©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.11.IMAWM-472

One-Class Maximum Margin Matrix Factorization

Bin Shen$^{\star 1}$***, Cheng Lu*******, Qifan Wang***$^{\star 1}$***, Si Chen***†***, Yan Yan***‡
****Purdue University, West Lafayette, IN, U.S.A***
†***School of Computer and Information Engineering, Xiamen University of Technology, Fujian, P. R. China***
‡ ***Department of Computer Science, Xiamen University, Fujian, P. R. China***

Abstract

Matrix factorization has been a key technique in learning latent factor models for many applications in computer vision and pattern recognition such as image annotation and collaborative prediction. Specifically, in collaborative filtering problems, the goal of matrix factorization is to predict the missing values based on the low-rank factorization gained based on observed entries. Among various algorithms, maximum margin matrix factorization has been a successful approach to discriminative collaborative filtering problems, where the input matrix is binary.

In this paper, we consider the problem of one-class discriminative collaborative filtering, where the data matrix is binary and only positive values can be observed, i.e. the entries of data matrix can be either observed as positive or missing. Many real applications fall in this category. For example, given an image with incomplete tag list: cat, tree, garden*, we are only sure the image has* cat *while not sure whether it has* grass *or not since the tag list is incomplete.*

To cope with this problem, one-class Maximum Margin Matrix Factorization (one-class MMMF), which inherits the merits of both the applicability of one-class SVM and the discriminative power of maximum margin matrix factorization, is proposed. Extensive experiments conducted on both simulated toy data and real benchmark image datasets demonstrate that the proposed approach is considerably superior to the traditional approaches, which simply assume unobserved entries as negative.

Introduction

Matrix factorization has been a key technique in learning latent factor models for many applications in computer vision and pattern recognition such as image annotation [9], collaborative prediction [6] and clustering [3]. Typical matrix factorization techniques seek to approximate a given data matrix by the product of two low-rank matrices such that the difference between the given matrix and its factorized form is minimized according to some certain optimality criterion depending on specific applications [7, 8, 19, 14]. In many real world applications, especially collaborative filtering problems such as tag completion [22], the data matrix is only partially observed. Accordingly, a natural task is to predict the missing values, for which matrix factorization is one of the most popular and successful approaches. For example, it achieves state-of-the-art performance on the large-scale Netflix dataset [1], which has more than one hundred million non-zero entries. Since matrix factorization does not require extra features and relies on only the non-zero entries to make prediction for the missing entries, it can be easily applied to different domains without careful engineering work of domain-specific features.

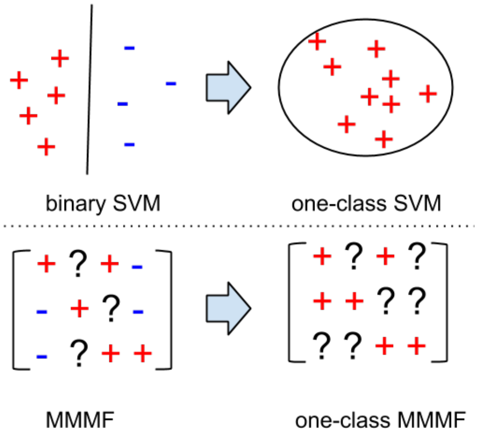


Figure 1: SVM and one-class SVM versus MMMF and one-class MMMF

In typical setting of matrix factorization with missing entries, the data matrix is assumed to be low-rank and all entries can be of arbitrary real values. Factorization is conducted by minimizing the loss function defined on the observed entries together with proper regularizer. In general, the observed entries of data matrix can be of arbitrary real values. However, for certain specific applications, the observed entries may be subject to various restrictions. For example, Nonnegative Matrix Factorization (NMF) [7, 8] only allows the data matrix to be nonnegative. Maximum Margin Matrix Factorization [19] considers the matrix composed of entries with binary values, which is either -1 or $+1$.

