mathrm{E}[r^*_j|O_i] = r(O_i, x_j)\frac{L^*+1}{L_i+1}, \tag{5}$$

where $r^*_j \equiv r(O^*_h, x_j)$ and $L^* = |X^*|$ and $L_i = |X_i|$. Because L^* is constant, if L_i is constant for all $y_i \in Y^*$, $\mathrm{E}[r^*_j|O_i]$ is proportional to $r(O_i, x_j) \equiv r_{ij}$. Two similarities between users, equations (1) and (3), are almost the same. However, the former denotes Pearson's correlation between two users' ratings scores, while the latter approximates Spearman's rank correlation (see equation (6)) between hidden complete response orders of two users. Furthermore, equation (2) represents the estimated rating score, while equation (4) can be considered the estimated rank in O^*_i up to scaling and shifting. Note that if L_i is not constant, r_{ij} in equations (3) and (4) should be replaced with $r_{ij}/(L_i+1)$.

We here describe some definitions related to orders. The distance, $d(O_a, O_b)$, is defined between two orders consisting of the same objects, that is, $X{\equiv}X_a{=}X_b$. One such distance is *Spearman's distance* $d_S(O_a, O_b)$ [Marden 95], which is defined as the sum of the squared differences between ranks; $d_S(O_a, O_b) = \sum_{x_j \in X}(r_{aj} - r_{bj})^2$. By normalizing the distance, the *Spearman's rank correlation* is defined as

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

2·3 Modified Mean Expected Rank

Before showing an improved version of the above basic CF methods, we will give a definition of a central order and its derivation method. This method is also used for dealing with multiple response orders in Chapter 4.

Given a set of orders $T = \{O_i | i = 1, \ldots, |T|\}$, a central order is the order that consists of objects $X_T = X(O_1) \cup \cdots \cup X(O_{|T|})$, and the sum of distances from the order to the constituents in T is the minimum. Formally,

$$\bar{O}_T = \arg_O \text{ s.t. } \min_{X(O)=X_T} \sum_{O' \in T} d(O', O), \tag{7}$$

where $d(O', O)$ is the distance between orders, e.g., Spearman's distance, over the objects in $X(O') \cap X(O)$. To derive central orders efficiently, we advocate a **Modified Mean Expected Rank (MMER)** method. Let $T(x_j)$ be the set of orders in T such that the object x_j is included, i.e., $T(x_j) = \{O_i : x_j \in X(O_i), O_i \in T\}$. Mean expected rank, $\mathrm{MER}(x_j)$, is defined as the mean of expected ranks (see equation (5)) of objects x_j over $T(x_j)$, i.e., $(1/|T(x_j)|)\sum_{O_i \in T(x_j)} \mathrm{E}[r^*_j|O_i]$. It would be reasonable to derive a central order by sorting objects $x_j \in T$ according to the corresponding the $\mathrm{MER}(x_j)$. However, information about the size of $T(x_j)$ is not taken into account. Between two objects x_a and x_b such that $|T(x_a)| > |T(x_a)|$, the value of $\mathrm{MER}(x_a)$ would be more reliable than that of $\mathrm{MER}(x_b)$. In such a case, if $\mathrm{MER}(x_a){=}\mathrm{MER}(x_b)$, which object should be ranked higher? Because the middle position of an order is considered providing no information, the object x_a whose $|T(x_a)|$ is larger should be ranked farther from the middle of a central order. To implement this idea, we introduce a "default rank", that is the mean rank given no sample order, i.e., $(L^*+1)/2$. MMER is defined as a mean of ranks over expected ranks in $T(x_j)$ together with one default rank. Formally, the $\mathrm{MMER}(x_j)$ is defined as

$$\mathrm{MMER}(x_j) = \frac{L^*+1}{|T(x_j)|+1}\left(\frac{1}{2} + \sum_{O_i \in T(x_j)} \frac{r_{ij}}{L_i+1}\right), \tag{8}$$

where $\mathrm{E}[r^*_j|O_i]$ is equation (5). In the MMER method, these $\mathrm{MMER}(x_j)$ are calculated for each $x_j \in X(\bar{O}_T)$, and a central order is then derived by sorting the objects in ascending order of $\mathrm{MMER}(x_j)$. Because these MMER can be calculated by scanning all the orders in T, only the time $O(\sum_{O_i \in T} L_i)$ is required. By adding the sorting time $O(|X(\bar{O}_T)| \log |X(\bar{O}_T)|)$, a central order can be derived.

