Nantonac Collaborative Filtering – A Model-Based Approach

Toshihiro Kamishima and Shotaro Akaho
National Institute of Advanced Industrial Science and Technology (AIST)
AIST Tsukuba Central 2, Umezono 1–1–1, Tsukuba, Ibaraki, 305–8568 Ja pan
mail@kamishima.net ⟨http://www.kamishima.net/⟩ and s.akaho@aist.go.jp

ABSTRACT

A recommender system has to collect users' preference data. To collect such data, rating or scoring methods that use rating scales, such as good-fair-poor or a five-point-scale, have been employed. We replaced such collection methods with a ranking method, in which objects are sorted according to the degree of a user's preference. We developed a technique to convert the rankings to scores based on order statistics theory. This technique successfully improved the accuracy of ranking recommended items. However, we targeted only memory-based recommendation algorithms. To test whether or not the use of ranking methods and our conversion technique are effective for wide variety of recommenders, we apply our conversion technique to model-based algorithms.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering

General Terms

Measurement

Keywords

Recommender System, Ranking Method, Order Statistics, Sensory Test

1. INTRODUCTION

A recommender system suggests items that a user would prefer. Collaborative filtering (CF) is an algorithm that implements this recommender system by automating the word-of-mouth paradigm. CF requires data that represent the degrees of a user's preferences in items, and a scoring or a rating method is widely used for collecting such data. In both methods, the system shows an item to a user and records the degree of the user's preference to the item on a scale. While a scoring method uses a numerical scale, e.g., a scale of 1 to 5, a rating method adopts ratings with ordered labels, e.g., {*good*, *fair*, *poor*}. The use of these measurement methods has been successful in performing CF. However, these methods have a few properties that are inappropriate when measuring a user's preferences, as we point out in the next section.

We have therefore advocated a framework called *Nantonac Collaborative Filtering*[1] [8, 9], which is a CF framework that adopts a ranking method to capture users' preference patterns. In a ranking method, a set of items is shown to a user, who ranks these items according to the degree of his/her preference.

In our previous work, we advocated a technique to convert ranking data to preference scores. Further, our experimental results showed that the relative degree of preference could be more accurately predicted from the converted scores. However, because we adopted only memory-based algorithms for recommendation, it was not clear whether the use of a ranking method and our conversion technique were useful for the other types of recommendation algorithms, namely model-based methods. Both memory- and model-based methods have their own pros and cons, and sophisticated model-based methods were developed and used in commercial systems [3]. We therefore tested whether or not the use of a ranking method is also beneficial to two model-based methods: pLSA [6] and matrix decomposition [12].

Our motivation in employing a ranking method is discussed in section 2. In sections 3 and 4, we present our model-based nantonac CF methods and experimental results, respectively. Section 5 summarizes our conclusions.

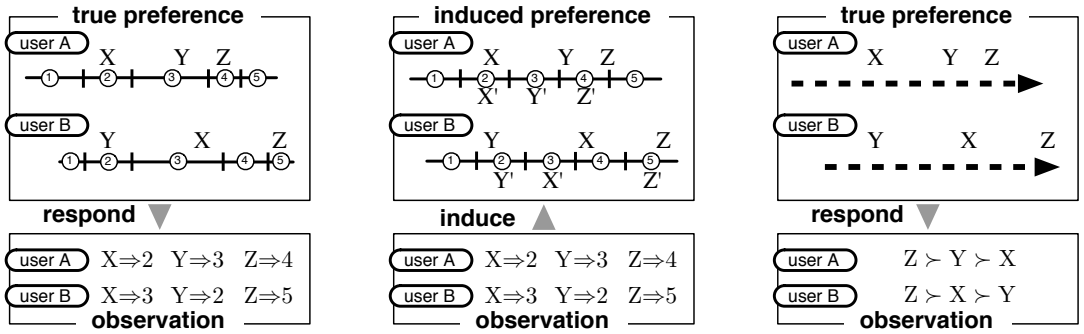
2. WHY IS A RANKING METHOD USED?

To accurately predict items that a user prefers, the precise measurement of a user's preference patterns is very important. For this measurement, a scoring or rating method has been adopted in almost all CF systems. To our knowledge, no other measurement methods, such as pairwise comparison, choice, or ranking methods, have been tested in CF.

