TAG RECOMMENDATION VIA ROBUST PROBABILISTIC DISCRIMINATIVE MATRIX FACTORIZATION

Cheng Lu[†]*, Bin Shen*[‡]*, Lu Zhang*[†] *and Jan Allebach*[†]

[†]School of Electrical and Computer Engineering [‡]Department of Computer Science
Purdue University, West Lafayette, IN 47907, USA

ABSTRACT

Low-rank matrix factorization serves as a key technique in learning latent factor models for many applications in machine learning. However, in many applications, observed data often exhibits different levels of noise. To address this issue, we propose a Robust Probabilistic Discriminative Matrix Factorization (RPDMF) method for binary matrix factorization on noise polluted data. We illustrate the benefits of our approach in real examples, and show how our method significantly outperforms Probabilistic Discriminative Matrix Factorization (PDMF) and classical method Weighted Nonnegative Matrix Factorization (WNMF) in the application of image tag completion.

Index Terms— Matrix factorization, image tag completion, data recovery, binary matrix

1. INTRODUCTION

Low-rank matrix factorization serves as a key technique in learning latent factor models for many applications. Most matrix factorization methods seek to represent the original matrix as the product of two low-rank matrices. Typical applications of matrix factorization include image annotation [1], image tag completion [2, 3, 4], collaborative prediction [5] and clustering [6]. For any specific real application [7, 8], the corresponding optimality criterion is defined so that the difference between the original matrix and its factorized form is expected to be minimized. Matrix factorization serves as a very effective method to address problems of missing data recovery and prediction. Because matrix factorization can recover missing data without help of extra features, it can be applied to different fields without designing new features which may require domain-specific knowledge.

Among all collaborative filtering problems [9, 10, 11], image tag completion is a very typical application in matrix factorization. Different from movie ratings and recommendations [12, 13], the matrix representation of image tag contains only binary elements. A positive sample in the matrix means that a certain tag data is associated with a given image, while a negative sample means that the information described by this tag has not been assigned to the given image.

Many methods [14, 15, 16] have been proposed for matrix factorization. However, these methods assume that the observed matrix is noise-free. Besides, the data in typical matrix factorization with missing elements can take on any arbitrary real value. The factorization is performed so that an objective function is minimized. The objective function is usually composed of the observed data in the original matrix and a regularizer that controls the model complexity. Some factorization methods are performed under certain restrictions. For example, Nonnegative Matrix Factorization (NMF) [14, 17], as suggested by its name, requires that all elements of the original matrix be nonnegative.

In this paper, motivated by the image tag completion task, we discuss a specific setting of matrix factorization with entries restricted to binaries, representing positive and negative samples respectively. Also in this setting, some observed elements are polluted by noise. In order to perform the data recovery, we mask some of the elements in the original matrix as unknown. An example of observed matrix is

$$\begin{pmatrix} ? & n_s & p_s & p_s \\ p_s & ? & p_s & n_s \\ n_s & p_s & ? & n_s \end{pmatrix}$$

,

where p_s represents a positive sample, n_s is a negative sample, and the '?' represents our masked missing data. We should notice that some of the elements with value p_s or n_s are mislabelled (flipped) in the observed matrix due to the noise. As we discussed previously, image tag completion falls exactly into this category. Given the original matrix representation X of an image set $X_{ij} = p_s$ means that the ith image has the tag indexed as j, while $X_{ij} = n_s$ means that the ith image does not include the information of tag j. We may lose some tags because the user input is not trusted, or because the information conveyed by a certain tag is difficult to distinguish in a given image. Those tags are represented by '?' in the matrix. And the noise (mislabelling) may be introduced during transmission or by human error.

We propose Robust Probabilistic Discriminative Matrix Factorization (RPDMF) in order to recover those missing elements which are labeled as '?' in the noisy binary matrix, i.e., to predict whether any missing element is either p_s or n_s.