2·4 Filling Missing Scores and Ranks

We describe an improved method for settling the problems stemming from the sparseness of ratings. As pointed out in [Breese 98], the similarities between users, equations (1) and (3), are derived based on the objects rated by both an active user and a sample user. However, such objects don't exist with high probability, since generally $|X_a|, |X_i| \ll |X^*|$. To

Table 1 Comparison between SD-based and nantonac CF

method	(A) N:5000, L:10	(B) N:500, L:10	(C) N:5000, L:5
nantonac	0.489	0.427	0.219
SD-based	0.438	0.369	0.195
shift	0.437	0.369	0.194
z-score	0.445	0.372	0.202
min-max	0.445	0.369	0.202
rank	0.444	0.379	0.205
RC	0.437	0.370	0.194

cope with this problem, we proposed a filling technique designed for nantonac CF in [Kamishima 04]. In this technique, the objects that are not sorted by either user are filled by a central order of D_O. We here show a slightly modified version using MMER in Section 2·3. Simply speaking, all ranks r_{ij} are replaced with $\mathrm{E}[r^*_j|O_i]$, and missing ranks are filled by $\mathrm{MMER}(x_j)$. When calculating equation (3), if x_j is sorted by the user y_i, i.e., $x_j \in S_i$, we replace r_{ij} with equation (5), i.e., $r_{ij}\frac{L^*+1}{L_i+1}$. If x_j is not sorted by y_i, i.e., $x_j \notin S_i$, r_{ij} is replaced with $\mathrm{MMER}(x_j)$. Note that the constant L^* is unknown, but this is canceled out in equation (3). Also, the summations are calculated not over $X_{a\cap i} = X_a \cap X_i$, but over $X_a \cup X_i$. Equation (4) is calculated from these R_{ai} and the original r_{ij}, and the candidate objects are sorted according to $\hat{r}_{ij}$.

3. Experiments on SS-NCF

To demonstrate the merits of adopting a ranking method, we show the experimental results in [Kamishima 03], and additional results on the conditions where SD scores are normalized. We then show the effectiveness of filling missing scores or ranks.

3·1 Comparison between SD-based and Nantonac CF

We used the same data set as those of [Kamishima 03], except that the data size was expanded to 5000(=N). We extracted two object sets X^A and X^B, of which the size was 10(=$|X_i|$), from 100(=$|X^*|$) possible objects. The objects in each set were sorted by 5000 users, and we then obtained two sets of orders, $\{O^A_i\}^N_{i=1}$ and $\{O^B_i\}^N_{i=1}$, that were used for testing and training, respectively. The object sets X^B_i were also evaluated by a SD-method, and the corresponding rating score sets are denoted by $\{S^B_i\}^N_{i=1}$. The prediction accuracy of the CF methods was evaluated by the following cross validation procedure: In each fold of test, the order/score set, $\{O^B_i\}/\{S^B_i\}$, was divided into two sets, and one set was considered to be the user DB D_O/D_S*2. Additionally, orders/scores were sequentially picked from the other set and treated as an active users' response O_a/S_a. From these D_O/D_S and O_a/S_a, we estimated the preference orders $\hat{O}^A_a$ for the objects in X^A. The estimation performance was measured by Spearman's ρ (Equation (6)) between the estimated order $\hat{O}^A_a$ and the response order $O^A_a \in \{O^A_i\}$. The larger ρ indicates the better recommendation.