One-Class Discriminative Collaborative Filtering In this paper, we consider a special setting of collaborative filtering, where only positive responses can be observed. In other words, the entries of data matrix can be either missing or observed as positive. For presentational convenience, here we consider the case that the data matrix is of binary values $(+1 or-1)$ and only $+1$ can be observed. However, it can be easily extended to cases where positive observed values are discrete or even real. An example observation matrix can be

$$\begin{pmatrix} +1 & ? & +1 & ? \\ +1 & +1 & ? & ? \\ ? & ? & +1 & +1 \end{pmatrix}.$$

Many real problems fall into this category. For example, in the setting of image annotation problem, the complete *image-tag*

[1] These authors are now with Google. This work was done when these authors were with Purdue University, West Lafayette.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.11.IMAWM-472

matrix X of size $m \times n$ is binary matrix, each column of which corresponds to an image and each row corresponds to a possible tag. In theory, the $X_{ij} = +1$ if the i_{th} image contains an instance of tag j, and $X_{ij} = -1$ if the image does not have an instance. However, in computer vision problems, machine learning algorithms usually rely on the training data which contains a set of images with tags, which are usually incomplete. Once an image is annotated with a label, say *dog*, the machine learning algorithm will treat this image as a positive sample for the class *dog*. The image is not annotated with label *tree*, however, it should not be simply treated as a negative sample of class *tree*, since it can be a missing label. Another example is the user preference prediction for videos. If a user finishes watching a video, it is probably true that the user likes the video. However, it does not mean the user does not like it if he does not. An alternative explanation is that he does not get exposure to that video at all. Not being observed is not the same as being negative.

We name the problem of recovering the missing entries in the binary matrix based on only positive observations in the matrix as One-Class Discriminative Collaborative Filtering (OCDCF), since traditional collaborative filtering techniques are not able to cope with this setting.

When the data matrix is with missing values, many algorithms simply treat the the missing values as negative [21, 10], though the missing values do not necessarily mean they are negative values, and thus introduce systematic errors into the future processing.

To cope with OCDCF, instead of assuming the unobserved entries as *negative*, we propose a one-class Maximum Margin Matrix Factorization (one-class MMMF) algorithm to predict the missing entries, i.e., to predict the missing ones are either truly negative or unobserved positive.

A piece of work close to our approach is one-class Support Vector Machine (one-class SVM) [17]. Unlike conventional support vector machine (SVM) [20], which considers training samples from two distinct classes, one-class SVM considers the situation that the training samples are from one class and the prediction task is to decide whether the testing samples are of the same class or not. Maximum Margin Matrix Factorization (MMMF) [19] introduces hinge loss into matrix factorization to incorporates it with discriminative power. The analogy between SVM and one-class SVM is very similar to the one between MMMF and our proposed algorithm, named one-class Maximum Margin Matrix Factorization, as shown in Figure 1.

This paper continues as follows: the related works, including one-class SVM and MMMF, are reviewed in section . Proposed one-class MMMF together with related optimization method is detailed in section . Experimental results on both synthetic and real benchmark datasets are presented in section **??**, followed by conclusion and discussion at the end.

Related Work

There are many work focusing on classifying samples of different categories. However, only a few handle multiple-class problem while only having samples from a single class [11, 13, 2, 15]. Among them, the classic one-class SVM is the most basic and relevant methods to ours.

Also, there has been substantial previous work concentrating on collaborative filtering, most of them are based on minimizing the bregman divergence of given data matrix and its factorization, although these loss functions are not suitable for the setting of classification like discriminative collaborative filtering. Maximum matrix factorization introduces hinge loss into factorization and thus is suitable for handling discriminative collaborative filtering problems.

One-Class SVM

We briefly review the formulation of Support Vector Machine(SVM) and one-class Support Vector Machine (one-class SVM) in this section, .

Given a set of samples with labels $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in R^m$ is the feature vector of i_{th} sample and y_i is the corresponding label (in the setting of binary classification, $y_i \in \{-1, +1\}$), conventional SVM learns a decision hyperplane $w \in R^m$ by maximizing the margin between samples of different classes, and the related optimization problem can be formulated as:

$$\min_{w} C \sum_{i=1}^{n} \max(0, 1 - y_i w^T x_i) + \frac{1}{2}\|w\|^2. \tag{1}$$

The first term in (1) is the hinge loss, the second term is a standard regularizer for w, and C is a parameter controlling the tradeoff between the hinge loss and the regularization on w. The bias term is omitted here to simplify the problem while keeping it entirely general since it could be easily introduced as an additional coordinate in the data.