We here show a weak point of a scoring method. In Figure 1(a), we intuitively show how scores are captured by a scoring method. The unobserved true preferences and observed scores are shown in the upper and lower panels of the figure, respectively. In the figure, because the true preference of item X is in user A's interval 2, user A will respond with a score of 2. If we are measuring a physical quantity, such as length or weight, the mapping from quantities to observed values can be defined based on an objective and invariant criterion, such as the speed of light or the kilogram prototype. However, when we measure a preference, it is difficult to share such an invariant and objective mapping between true preference and observed score, because each user uses his/her own mapping based on

[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.

RecSys2010, September 26–30, 2010, Barcelona, Spain.



(a) Mapping from true preferences to observed scores

(b) Backward mapping from observed scores to induced preferences

(c) Observations measured by a ranking method

Figure 1: Mapping between unobserved true preferences and observed scores

a subjective and variable criterion in his/her own mind. The mappings therefore tend to be inconsistent between users A and B. Further, when mapping back from the observed scores to the induced preference as in Figure 1(b), the system uses a common scale, and thus the induced degrees of preference might shift from the original. For example, item X lies in interval 3 of user B's mapping scale as in Figure 1(a). However, in Figure 1(b), B's true preference lies in interval 4 of the common mapping, while the induced preference (depicted by X' in the figure) deviates from the original degree. To avoid these defects, we advocated nantonac CF [8], in which a user's preference is measured by a ranking method. In this ranking method, a user responds by ordering objects according to his/her preference (Figure 1(c)). Mapping from true preference to observed order is simple: the more preferred items are ranked higher, and these mappings are common for all users.

One might think that these shifts in scores can be canceled by using calibration techniques. In [4], instead of the Pearson correlation, rank correlation is used to evaluate the similarity between two users. Further, rating scores are normalized by subtracting the mean of scores. According to our previous work [9], the adoption of a ranking method is advantageous even if these techniques are employed. This can be explained as follows. As pointed out in [13], only trained experts, e.g., wine tasters, can maintain a consistent mapping throughout a given session, and untrained users' mappings generally change for each response. It is known that users' responses are roughly correlated, but can drift slightly [5]. In a ranking method, this is not problematic, because only the simultaneously evaluated items are considered.

However, this ranking method has a few limitations. It is generally difficult to sort so many items at the same time. This limitation can be alleviated by sorting small multiple sets of items separately as in [9]. A scheme to collect ordered pairs was proposed in [7]. Another restriction is the lack of absolute information about preference. Because ranking methods merely provide a user's relative preferences, the system can predict which items are more preferred than compared items, but it cannot determine whether an item is absolutely preferred among all items. Therefore, a ranking method is inappropriate for a system displaying absolute ratings, e.g., five stars, but it is useful for estimating which choice is better to support users' decision making.

3. METHODS

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 (sample users). We first formalize a standard CF task using preferences captured by a scoring method. $x \in \{1, \ldots, n\}$ and $y \in \{1, \ldots, m\}$ denote a user and an item, respectively. s_{xy} denotes the score given by a user x to an item y. The score takes one of the values on a rating scale, such as a five-point scale, and represents the degree of preference. A training set consists of tuples, $\mathcal{D} = \{(x_k, y_k, s_{x_k y_k})\},\ k = 1, \ldots, N$. A set of items rated by user x is denoted by $\mathcal{Y}_x = \{y | (x' = x, y, s) \in \mathcal{D}\}$. Given the set $\mathcal{D}$, we want to derive a function, $\hat{s}_{xy} = f(x, y)$, that predicts a preference score for any pairs of a user x and an item y.

We move on to a nantonac CF incorporating a ranking method. In the case of nantonac CF, the system shows a set of items, $\mathcal{Y}_x$, to user x, who sorts them according to the degree of his/her preference. The sorted order is denoted by $O_x = y_{l_1} \succ \cdots \succ y_{l_j} \succ \cdots \succ y_{l_{|O_x|}}$, where $|O_x|$ is the length of an order O_x. The order $y_1 \succ y_2$ means that "y_1 is preferred to y_2." The j-th item, y_{l_j}, is an element of $\mathcal{Y}_x$.