We show the mean ρ in Table 1. The columns (A), (B), and (C) show the means of ρ when the data set size, N, and the length of response orders, L, are changed. In the first and second rows, we show the results derived by a nantonac method in Section 2·2 and an SD-based method in Section 2·1, respectively. In all conditions, a nantonac CF method performed better, and the differences were statistically significant. Hereafter, the term "significant" means that the difference between the means of ρ is statistically significant at the level of 1% by a paired t-test. In addition, the larger N or L was, the more significant the difference was. When fixing L=10, the differences were significant if $N \geq 300$. When fixing N=5000, the differences were significant if $L \geq 5$. As described in Section 1·1, since we can measure preferences based on a compatible scale by a ranking method, response orders are collectively more informative than ratings.

We additionally show that the normalization of scores cannot help to overcome this advantage. In [Herlocker 99], it was shown that the normalization of scores can improve the prediction performance. Before applying an SD-based CF algorithm, we normalized the rating scores by the following procedures.

shift: s_{ij} was normalized to $s_{ij} - \tilde{s}_j$, where $\tilde{s}_j$ is the mean over S_i.

z-score: After the above shift normalization, the scores were divided by the standard deviation over S_i.

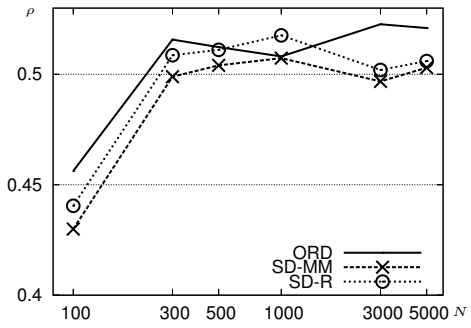
min-max: s_{ij} is normalized to $\frac{s_{ij} - \min_j\{s_{ij}\in S_i\}}{\max_j\{s_{ij}\in S_i\} - \min_j\{s_{ij}\in S_i\}}$.

rank: After the objects in S_i are sorted in ascending order of the corresponding scores, each score is converted to the ranks in the order.

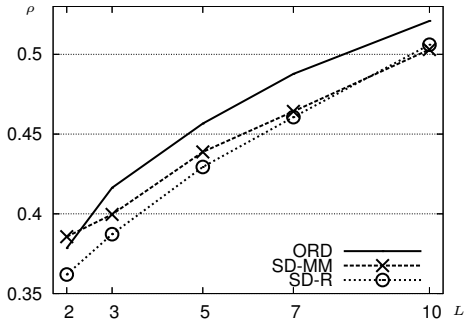
RC: As R_{ai}, we used rank correlation instead of Pearson's correlation.

In the third to seventh columns, we show the means of ρ derived by each normalization technique. As pointed out in [Herlocker 99], these normalizations

*2 Because we performed 10-fold cross validation, $|D_O|/|D_S|$ became $9N/10$



(a) Variations of means in the size of data sets



(b) Variations of means in the length of response orders, L

Fig. 2 Comparison with filling and normalizing techniques

were effective for improving prediction accuracies. However, the nantonac CF was still superior to these SD-based methods adopting score normalization. In addition, the differences were statistically significant. These experimental results support our claim that the SD scores cannot completely be normalized due to the reasons described in Section 1·1.