Unlike SVM, which have samples from different classes in training data, one-class SVM considers situations where only samples belonging to a single class are available. Given n samples of a same class, one-class SVM aims to find a compact ball such that all samples lie in it. Combining the hinge loss and standard regularizer, the problem can be formulated as:

$$\begin{aligned} \min_{w,\rho} \frac{1}{2}\|w\|_2^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \xi_i - \rho \\ s.t.\ \ \langle w, x_i \rangle \geq \rho - \xi_i \\ \xi_i \geq 0\ \ \forall i \end{aligned} \tag{2}$$

where $\nu \in (0, 1]$ is the tradeoff parameter. For both SVM and one-class SVM, kernel trick can be easily introduced, and thus the model is not necessary to be linear.

Maximum Margin Matrix Factorization

Maximum Margin Matrix Factorization (MMMF) copes with discriminative collaborative filtering problems without looking at domain specific features. Given a binary matrix X of size $m \times n$ with missing values, MMMF aims to find low-rank approximation of X, denoted as $\hat{X}$ by minimizing a linear combination the trace norm of $\hat{X}$, which serves as a regularizer to minimize the rank approximately,and its hinge loss relative to X:

$$\min_{\hat{X} \in R^{m \times n}} \|\hat{X}\|_{\Sigma} + C \sum_{(i,j) \in S} \max(0, 1 - X_{ij}\hat{X}_{ij}), \tag{3}$$

where $\|\cdot\|_{\Sigma}$ denotes the trace norm and S is set of indices of entries being observed.

DOI: 10.2352/ISSN.2470-1173.2016.11.IMAWM-472

Semi-Definite Programming (SDP) is introduced to optimize the objective function above, however, it is very difficult for SDP to scale up. The optimization problem can be simplified if the rank of matrix $\hat{X}$ is fixed to some small p. Then the optimization problem of fixed low-rank MMMF can be reformulated as:

$$\begin{aligned}\min_{W\in R^{m\times p}, H\in R^{n\times p}} \sum_{(i,j)\in S} & \max(0, 1-X_{ij}\langle W_i, H_j\rangle) \\ & +\alpha\|W\|_F^2+\beta\|H\|_F^2,\end{aligned} \tag{4}$$

where W_i is the i_{th} row of W and H_j is the j_{th} row of H and α and β are parameters controlling the strength of standard regularizers on W and H.

Once W and H for learned, to predict X_{ij}, $\forall(i,j)\notin S$, we can easily rely on the corresponding W_i and H_j.

One-Class Maximum Margin Matrix Factorization

In this section, we describe the proposed one-class Maximum Margin Matrix Factorization (one-class MMMF) with the formulation, optimization and discussion. This proposed algorithm is designed to cope with the one-class discriminative collaborative filtering (OCDCF) problem, where only positive responses can be observed.

Formulation

Given a matrix $X\in R^{m\times n}$ with missing values, let S denote the set of observed entries in this matrix and all of these observed entries are positive responses, i.e., $X_{ij}=1, \forall\,(i,j)\in S$. Note when $(i,j)\notin S$, X_{ij} is not observed and it is not assumed to be negative. Given these positive entries in S, the goal is to predict if X_{ij} should be positive or not $\forall\,(i,j)\notin S$.

Similarly to other approaches, two low-rank matrices $W\in R^{m\times p}$ and $H\in R^{n\times p}$ are sought to approximate the given matrix X in one-class MMMF. With observed entries being of same value, traditional loss functions, such as squared errors, are not applicable or meaningful. Also, without proper regularizer, it is easy to get stuck on trivial solutions, for example, $W=\{\frac{1}{p}\}^{m\times p}$ and $H=\{1\}^{n\times p}$ is a combination that perfectly fits all observed entries since WH results in a matrix that has all entries equaling to 1. However, this trivial solution obviously overfits the observed values and does not generalize, and thus cannot be adopted to predict the missing values.

Instead, we follow the same manner of one-class SVM, and introduce hinge loss with proper regularizer. In one-class SVM, one compact ball is sought to cover all observed positive samples. The ball is said to be compact means that its radius is minimized at the same time when the hinge loss is minimized. Take *image-tag* matrix as $X\in R^{m\times n}$ in one-class MMMF for example. In this case, rows of W can be viewed as latent representation of images and rows of H can be treated as latent representation of labels. In one-class MMMF, for each image a compact ball is sought to cover all the observed related labels, and for each label a compact ball is sought to cover all the observed related images. Considering these two parts above, the one-class MMMF objective function is designed as:

$$\begin{aligned}& \frac{1}{2}\|W\|_F^2+\frac{1}{2}\|H\|_F^2+C\sum_{(i,j)\in\ S}\xi_{ij} \\ s.t\ \ & \kappa(W_i,H_j)\geq 1-\xi_{ij}\ \ \forall\,(i,j)\in\ S \\ & \xi_{ij}\geq 0\ \ \forall\,(i,j)\in\ S\end{aligned} \tag{5}$$

where $S=\{(i,j)|X_{ij}=1\}$ and C is the parameter controlling the tradeoff between hinge loss and regularizer. $\kappa(.,.)$ is a kernel function. here we adopt the linear kernel, which means $\kappa(x,y)=\langle x,y\rangle$.

