The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

Collaborative filtering based on a Weighted Maximum Margin Matrix Factorization

Masahiro Furuya, Shigeyuki Oba
Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara, 630-0192

Shin Ishii
Graduate School of Informatics
Kyoto University
Gokasho, Uji, Kyoto, 611-0011

Abstract

Collaborative filtering is a marketing method to recommend items which will be preferred by customers, where the degree of preferring each item by each user is predicted by a pre-stored preference data matrix of various users and items. A common approach to collaborative filtering is to approximate an observed target matrix and predict missing elements in the matrix. The maximum margin matrix factorization (MMMF) was recently suggested as an effective method for handling binary or ordinal-discrete valued target matrix (Srebro *et al.*, 2005). In this study, we proposed a weighted MMMF which extends the MMMF to deal with the cases where some matrix elements may have priorities to other elements. We divided a benchmark user-movie matrix into sparse and dense sub-matrix blocks, and put different weights to the blocks. By setting appropriate weights to the blocks, we could improve the mean prediction accuracy.

1 Introduction

From a standpoint of machine learning research, collaborative filtering is defined as an estimation problem of a true matrix X behind the observed matrix Y which includes noise and missing values.

A conventional approach to collaborative filtering is to fit a low rank factor model to the target matrix, so as to be used for further prediction [5][6][7]. A linear k-factor model is given by making a $n \times d$ target matrix Y be well represented by a low rank decomposition $X \equiv UV'$, where U and V are $n \times k$ and $d \times k$ matrices, respectively. To estimate the low-rank matrix X, a usual way is based on minimization of the sum-squared distance between the target matrix Y and the approximated matrix X. The assumption behind the linear low-rank model with k-factors is that there are actually k factors which dominate the user's rating behaviors. In general, however, the rank, i.e., the effective number of factors, is unknown, and then, we need to estimate it somehow.

Recently, Rennie and Srebro (2005) [1][2] proposed a promising approach, the maximum margin matrix factorization (MMMF) which is applicable for matrices with binary or ordinal ratings values. The MMMF employs the trace norm of X as a regularization term rather than setting a pre-determined rank to the approximated matrix X. In addition, a hinge loss function is used as the objective function, based on the idea of maximum margin, like in the support vector machines.

To introduce a weight to the objective function is one possibility to reduce prediction error, which leads to the weighted low-rank matrix factorization [4]. For example, better prediction can be achieved by putting smaller weights to elements with possibly a large noise and larger weights to reliable elements.

In this work, we propose a new approach for collaborative filtering, the weighted maximum margin matrix factorization (WMMMF). The WMMMF includes the above-mentioned two ideas: the MMMF and the weighted low rank approximation where we assume different weights between matrix parts with a small and large missing-value rates. Such an assumption is expected to be effective because items and users with different patterns of missing entries can have different levels of reliability. We apply our method to a benchmark movie-rating dataset and compare the prediction performance with those by some conventional methods, and show that our new method exhibits better than the MMMF with uniform weights.

2 Methods for collaborative filtering

In this section, we explain a method for collaborative filtering, the MMMF, and our extension, the WMMMF.

2.1 Matrix factorization

Given an observation matrix Y with noise and missing elements, our task is to predict the missing elements in Y. The basic strategy for this problem is to approximate the observation matrix $Y \in \mathbb{R}^{n \times d}$

by a lower rank matrix $X = UV' \in \mathbb{R}^{n\times d}$. where $U \in \mathbb{R}^{n\times k}$, $V \in \mathbb{R}^{d\times k}$, and $k < \min(n, d)$ is a rank of X. We call X the low rank matrix approximation of $Y = X + E$ if the error matrix $E \in \mathbb{R}^{n\times d}$ is small enough under some criterion.

It is trivial the error matrix is minimum when $Y = X$; this is a typical overfitting. In order to avoid such overfitting, the most popular way is to set a rank of the approximated matrix X to be appropriately small. Another way is to introduce a penalty term defined as the sum of squares of elements in U and V, i.e., $||U||^2_{Fro} + ||V||^2_{Fro}$ where $||\cdot||_{Fro}$ denotes the Frobenius norm of a matrix. This "conditional" problem does not consider any restriction of the rank of X.