3·2 Filling Missing Scores or Ranks

In this section, we applied both normalization and filling methods. In Figure 2, we show the results when the size of the data sets, N, and the length of the response orders, L, were varied. The curves labeled ORD is the result derived by a nantonac CF method with the MMER filling technique. For SD-based methods in which missing scores are filled by the means over scores to the target object, i.e., $\{s_{i'j}|y_{i'} \in Y_j\}$. Additionally, two normalization techniques, the min-max and rank, are applied, and the corresponding results are denoted by SD-MM and SD-R. In Figure 2, we show the means of ρ derived by these three methods. In Figure 2(a) and Figure 2(b), we show the results when varying the size of the data sets and length of the response orders, respectively. In Figure 2(a), ORD is significantly better than the two SD-based methods if $N \geq 3000$. In Figure 2(b), ORD was significantly better than SD-R over all L, and was significantly better than SD-MM if $L \geq 3$. However, ORD is significantly worse if $L=2$. Information captured by the ranking method severely lessens if $L=2$. Even in such a disadvantageous case, the ORD was better than the SD-R. In addition, the ORD exceeded the SD-MM and SD-R in many cases. We consider these observations to strongly show the merits of nantonac CF.

4. Methods Designed for Multiple Response Orders

We extend the nantonac CF methods in Chapter 2 so as to be able to deal with multiple response orders.

We first extended the SS-NCF (Single-Single Nantonac CF) in Section 2·2 so that the active user still returns only one order per user, but the sample users are allowed to provide multiple responses per user. We call this extended version the **Single-Multi Nantonac CF (SM-NCF)**. Specifically, K_i object sets, $X_{ik} \subseteq X^*, k=1,\ldots,K_i$, are sequentially shown to the user y_i. Note that these object sets are generally different each other. He/She then returns the order response $O_{i1},\ldots,O_{i,K_i}$ for each object set. These K_i responses make a set Q_i. The user DB then becomes a set of orders sorted by all the sample users, $D_Q = \{Q_i|y_i \in Y_S\}$. In SM-NCF, the active user's response is a single order, O_a. Given the O_a and D_Q, the goal of the SM-NCF is to estimate the preference pattern of the active user.

This SM-NCF task is achieved as follows. Each order set in $Q_i \in D_Q$ is converted to its corresponding central order. That is to say, objects $X_{Q_i} = X_{i1} \cup \cdots \cup X_{i,K_i}$ are sorted in ascending order of MMER(x_j) in Section 2·3. The resultant central order is denoted by $\bar{O}_i$. These central orders are collected into $D_Q^{\text{center}} = \{\bar{O}_i|y_i \in Y_S\}$, and the SS-NCF method is performed on this D_Q^{center} and O_a. Note that we equally assign the midrank [Marden 95, chapter 11] to the objects whose MMER values are equal.

We further extended this SM-NCF so that multiple order responses are allowed not only to sample users, but to an active user. This **Multi-Multi Nantonac CF (MM-NCF)** is formalized as follows: The active user sorts the objects in each object set, $X_{a1},\ldots,X_{a,K_a}$, and returns orders $Q_a = \{O_{ak}\}_{k=1}^{K_a}$, where K_a is the number of the active user's responses.

Table 2 SS-NCF vs SM-NCF for SUSHI data

Methods	2	3	5	7
SS-NCF	0.378	0.416	0.457	0.488
SM-NCF	0.519	0.529	0.535	0.527

Given the Q_a and $D_Q = \{Q_i | y_i \in Y_S\}$, the goal of the MM-NCF is to estimate the preference pattern of the active user.

We here describe two types of meta-methods to carry out the MM-NCF by using the above SM-NCF method as a subroutine. The first meta-method is **Pre-Centering**. Before performing the SM-NCF, the central order of the active user's order set Q_a is derived. By treating this central order $\bar{O}_a$ as a single response order, candidate objects are sorted by using the above SM-NCF method. The second is the **Post-Centering** method. For each order response $O_{ak} \in Q_a$, the same set of candidate objects X_q are sorted according to the degree of preference estimated by applying a SM-NCF method to O_{ak} and D_Q. Consequently, we obtain a set of estimated orders, $\hat{O}_{q1}, \ldots, \hat{O}_{q,K_a}$, that consist of objects X_q. Finally, the final preference order is obtained by calculating the central order of these estimated orders. Note that the post-centering is about K_a times slower than the pre-centering.