Optimization for One-Class MMMF

Here we present the optimization of equation 5 with respect to $W\in R^{m\times p}$ and $H\in R^{n\times p}$. Optimizing with respect to W and H jointly is difficult due to nonconvexity. However, optimization with respect to W or H only is convex and easy to solve. Thus, we refer to alternative optimization approach by repeating the following two steps until convergence.

Step 1: Fix H, optimize W. To optimize equation 5 with respect to W, we have:

$$\begin{aligned}& \arg\min_{W\in R^{m\times k}}\frac{1}{2}\|W\|_F^2+\frac{1}{2}\|H\|_F^2+C\sum_{(i,j)\in\ S}\xi_{ij} \\ & s.t\ \ \kappa(W_i,H_j)\geq 1-\xi_{ij}\ \ \forall\,(i,j)\in\ S \\ & \xi_{ij}\geq 0\ \ \forall\,(i,j)\in\ S\end{aligned} \tag{6}$$

Moreover, the optimization problem above can be further decomposed into a set of independent subproblems, each of which optimizes with respect to a row of W. Without loss of generality, we consider the optimization problem with respect to i_{th} row of W, i.e. W_i.

$$\begin{aligned}& \arg\min_{W_i\in R^k}\frac{1}{2}\|W\|_F^2+\frac{1}{2}\|H\|_F^2+C\sum_{i,j}\xi_{ij} \\ =& \arg\min_{W_i\in R^k}\frac{1}{2}\|W\|_F^2+C\sum_{(i,j)\in S}\xi_{ij} \\ =& \arg\min_{W_i\in R^k}\frac{1}{2}\|W_i\|_2^2+C\sum_{(i,j)\in S}\xi_{ij} \\ & s.t\ \ \kappa(W_i,H_j)\geq 1-\xi_{ij}\ \ \forall\,(i,j)\in\ S \\ & \xi_{ij}\geq 0\ \ \forall\,(i,j)\in\ S\end{aligned} \tag{7}$$

The problem above is a standard convex programming problem. Actually, since the i is fixed, it reduces to a standard one-class SVM optimization problem. When we are optimizing with respect to W_i, the rows of H are considered as samples to be classified and the i_{th} row of matrix X contains the corresponding labels, which are either missing or positive. This can be efficiently solved by standard optimization toolboxes. Also, many of the techniques [16, 18] that help to speed up the conventional SVM could be easily introduced without much modification.

Step 2: Fix W, optimize H. Similarly, the optimization with respect to rows of H is in the same manner:

$$\begin{aligned}& \min_{H_i\in R^k} C\sum_j\xi_{ji}+\frac{1}{2}\|H_i\|_F^2 \\ & s.t\ \ \kappa(W_j,H_i)\geq 1-\xi_{ji}\ \ for\ (j,i)\in\ S\end{aligned} \tag{8}$$

Algorithm 1 One-Class Maximum Margin Matrix Factorization

```
Require: $X \in R^{m \times n}$ with $S$, the set of observed entries; $p$, the
    dimension of latent space.
Initialize $W \in R^{m \times p}$, $H \in R^{n \times p}$
for $t = 1, ..., max\_iter$ do
    for $i = 1, ..., m$ do
        Update $W_{i.}$ by minimizing equation 7
    end for
    for $i = 1, ..., n$ do
        Update $H_{i.}$ by minimizing equation 8
    end for
end for
return W, H
```

The overall algorithm of alternative optimization is summarized in Algorithm 1.

Discussion

Overall, we fix H and update all rows of W, and then fix W and update rows of H. In each step, an optimization problem, same as standard one-class SVM, is solved. Since different rows of W or H can be updated independently given fixed H or W, respectively, the optimization methods can be easily run in parallel to speed up.

The algorithm is guaranteed to converge, since there is obviously a lower bound for the objective function, for example 0. Also, each of the updating steps will decrease the objective function.