We proposed a technique to extend CF methods that targeted scores in order to enable to deal with preference orders. In this technique, each preference order is converted into a set of scores based on the following theorem. We assume the existence of the unobserved complete preference order, O^*, which is generated by sorting all the items $\{1, \ldots, m\}$ according to the user's preference. Then, a portion of the items are sampled uniformly at random from this order, and these items are sorted so as to be concordant with this complete order. This resultant order is treated as the user's response order, O. We here denote the rank of an item y in an order O by $r(y, O)$, which indicates that y appears at the $r(y, O)$-th position in the order O. According to [1], the conditional expectation of the rank of an item y in an order O^* given O is

$$\mathrm{E}[r(y, O^*)|O] = r(y, O)\frac{|O^*| + 1}{|O| + 1}. \quad (1)$$

Because $|O^*|$ is constant for any observed order O, $\mathrm{E}[r(y, O^*)|O]$ is proportional to $r(y, O)/(|O| + 1)$. Based on this theorem, we convert a response order,

$$O_x = y_{l_1} \succ \cdots \succ y_{l_j} \succ \cdots \succ y_{l_{|O_x|}},$$

into a set of tuples,

$$\Big\{(x, y_{l_1}, 1/(|O_x| + 1)), \ldots, (x, y_{l_j}, j/(|O_x| + 1)), \ldots, (x, y_{l_{|O_x|}}, |O_x|/(|O_x| + 1))\Big\}.$$

Standard CF methods targeting scores can be applied to this converted set of scores. The only difference from standard scores is

that items with smaller converted scores indicates a stronger preference in items, whereas a larger value implies a stronger preference on a standard score scale. This conversion technique might seem rather brute-force, but it has worked well even in tasks other than CF, such as clustering [10] or object ranking [11].

We previously applied this conversion technique to a memory-based CF method developed for Grouplens [14] and its extensions [4]. We here introduce this technique to two model-based CF methods: pLSA [6] and matrix decomposition [12].

pLSA was originally proposed to derive a compact representation of words and documents, but it was applied to a CF task [6] and was used in a commercial system, GoogleNews [3]. The three-way model in [6] was slightly modified so as to deal with real value scores. Formally, z is a latent discrete random variable, whose domain is $\{1, \ldots, K\}$, and follows a categorical distribution, which is a K-way generalization of a Bernoulli distribution. Discrete random variables for a user and an item are denoted by x and y, respectively. Each of them conditionally follows a categorical distribution given z. In addition, a real random variable for scores, s, conditionally follows a normal distribution given z. Consequently, a log-likelihood function for a training set $\mathcal{D}$ is

$$\mathcal{L}(\mathcal{D}; \Theta) = \sum_{(x,y,s)\in\mathcal{D}} \log \sum_z \Pr[z] \Pr[x|z] \Pr[y|z] \mathcal{N}(s; \mu_z, \sigma_z^2).$$

Parameters maximizing this function can be easily derived by using an EM algorithm. Once model parameters are learned, a preference score for an item y by a user x can be inferred by calculating a conditional expectation,

$$\hat{s}_{xy} = \mathrm{E}[s|x, y] = \frac{\sum_z \mu_z \Pr[z] \Pr[x|z] \Pr[y|z] \mathcal{N}(s; 0, \sigma_z^2)}{\sum_z \Pr[z] \Pr[x|z] \Pr[y|z]}.$$

We further introduced a bias cancellation technique in [2], which was shown to alleviate the influences of various biases in scores collected by a scoring method. First, to alleviate a global effect, we computed b, which is the mean score over all scores in $\mathcal{D}$, from each score, and we get modified scores $s'_{xy} = s_{xy} - b$. From each modified score, the mean of all scores s' of an item y, d_y, is subtracted, and we get $s''_{xy} = s'_{xy} - d_y$. We further subtract the mean score over all scores s'' rated by a user x, c_x, and get $s'''_{xy} = s''_{xy} - c_x$. Scores in a training set $\mathcal{D}$ are replaced with these modified scores, and the above pLSA model is learned. After a modified score, $\hat{s}'''_{xy}$, is estimated, the score is corrected by adding biases, i.e., $\hat{s}_{xy} = \hat{s}'''_{xy} + b + c_x + d_y$.