Another method proposed specifically to deal with missing data in a matrix is Weighted Nonnegative Matrix Factorization (WNMF) [18], which excludes missing data in the cost function by introducing a masking matrix as part of the optimization.

2. ROBUST PROBABILITY DISCRIMINATIVE MATRIX FACTORIZATION

Given a matrix $X \in S^{m\times n}$, $S = \{p_s, n_s\}$ with missing values, let G denote the set of observed elements in the matrix X. All these observed entries are either 1 or -1, i.e., $p_s = 1$, $n_s = -1$. All the missing elements are denoted as 0. Given the observed set G, our goal is to predict weather $X_{ij} = 1$ or -1 for all $X_{ij} \notin G$.

We need to find a low-rank matrix $\hat{X} = \hat{W} \times \hat{H}^T$ to approximate the target matrix X. We obtain $\hat{X}$ by minimizing a linear combination of the norms of W and H, which are two regularizers intended, respectively, to avoid overfitting, and to control its logistic loss:

$$\min_{W\in R^{m\times p},\, H\in R^{n\times p}} C \sum_{(i,j)\in G} \log(1 + e^{-X_{i,j}\langle W_{i.}, H_{j.}\rangle}) + \alpha\|W\|_F^2 + \beta\|H\|_F^2. \tag{1}$$

Here α and β are parameters controlling the strength of regularizers for W and H. In our case, $\alpha = \beta$ since they are of equal importance. Since the logistic loss has probabilistic interpretation [19], we call this Probabilistic Discriminative Matrix Factorization (PDMF).

In real-world applications, the training data X may be polluted by noise. To handle the noise, we propose a robust version of PDMF that is called Robust Probabilistic Discriminative Matrix Factorization (RPDMF). For any $X_{i,j}$, we introduce a random variable $I_{i,j}$ where $I_{i,j} = 1$, if $X_{i,j}$ is not polluted; and $I_{i,j} = 0$, otherwise.

$$\min_{W\in R^{m\times p},\, H\in R^{n\times p}} C \sum_{(i,j)\in G} I_{i,j} \log(1 + e^{-X_{i,j}\langle W_{i.}, H_{j.}\rangle}) + \alpha\|W\|_F^2 + \beta\|H\|_F^2 - q \sum_{(i,j)\in G} I_{i,j}. \tag{2}$$

Here, q is parameter that controls the noise level; and C is the box constraint.

2.1. Optimization

We now discuss the optimization shown in (2) with respect to $W \in R^{m\times p}$, $H \in R^{p\times n}$ and $\{I_{i,j}|(i,j) \in G\}$. Joint optimization with respect to W, H, and $\{I_{i,j}|(i,j) \in G\}$ is very difficult due to nonconvexity of the cost function. However, if we optimize only one of W, H, or $\{I_{i,j}|(i,j) \in G\}$ at a time, the problem becomes easy to solve. So we propose an alternate optimization method by repeating following three steps until convergence is reached.

Note that PDMF is a special case RPDMF, when all $I_{i,j}$ are set to 1 for $(i,j) \in G$.

Step 1: Fix H and $\{I_{i,j}|(i,j) \in G\}$, optimize W. Optimize (2) with respect to W.

$$\begin{aligned} W^* &= \arg\min_{W\in R^{m\times p},\, H\in R^{n\times p}} C \sum_{(i,j)\in G} I_{i,j} \log(1 + e^{-X_{i,j}\langle W_{i.}, H_{j.}\rangle}) \\ &\quad + \alpha\|W\|_F^2 + \beta\|H\|_F^2 - q \sum_{(i,j)\in G} I_{i,j} \\ &= \arg\min_{W\in R^{m\times p},\, H\in R^{n\times p}} \sum_{(i,j)\in G} I_{i,j} \log(1 + e^{-X_{i,j}\langle W_{i.}, H_{j.}\rangle}) \\ &\quad + \alpha\|W\|_F^2. \end{aligned} \tag{3}$$