5. Experiments on SM-NCF and MM-NCF

We applied the methods in Chapter 4 to the real data set in Section 3·1. In this data set, only one response order per user could be used for estimating the preference patterns; thus, we could not perform the SM-NCF or MM-NCF experiments. Therefore, to simulate multiple order responses, each single response was converted into multiple response orders. For each response order O_i, sets of objects X_{ik} were randomly sampled without replacement. Note that sampling with replacement was used if $L_i > 5$. For each sampled set X_{ik}, the objects in the set were sorted so as to be concordant with the original response O_i; consequently, we got O_{ik}. The response order sets of user y_i were formed by collecting these orders. We used a notation such as $L \times K$, where L is the length of each response and K is the number of responses per user. 10-fold cross validation was performed.

In this section, we tested the SM-NCF method in Chapter 4. Hereafter, we used a nantonac CF method with a filling technique in Section 2·4. We first checked the minimum requirement "Could the estimation precision be improved by using multiple order responses on SUSHI data?" In Table 2, we showed the means of ρ estimated by the methods in Chapter 4. The 1st to 4th columns contained the means estimated by using responses whose length, L_i, were 2 to 7, respectively. As the active users' responses, the original orders of length $L_a = 10$ were used. The SS-NCF methods adopted single responses per sample user, while the other SM-NCF methods adopted two responses. Apparently, the SM-NCF methods were advantageous to the corresponding SS-NCF methods, especially if the lengths of the responses were short. It should be concluded that additional information collected from multiple responses was useful for improving the recommendation accuracy.

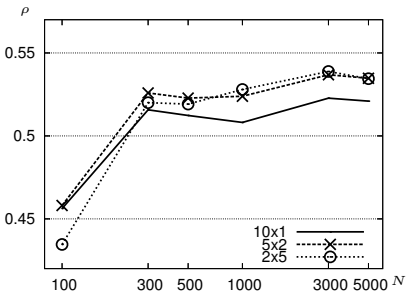


Fig. 3 Effects of divisions into multiple response orders

Next, by dividing a single response into multiple ones, some information was lost. For example, when O_i=$x_1 \succ x_2 \succ x_3 \succ x_4$ is divided into O_{i1}=$x_1 \succ x_3$ and O_{i2}=$x_2 \succ x_4$, the information "x_1 is preferred to x_2" was lost. Figure 3 showed variations of the mean ρ in the size of the data sets, N. SS-NCF denotes the result derived from the data set that consists of active and sample response orders whose lengths are L_a=10 and L_i=10, respectively. The curve labeled 2×5 represents the result obtained from data whose sample users' response sets were composed of 5 orders with lengths of 2. The 5×2 curve is similarly defined. Surprisingly, even if the sample orders were divided into multiple ones, the prediction performance was hardly degraded at all. The reason for this would be because lost information in a some sample user's response is compensated for another user's response. It would therefore be beneficial to collect multiple responses per sample user.

We carried out an experiment of two MM-NCF meta-methods in Chapter 4 by using the SM-NCF as a subroutine. The sizes of data sets N were 5000.

As a baseline, we chose the SS-NCF result under the condition L_a=10 and L_i=10. Because no information is lost by division into multiple responses, this result would be the upper limit. In this case, the mean ρ was 0.521. Next, we applied pre-centering and post-centering MM-NCF methods. The lengths of the responses L_a and L_i were 7, and the number of responses per active/sample user was 2, i.e., K_a=2 and K_i=2. We obtain the mean ρ's 0.514 and 0.499 by using the pre-centering and the pre-centering methods, respectively. Compared to these results with the upper baseline 0.521, the performance of the MM-NCF methods were slightly degraded. This is because active users' responses were divided, and information was lost. This result was contrasted with the results in Figure 3, in which the prediction accuracy was hardly degraded at all. The reason for this is because lost information in some sample users' responses can be compensated for that in by other sample users' responses, in the Figure 3 experiments. However, no information is available to compensate for the information lost in active users' response, in this experiment. When comparing the pre-centering and the post-centering, the former was significantly better. By using the pre-centering method, the length of the active users' responses became longer, and thus objects were more frequently contained in both the active and sample users' responses. Hence, the estimation performance could be improved. Furthermore, the pre-centering was computationally efficient. Consequently, the following method would be suggested for performing MM-NCF: First, for all the response sets, Q_a and $Q_i \in D_Q$ are converted to central orders $\bar{O}_a$ and $\bar{O}_i$, respectively. Then, a SS-NCF method is applied to this converted set.