The second model-based method is based on matrix decomposition (MD for short) in equation (4) in [12]. In this model, a preference score is predicted by

$$\hat{s}_{xy} = b + c_x + d_y + \mathbf{u}_y^\top \Big[\mathbf{v}_x + (\sum_{y'\in\mathcal{Y}_x} \mathbf{w}_{y'})/\sqrt{|\mathcal{Y}_x|}\Big], \quad (2)$$

where b, c_x, and d_y are parameters for canceling global, per-user, and per-item biases, respectively. $\mathbf{u}_y$, $\mathbf{v}_x$, and $\mathbf{w}_y$ are parameters with K-dimensional vectors. A dot product of $\mathbf{u}_y$ and $\mathbf{v}_x$ represents the cross effect between items and users, and the term $\mathbf{w}_y$ is intended to take into account the information about which items each user rates. These parameters are tuned so as to minimize the following loss function:

$$\mathrm{loss}(\mathcal{D}; \Theta) = \sum_{(x,y,s)\in\mathcal{D}} (s_{xy} - \hat{s}_{xy})^2 + \lambda R,$$

where R is the sum of L_2-regularization terms for all parameters except global bias, b, and λ is a regularization hyperparameter. The parameters are optimized so as to minimize this loss function. Once the parameters are learned, preference scores for any user and item pairs can be predicted by equation (2).

4. EXPERIMENTS

We next applied the model-based methods to our data set collected by using both ranking and scoring methods.

4.1 Datasets

To test the effectiveness of adopting a ranking method, we applied the methods in the previous section to our *sushi* data sets[2]. These data sets were collected by the same procedure as in [8], but the number of users was increased to 5000.

The preference data were collected through the following procedure. Before collecting the data, we surveyed menu data from 25 sushi restaurants found on the Web. For each type of sushi, we counted the number of restaurants that listed the sushi on their menu. From these counts, we derived the probability that each item would be supplied. Note that using this distribution violates the uniform assumption of equation (1), but even in such a case, the later experimental results show the effectiveness of our score conversion technique. By eliminating unfamiliar or low-frequency items, we compiled a list of 100 items.

We generated two item sets, which were presented as $\mathcal{Y}_x$ to each user. The type A set ($\mathcal{Y}^A$) was common for all users and composed of ten items: shrimp, sea eel, tuna, squid, sea urchin, salmon roe, egg, fatty tuna, tuna roll, and cucumber roll. This set was used for testing. The other type B sets ($\mathcal{Y}_x^B$) were different for each user. Ten items were randomly sampled according to the above probability distribution of items. The orders in this item set were treated as user responses. Note that $\mathcal{Y}^A$ and $\mathcal{Y}_x^B$ had an overlap of 2.58 items per order on average.

We collected the responses via a commercial Web survey service. The following queries were presented for each user x:

1) We asked the user to sort items in the $\mathcal{Y}^A$ set according to his/her preference and get a response order O_x^A.

2) We asked the user to rate the items in the $\mathcal{Y}_x^B$ set by a scoring method using a five-point scale. The set of response scores was denoted by S_x^B.

3, 4) Next two questions were irrespective to preferences. These two questions lessened the influence of query 2 on query 5.

5) We asked the user to sort the items in the $\mathcal{Y}_x^B$ set according to his/her preference and thus obtained a response order O_x^B.

6) The users were asked some demographic questions.

We screened users whose demographic features were rare or whose response times were either too short or too long, and the data set consequently included 5000 tuples: (O_x^A, O_x^B, S_x^B). We checked the ratio of responses that contained a contradiction between S_x^B and O_x^B. Here, a contradiction means that, although the item y_a precedes y_b in O_x^B, the score of y_b in S_x^B is rated higher than that of y_a, and vice versa. Only 31.7% of users rated all items without such a contradiction. This fact at least shows that different aspects of preference can be captured by ranking and scoring methods.

4.2 Experimental Results

Two model-based methods, pLSA and MD, were applied to the above sushi data set. Response orders, O_x^B, of all users were merged and converted, and we obtained a data set $\mathcal{D}_O$. Similarly, response scores, S_x^B, of all users were merged, forming another data set $\mathcal{D}_S$. Hyperparameters of model-based methods were tuned by minimiz-

[2]Sushi is a Japanese food. Data sets can be obtained from the site: `http://www.kamishima.net/sushi/`.

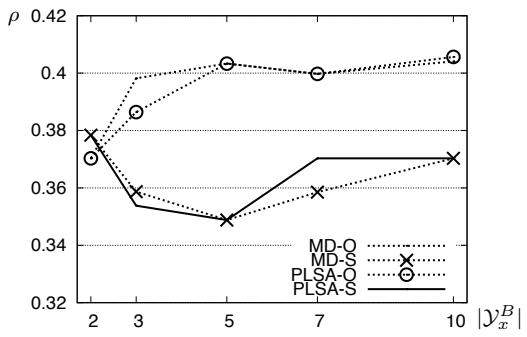


Figure 2: Changes in rank correlations between true and predicted orders

ing the squared errors derived by five-fold cross validation for each data set. We set the hyperparameters as follows:

method \ data	$\mathcal{D}_O$	$\mathcal{D}_S$
pLSA	K=2, $\alpha = 1.0$	K=3, $\alpha = 0.1$
MD	K=5, λ=0.01	K=5, λ=0.03