Experimental Results

In order to evaluate our proposed one-class MMMF method, we first test it on a **synthetic dataset**. Then one-class MMMF is further evaluated on two types of experiments in real applications: image tag completion and street view text word recovery. For image tag completion, we use public dataset: **MIRFLICKER-25K** [5]. For street view text letter recovery, we use **ICDAR2003** [12], which is a competition dataset for text robust reading.

For the **synthetic dataset**, we first generate two base matrices W and H, in which the elements are uniformly distributed in $[0,1]$. Then we threshold matrix $W \times H$ to generate original matrix X so that partial elements in X are labeled as 1. Specifically, X_{ij} is set to 1 if $(WH)_{ij}$ is greater than the predefined threshold. In our implementation, the threshold is tuned so that about 25% of elements of X are labeled as 1.

The database **MIRFLICKER-25K** includes 25000 images with 38 different tags (*e.g.* baby, dog, flower, tree, transport, river, flag, *etc.*). We use all 38 tags in our experiments and generate matrix representation for this database. Every entry of this matrix is either unknown or 1 while $X_{ij} = 1$ means that the i_{th} image has tag j, while $X_{ij} = 1$ means it is unknown. However, many images in this dataset have a small number of tags. Thus they provide little information about statistical correlations among different tags. So we apply preprocessing to this dataset to exclude those images that have fewer than 12 tags. Because of this preprocessing, the matrix $X = (x_{ij})_{m \times n}$ to be factorized has a size of 104×38.

On **ICDAR2003** database, it includes 1156 street view images with text in each of them. We extract text in each image from its tag file to generate original matrix X to be factorized. Beacause we only focus on recovering semantic information in each image based on matrix factorization, we do not distinguish lower-case and upper-case letter in our application. In the original matrix X, each row represents an image while each column represents an english letter. For example, $X_{32} = 1$ means that in the 3rd image there is *at least one* letter 'b'. So the original matrix to be factorized in this case has a size of 1156×26. The dimensionality of matrix representation (after preprocessing) for these three datasets is give in Table 1.

Table 1: Dimensionality of the three datasets

Dataset	m	n	p
Synthetic	100	100	40
MIRFLICKR-25K	91	37	20
ICDAR2003	108	26	13

Baseline Algorithms

There are many related works that handle collaborative filtering. Based on the popularity, performance and closeness to our proposed algorithm, The proposed one-class MMMF algorithm is evaluated against classical method: Weighted Nonnegative Matrix Factorization (WNMF)[4]. WNMF method excludes those entries which are masked in the original matrix from the cost function. By masking, we can easily divide the original matrix into two parts: training and testing.

Implementation Details and Evaluation

Now we characterize parameters and performance measures for these methods. For all datasets mentioned above with original matrix $X \in R^{m \times n}$, we choose $W \in R^{m \times p}$ and $H \in R^{n \times p}$ where $p \approx \frac{n}{2}$. The sizes of W and H for different datasets are shown in Table 1.

Then we randomly mask 20% of elements in the original matrix X and take them as testing set. The rest 80% of elements are retained for training which is denoted as X^t. In order to show that our algorithm can better handle matrix with unknown entries, we further randomly mask σ percent of elements in X^t as unobserved. For WNMF, the unobserved entries are treated as negative, as many traditional algorithms do, thus each of the corresponding enties is replaced by 0. Finally, our input matrix to be factorized is denoted as X^*. For WNMF, we can achieve 20% masking by specifying the weight matrix M in the objective cost function:

$$O_{wnmf} = ||M \odot (X - WH)||_2^F, \tag{9}$$

where $m_{ij} = 0$ if this element is masked for testing, while $m_{ij} = 1$ if it is in the training set. The goal is to recover original X based on X^*, where X^* contains only 80% of original elements and σ percent of its elements are unobserved.

For one-class MMMF, we initialize two matrices $W \in R^{m \times p}$ and $H \in R^{n \times p}$ as $W_{ij} \sim U[0,1]$, $H_{ij} \sim U[0,1]$. On all the datasets, our experiments show that 30 iterations are sufficient for one-class MMMF to reach convergence. Let W^1 and H^1 denote the initial matrices. Assume in the iteration k, we sequentially optimize each row of W^k followed by each row of H^k. In order to update each row W_i^k, all the rows of H^k which correspond to non-zero entries in X are viewed as samples for optimization. Symmetrically, to update each row H_j^k, all the rows of W^k which correspond to non-zero entries in X are viewed as samples for optimization. We use *BFGS* Quasi-Newton approach in every iteration to update from W^k, H^k to W^{k+1}, H^{k+1}. Experiments show that BFGS has the faster convergence speed when compared to other gradient methods like *DFP* and *Conjugate Gradient*. Once the iteration reaches convergence, we get recovered matrix $\hat{X}$ as $\hat{X} = W \times H$. And an optimal threshold value T^* is searched to binarize $\hat{X}$.