6. Conclusions

We proposed the nantonac CF framework, in which the CF employs the ranking method. In the former framework, because exactly one order response per user was allowed, the total amount of preference patterns that could be collected from users was limited. Therefore, we extended this nantonac CF so that multiple responses could be collected from each user. We developed several methods for making recommendations based on these multiple order responses and empirically showed that these methods worked well. We plan to combine this pure collaborative filtering with content-based filtering, which is the recommendation method using the characteristics of objects.

Acknowledgments

This work is supported by the grants-in-aid 14658106 and 16700157 of the Japan society for the promotion of science.

♢ References ♢

[Arnold 92] Arnold, B. C., Balakrishnan, N., and Nagaraja, H. N.: *A First Course in Order Statistics*, John Wiley & Sons, Inc. (1992)

[Ben Schafer 01] Ben Schafer, J., Konstan, J. A., and Riedl, J.: E-Commerce Recommendation Applications, *Data Mining and Knowledge Discovery*, Vol. 5, pp. 115–153 (2001)

[Breese 98] Breese, J. S., Heckerman, D., and Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering, in *Uncertainty in Artificial Intelligence 14*, pp. 43–52 (1998)

[Herlocker 99] Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J.: An Algorithmic Framework for Performing Collaborative Filtering, in *Proc. of The 22nd Annual ACM SIGIR Conf. on Research and Development in Information Retrieval*, pp. 230–237 (1999)

[Herlocker 00] Herlocker, J. L., Konstan, J. A., and Riedl, J.: Explaining Collaborative Filtering Recommendations, in *Proc. of The Conf. on Computer Supported Cooperative Work*, pp. 241–250 (2000)

[Kamishima 03] Kamishima, T.: Nantonac Collaborative Filtering: Recommendation Based on Order Responses, in *Proc. of The 9th Int'l Conf. on Knowledge Discovery and Data Mining*, pp. 583–588 (2003)

[Kamishima 04] Kamishima, T. and Akaho, S.: Filling-in Missing Objects in Orders, in *Proc. of The 4th IEEE Int'l Conf. on Data Mining*, pp. 423–426 (2004)

[Luaces 04] Luaces, O., Bayón, G. F., Quevedo, J. R., Díez, J., del Coz, J. J., and Bahamonde, A.: Analyzing Sensory Data Using Non-linear Preference Learning with Feature Subset Selection, in *Proc. of the 15th European Conf. on Machine Learning*, pp. 286–297 (2004), [LNAI 3201]

[Marden 95] Marden, J. I.: *Analyzing and Modeling Rank Data*, Vol. 64 of *Monographs on Statistics and Applied Probability*, Chapman & Hall (1995)

[Nakamori 00] Nakamori, Y.: *Kansei data Kaiseki*, Morikita Shuppan Co., Ltd. (2000), (in Japanese)

[Osgood 57] Osgood, C. E., Suci, G. J., and Tannenbaum, P. H.: *The Measurement of Meaning*, University of Illinois Press (1957)

[Resnick 94] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews, in *Proc. of The Conf. on Computer Supported Cooperative Work*, pp. 175–186 (1994)

[Ziegler 05] Ziegler, C.-N., McNee, S. M., Konstan, J. A., and Lausen, G.: Improving Recommendation Lists Through Topic Diversification, in *Proc. of The 14th Int'l Conf. on World Wide Web*, pp. 22–32 (2005)