(α is a parameter of Dirichlet prior for $\Pr[x|z]$ and $\Pr[y|z]$)

Using these hyperparameters and and entire data set, $\mathcal{D}_O$ or $\mathcal{D}_S$, the models were trained. For each user x, the scores of all items in $\mathcal{Y}^A$ were predicted by each of these models. By sorting these items according to this user's predicted scores, we obtained a predicted preference order, $\hat{O}_x^A$. The concordance between the true order, O_x^A, and the predicted order, $\hat{O}_x^A$, was measured by Spearman's rank correlation, ρ, a widely used metric of the concordance between two orders. Note that MAE or squared error were widely used, but such absolute evaluation metrics are meaningless for data captured by a ranking method, as described in section 2.

We changed the number of rated items per user, i.e., $|\mathcal{Y}_x^B|$, by subsampling, and corresponding changes in Spearman's ρ are shown in Figure 2. MD or PLSA indicate the types of methods, and suffixes O and S indicate that these results were obtained from training sets, $\mathcal{D}_O$ and $\mathcal{D}_S$, respectively. If the number of items per user was three or more, adoption of a ranking method improved the accuracy of prediction. However, when $|\mathcal{Y}_x^B|$ was two, a scoring method is superior, as was also observed in our previous work [8]. In the case of a scoring method, two items are rated by using a five-point scale, and thus there are 10 possible choices in total. In the case of a ranking method, one can choose which of two items is preferred. Clearly, less information is provided from users with a ranking method. However, we can conclude that adopting a ranking method is generally fruitful for obtaining the recommendations performed by model-based methods.

5. CONCLUSION

We previously developed memory-based CF methods to deal with preference orders collected by using a ranking method. Here, the same technique is embedded into two model-based methods, pLSA and MD, which were applied to our sushi data set. Experimental results showed that it was effective for adopting a ranking method for collecting preference data.

Unfortunately, compared to memory-based methods, these model-based methods were inferior. We therefore plan to improve these model-based methods for orders.

6. ACKNOWLEDGMENTS

This work is supported by the grants-in-aid 14658106, 16700157, and 21500154 of the Japan society for the promotion of science.

7. REFERENCES

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

[2] R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Proc. of The 7th IEEE Int'l Conf. on Data Mining*, pages 43–52, 2007.

[3] A. Das, M. Datar, A. Garg, and S. Rajaram. Google News personalization: Scalable online collaborative filtering. In *Proc. of The 16th Int'l Conf. on World Wide Web*, pages 271–280, 2007.

[4] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proc. of The 22nd Annual ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 230–237, 1999.

[5] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. Recommending and evaluating choices in a virtual community of use. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 194–201, 1995.

[6] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *Proc. of the 16th Int'l Joint Conf. on Artificial Intelligence*, pages 688–693, 1999.

[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. 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.

[9] T. Kamishima and S. Akaho. Nantonac collaborative filtering — recommendation based on multiple order responses. In *Proc. of The Int'l Workshop on Data-Mining and Statistical Science*, pages 117–124, 2006.

[10] T. Kamishima and S. Akaho. Efficient clustering for orders. In D. A. Zighed, S. Tsumoto, Z. W. Ras, and H. Hacid, editors, *Mining Complex Data*, volume 165 of *Studies in Computational Intelligence*, chapter 15, pages 261–280. Springer, 2009.

[11] T. Kamishima, H. Kazawa, and S. Akaho. A survey and empirical comparison of object ranking methods. In J. Fürnkranz and E. Hüllermeier, editors, *Preference Learning*. Springer, 2010. [to appear].

[12] Y. Koren. Collaborative filtering with temporal dynamics. In *Proc. of The 15th Int'l Conf. on Knowledge Discovery and Data Mining*, pages 447–455, 2009.

[13] O. Luaces, G. F. Bayón, J. R. Quevedo, J. Díez, J. J. del Coz, and A. Bahamonde. Analyzing sensory data using non-linear preference learning with feature subset selection. In *Proc. of the 15th European Conf. on Machine Learning*, pages 286–297, 2004. [LNAI 3201].

[14] 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.