2.2 MMMF for binary target

Srebro proposed the maximum margin matrix factorization (MMMF) for collaborative filtering, which uses the trace-norm for a regularization term and a hinge-loss function for the error criterion [2].

In the MMMF for a binary target matrix, $Y \in \{\pm 1\}^{n\times m}$, we seek a real-valued matrix X which approximates Y with the smallest error. How good is the approximation X is evaluated by the objective function:

$$J(X) = C||X||_\Sigma + \sum_{ij\in S} h(Y_{ij}X_{ij}), \tag{1}$$

where $h(z) = \max(0, 1 - z)$ is the hinge loss, C is a trade-off constant, and summation is taken for all observed entries in Y.

The trace norm $||X||_\Sigma$ is the sum of the singular values of X, and is given by the Frobenius norm of the factored matrix as follows.

Lemma 1. If the matrix X is factorized as $X = UV'$, the trace norm $||X||_\Sigma$ is given by

$$\begin{aligned} ||X||_\Sigma &= \min_{X=UV'} ||U||_{Fro}||V||_{Fro} \\ &= \min_{X=UV'} \frac{1}{2}(||U||^2_{Fro} + ||V||^2_{Fro}). \end{aligned}$$

By using the character of the trace norm in Lemma 1, Srebro *et al.* [2] also obtained a factored version of the minimization of the objective function

$$J(U, V) = \frac{C}{2}(||U||^2_{Fro} + ||V||^2_{Fro}) + \sum_{ij\in S} h(Y_{ij}X_{ij}), \tag{2}$$

where min $J(U, V)$ is equivalent to min $J(X)$.

2.3 Weighted maximum margin matrix factorization for binary target

We introduce a weight parameter into the MMMF and derive the WMMMF. This is a combination of the ideas of MMMF [2] and weighted low-rank approximation [4]. In the WMMMF, a weighted loss is used as the objective function

$$J(U, V) = \frac{C}{2}(||U||^2_{Fro} + ||V||^2_{Fro}) + \sum_{ij\in S} W_{ij}h(Y_{ij}X_{ij}). \tag{3}$$

W_{ij} is a weight value corresponding to the ij element of the target matrix, Y_{ij}. The weight represents the importance of elements of the target matrix for prediction, so that, for example an unreliable element with a large noise has a small weight and hence a low contribution to the prediction.

2.4 Weighted MMMF for rating target

To consider ordinal rating data, $Y_{ij} \in \{1, 2, ..., R\}$, in collaborative filtering, we use an ordinal hinge loss in place of the hinge loss used in [1]. Then, the objective function of WMMMF for ordinal rating data is given by

$$\begin{aligned} J(U, V, \theta) &= \frac{C}{2}(||U||^2_{Fro} + ||V||^2_{Fro}) \\ &+ \sum_{r=1}^{R-1}\sum_{ij\in S} W_{ij}h(T^r_{ij}(\theta_{ir} - X_{ij})), \end{aligned} \tag{4}$$

where T^r_{ij} is such a binary index that $T^r_{ij} = +1$ for $r \geq Y_{ij}$ and $T^r_{ij} = -1$ for $r < Y_{ij}$, and $\theta_{i1}, ..., \theta_{ir}, ..., \theta_{i(R-1)}$ are $R - 1$ thresholds for the ith user; the thresholds define the relationship between a real valued X_{ij} and ordinal rating value of Y_{ij}, so that $\theta_{i(Y_{ij}-1)}+1 \leq X_{ij} \leq \theta_{i(Y_{ij})} - 1$ holds for each element. In the MMMF and WMMMF, the thresholds θ_{ir} should be determined from data, and different setting of thresholds dependent on users may cause better prediction. Therefore, we try to minimize the objective function with respective to the matrix $X = UV'$ and the thresholds θ.