For one-class MMMF, different values for box constraint C and threshold T will generate different estimates $\hat{X}$. We perform an exhaustive search on C and T to find the best combination (C^*, T^*) that can maximize the F_1 score in the testing set. For WNMF, we only need to search for the optimal T^* that leads to the maximum F_1 score in the testing set. Because of the randomness in initialization and optimization, we repeat the whole factorization process three times for both methods, at every level of σ, to obtain three maximum F_1 scores. For every σ, we calculate its corresponding mean maximum F_1 score mF_1^{max} for all three methods. The testing results on three datasets with different unknown portions $\sigma = 0.2, 0.25, 0.30, 0.35, 0.40$ are given in Table 2. As a visualization of Table. 2, we also present our testing results in Fig. 2. Note, in the figure and table reporting results, OCMMMF is short for one-class MMMF for notational convenience.

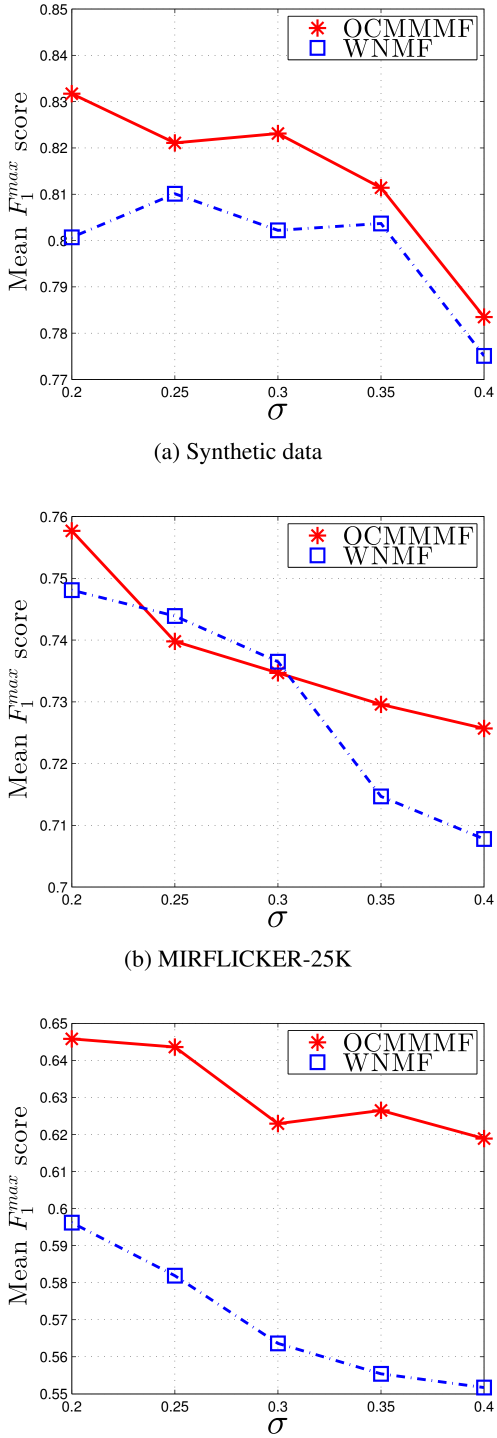
Through the results in Table 2, we can conclude that one-class MMMF achieves higher mF_{pn} scores in most cases when compared to WNMF on all 3 datasets. It consistently outperforms the WNMF method. Especially on the **ICDAR2003** dataset for word completion task, it significantly outperforms the WNMF by about 8%.

Discussion

The experiments show one-class MMMF is favored against the baseline algorithm in term of the mF_1^{max} score. However, one-class MMMF is much more computationally extensive compared to WNMF. Even though we do not need to deal with joint optimization in one-class MMMF to get gradients of its overall cost functions. It still require extensive convex optimization in every iteration when updating W^k and H^k, in which we apply BFGS for every row in W or column in H until it converge based on the 'samples' in other matrix. So there is probably some space to reduce the computational complexity.

Conclusion

In real world applications, data are missing in many information resources. However, not being observed is not equal to nonexistence. As one of the most popular discriminative collaborative filtering methods, Maximum Margin Matrix Factorization (MMMF) introduces the typical classification loss - hinge loss into matrix factorization. With the observation that in some real situations negative values cannot be observed and entries of data matrix can only be either positive or missing, to cope with the one-class discriminative collaborative filtering problems, we extend the MMMF to one-class MMMF, which does not require negative entries and is able to predict the missing entries. Experiments on both synthetic data and real datasets show the efficacy of the proposed algorithm in predicting the missing entries of data matrix with observation restricted to positive reponses.



(a) Synthetic data

(b) MIRFLICKER-25K

(c) ICDAR2003

Figure 2: Testing results on three datasets with different testing portions

For future research, how to incorporate side information into this setting to combine one-class discriminative collaborative filtering and one-class content based classification might be

Table 2: mF_1^{max} scores on three datasets for OCMMMF and WNMF

Dataset	Synthetic		MIRFLICKER-25K		ICDAR2003	
σ	WNMF	OCMMMF	WNMF	OCMMMF	WNMF	OCMMMF
0.20	0.8007	0.8317	0.7481	0.7577	0.5962	0.6458
0.25	0.8101	0.8211	0.7439	0.7398	0.5819	0.6436
0.30	0.8022	0.8231	0.7365	0.7347	0.5636	0.6229
0.35	0.8037	0.8114	0.7147	0.7296	0.5554	0.6265
0.40	0.7751	0.7835	0.7078	0.7257	0.5517	0.6189

a promising direction. Also, it is interesting to speed up the one-class MMMF algorithm using tricks similar to the ones proposed by researchers for speeding up one-class MMMF.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under grants 61503315 and 61571379, and the Xiamen University of Technology High Level Talents Project under grant YKJ14020R.

References

[1] R. M. Bell and Y. Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[2] Y. Chen, X. S. Zhou, and T. S. Huang. One-class svm for learning in image retrieval. In *International Conference on Image Processing*, volume 1, pages 34–37. IEEE, 2001.

[3] C. H. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SIAM International Conference on Data Mining*, volume 5, pages 606–610, 2005.

[4] Q. Gu, J. Zhou, and C. H. Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *SDM*, pages 199–210. SIAM, 2010.

[5] M. J. Huiskes and M. S. Lew. The mir flickr retrieval evaluation. In *International Conference on Multimedia Information Retrieval*, New York, NY, USA, 2008. ACM.

[6] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[7] D. D. Lee and H. S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791, 1999.

[8] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562. MIT Press, 2000.

[9] Z. Li, J. Liu, X. Zhu, T. Liu, and H. Lu. Image annotation using multi-correlation probabilistic matrix factorization. In *International Conference on Multimedia*, pages 1187–1190. ACM, 2010.

[10] Z. Lin, G. Ding, M. Hu, J. Wang, and X. Ye. Image tag completion via image-specific and tag-specific linear sparse reconstructions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1618–1625, June 2013.

[11] W. Liu, G. Hua, and J. Smith. Unsupervised one-class learning for automatic outlier removal. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3826–3833, June 2014.

[12] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. Icdar 2003 robust reading competitions. In *International Conference on Document Analysis and Recognition*, volume 2, pages 682–682. IEEE Computer Society, 2003.

[13] L. M. Manevitz and M. Yousef. One-class svms for document classification. *Journal of Machine Learning Research*, 2:139–154, 2002.

[14] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.

[15] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *International Conference on Data Mining*, pages 502–511. IEEE, 2008.

[16] J. Platt et al. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

[17] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, July 2001.

[18] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *International Conference on Machine Learning*, pages 807–814, 2007.

[19] N. Srebro, J. D. M. Rennie, and T. S. Jaakola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1329–1336. MIT Press, 2005.

[20] V. N. Vapnik. *Statistical learning theory*. Wiley, Sept. 1998.

[21] Q. Wang, B. Shen, S. Wang, L. Li, and L. Si. Binary codes embedding for fast image tagging with incomplete labels. In *European Conference on Computer Vision*, pages 425–439, 2014.

[22] L. Wu, R. Jin, and A. K. Jain. Tag completion for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):716–727, 2013.