We use a gradient descent method for locally minimizing $J(U, V, \theta)$ based on the partial derivative with respect to each element of U, V, and θ :

$$\frac{\partial J}{\partial U_{ia}} = CU_{ia} - \sum_{r=1}^{R-1}\sum_{j|ij\in S} w_{ij}T^r_{ij}h'\big(T^r_{ij}(\theta_{ir} - U_iV'_j)\big)V_{ja}, \tag{5}$$

$$\frac{\partial J}{\partial V_{ja}} = CV_{ja} - \sum_{r=1}^{R-1}\sum_{i|ij\in S} w_{ij}T^r_{ij}h'\big(T^r_{ij}(\theta_{ir} - U_iV'_j)\big)U_{ia}, \tag{6}$$

$$\frac{\partial J}{\partial \theta_{ir}} = \sum_{j|ij\in S} w_{ij}T^r_{ij}h'\big(T^r_{ij}(\theta_{ir} - U_iV'_j)\big). \tag{7}$$

In order to obtain stable derivative, we used a smoothed hinge loss function in place of the naive hinge loss function in the following experiments.

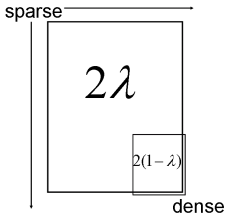

Figure 1: The relationship between the sub-matrices, the dense block and the sparse blocks, and their weight values

3 Experiments

We examined our methods by a subset of the EachMovie dataset [3]. It contains a matrix of 5000 users for 1623 movies, each of whose elements is a rating value $\{1, 2, \ldots, 6\}$. 1000 users were picked up out of 5000 from the original data into a 1000×1623 matrix among which there are 63592 observed entries. We divided the observed elements in the data matrix into 70% for training and 30% for test; we randomly prepared 10 different sets of training and test elements.

In the preprocessing phase, we divided users into sparse and dense users according to the numbers of missing values in their vectors, and also divided movies into sparse and dense ones similarly. We called the sub-matrix of dense users and dense movies a dense block and the others sparse blocks. The dense block contained about 8000 observed entries and missing rate in this block was about 60%, and the sparse blocks contained 36000 observed entries and missing rate in these blocks was about 97%. We put different weights $2(1-\lambda)$ and 2λ to the dense and sparse blocks, respectively (See Figure 1).

In the training phase, we applied the WMMMF with various values of the weight λ and the trade-off constant C. For each missing entry, a discrete rating value was estimated by applying discretization.

In evaluation, we used mean absolute error (MAE) as the prediction accuracy. We also examined how the performance behaves for various values of the weight λ and the trade-off constant C (Figure 2). We found that the best result was achieved at $\lambda = 0.65$ and $C = 10^{11\ 8} = 23.71$. If appropriate values for the weight and the trade-off constant are given, our method obtained a better result than the MMMF which is equivalent to our WMMMF with a special setting of $\lambda = 0.5$.

The best prediction error for each method for each training/test division is shown in table 1. The prediction errors decreased by introducing weights in all cases. It should be noted that the difference in the

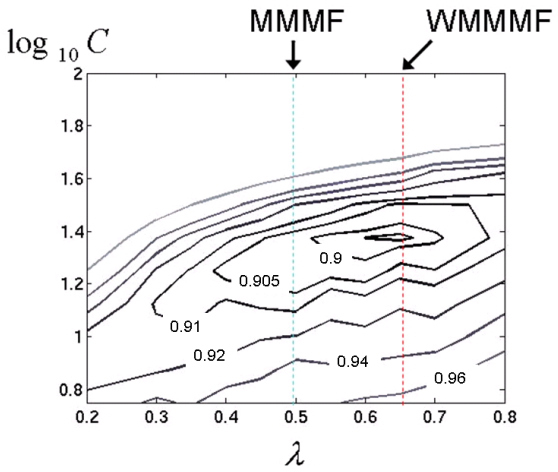

Figure 2: A contour plot of MAE to show the dependence of the weight λ and the trade-off constant C

average error between by the WMMMF and MMMF was comparable to the standard deviation, implying the improvement was not so substantial. And, the estimated accuracies should include overestimation because the weight parameter was optimized with respect to the test accuracy, which leads to incomplete validation. However, the optimized values of the weight λ was consistently larger than 0.5, suggesting that putting a larger weight to the sparse blocks improved the performance.

Next, we applied the WMMMF to another situation where we perform prediction of only a part of matrix of particular users and particular items. In a recommendation system, there can be some cases to seek users who will prefer particular items, or to seek items which will be preferred by particular users. In such a situation, the requested users and items should be highly weighted for better prediction. Here, we regarded the dense part as to be estimated and evaluated prediction errors only in the dense part. We compared three ways of estimation; MMMF1 applies MMMF only to the dense part, MMMF2 applies MMMF with uniform weights, and WMMMF provides a best value of weight to the dense part (Table 2). We found that the prediction error was decreased by the WMMMF for every training/test division, and then concluded the WMMMF is a good method in such a situation.

4 Discussion and Conclusions

In this study, we proposed a new method for collaborative filtering. The experiments showed that our

Table 1: Prediction errors of user-movie rating data. MAE performed by MMMF and WMMMF, and the best weight value λ in WMMMF are shown for 10 trials of training/test divisions.

Dataset	1	2	3	4	5	6	7	8	9	10	mean	std.
MMMF	0.9012	0.9059	0.9121	0.9102	0.9084	0.9157	0.9136	0.9102	0.9135	0.9122	0.9103	0.0043
WMMMF	0.8973	0.9045	0.9078	0.908	0.902	0.9118	0.9075	0.9053	0.9127	0.9086	0.9065	0.0045
λ	0.65	0.55	0.55	0.6	0.65	0.55	0.6	0.6	0.65	0.6		

Table 2: Prediction errors only on the dense part of user-movie rating data. Three different ways, MMMF1, MMMF2, and WMMMF are compared. MAEs obtained by the three ways and the best weight values λ are shown for 10 datasets.

Dataset	1	2	3	4	5	6	7	8	9	10	mean	std.
MMMF1	0.8602	0.8701	0.8742	0.8726	0.8704	0.8789	0.8673	0.8696	0.8731	0.8877	0.8724	0.0072
MMMF2	0.8364	0.859	0.8638	0.8505	0.8673	0.8442	0.8316	0.8385	0.8451	0.8382	0.8474	0.0123
WMMMF	0.8352	0.8501	0.8537	0.849	0.8527	0.8408	0.8297	0.8443	0.8375	0.8321	0.8425	0.0087
λ	0.35	0.4	0.3	0.45	0.3	0.45	0.4	0.45	0.45	0.45		

WMMMF is effective. In our method, however, the weight matrix W has been added to the variables of the original MMMF, U, V, θ, so this method needs a more computational cost than the MMMF. It is necessary to devise a method for obtaining good values of the weight and the trade-off constant in an efficient manner. In addition, since the objective function $J(U, V, \theta)$ is not convex, the optimization process sometimes falls into local minima as can be seen in the original MMMF. To overcome these problems is our future study.

Acknowledgements

This work was supported by a Grant-in-Aid for Young Scientists 19710172 from the MEXT.

References

[1] Rennie, J.D.M. and Srebro, N., "Fast maximum margin matrix factorization for collaborative prediction," *Proceedings of the 22nd International Conference on Machine Learning (ICML)*,pp.713–719, 2005.

[2] Srebro, N. and Rennie, J.D.M. and Jaakkola, T., "Maximum-margin matrix factorization," *dvances in Neural Information Processing Systems*, 2005.

[3] Yu, K. and Chu, W. and Yu, S. and Tresp, V. and Xu, Z., "Stochastic Relational Models for Discriminative Link Prediction," *Advances in Neural Information Processing Systems*, pp. 1553–1560, 2006.

[4] Srebro, N. and Jaakkola, T., "Weighted low-rank approximations," *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 720–727, 2003.

[5] Billsus, D. and Pazzani, M.J., "Learning collaborative information filters," *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 46–54, 1998.

[6] Hofmann, T., "Latent semantic models for collaborative filtering," *ACM Transactions on Information Systems (TOIS)*, pp. 89–115, 2004.

[7] Marlin, B. and Zemel, R.S., "The multiple multiplicative factor model for collaborative filtering," *ACM International Conference Proceeding Series*, pp. 576-583, 2004.

[8] Rennie, J.D.M. and Srebro, N., "Loss Functions for Preference Levels: Regression with Discrete Ordered Labels" *Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling*, 2005.

[9] Canny, J., "GaP: a factor model for discrete data," *Proceedings of the 27th annual international conference on Research and development in information retrieval*, pp. 122–129, 2004.