We can further decompose the problem in (3) into a set of independent subproblems, where each subproblem is optimization over a row of W. If we assume that we need to optimize with respect to i-th row of W denoted as $W_{i.}$

$$W_{i.}^* = \arg\min_{W_{i.}\in\mathcal{R}^p} \frac{1}{2}\|W_{i.}\|_F^2 + C \sum_{(i,j)\in G} I_{i,j} \log(1 + e^{-X_{i,j}\langle W_{i.}, H_{j.}\rangle}). \tag{4}$$

Then the problem above become a standard convex optimization problem which can be easily solved by any gradient method.

Step2: Fix W and $\{I_{i,j}|(i,j) \in G\}$, optimize H. Symmetrically, we optimize respect to each row of H in the same manner.

Step 3: Fix W and H, optimize $\{I_{i,j}|(i,j) \in G\}$.

This problem can be decomposed into a set of independent problems, each of which responds to $I_{i,j}$:

$$\begin{aligned} I_{i,j}^* &= \arg\min_{I_{i,j}} C I_{i,j} \log(1 + e^{-X_{i,j}\langle W_{i.}, H_{j.}\rangle}) \\ &\quad + \alpha\|W\|_F^2 + \beta\|H\|_F^2 - q I_{i,j} \\ &= \arg\min_{I_{i,j}} C I_{i,j} \log(1 + e^{-X_{i,j}\langle W_{i.}, H_{j.}\rangle}) - q I_{i,j}. \end{aligned} \tag{5}$$

Then the optimization can be easily done by setting $I_{i,j}$ to 1 if $\log(1 + e^{-X_{i,j}\langle W_{i.}, H_{j.}\rangle}) < q/C$; and setting $I_{i,j}$ to 0 otherwise.

We summarize these three steps in the Algorithm 1. Overall, in every iteration, we first fix H, I and update all rows of W. Then we fix W, I and update all rows of H. Finally, we fix W, H and update I. In each step, a convex optimization problem is solved by a gradient-descent method. 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 computation.

Our proposed algorithm is guaranteed to converge, since the objective function is lower bounded by zero, and each of the updating steps can only decrease the objective function, or leave it unchanged.

Algorithm 1 One-Class Maximum Margin Matrix Factorization

Require: $X \in \{1, -1\}^{m \times n}$ with G, the set of observed entries; p, the dimension of the latent space.

```
Initialize W ∈ R^{m×p}, H ∈ R^{n×p}
for t = 1,...,max_iter do
    for i = 1,..., m do
        Update W_{i.}.
    end for
    for i = 1,..., n do
        Update H_{i.}.
    end for
    for i, j ∈ G do
        Update I_{i,j}.
    end for
end for
return W, H
```

3. EXPERIMENTAL RESULTS

In order to evaluate our proposed RPDMF method, we first conduct an experiment on a **synthetic dataset**. Then we apply PDMF to the task of image tag completion. The two public datasets used for performing image tag completion are **NUS-WIDE TAGGED** [20] and **MIRFLICKR-25K** [21].

To create the **synthetic dataset**, we first generate two base matrices $W' = (w'_{ij})_{m \times p}$ and $H' = (h'_{ij})_{n \times p}$, where the elements of W' and H' are uniformly distributed $w'_{ij} \sim U[0, 1]$, $h'_{ij} \sim U[0, 1]$. Then we threshold the matrix $X' = W' \times (H')^T$ to obtain the binary matrix X so that approximately 50% of elements in X are positive samples while the rest are negative samples.

The **NUS-WIDE TAGGED** dataset includes $269,648$ images and 81 associated tags (*e.g* airport, animal, beach, bear, *etc.*). So the original X' is a $269,648 \times 81$ matrix in which each row represents a tagged image, while each column represents a possible tag. If the ith image has a specific tag j, then the $X'(i, j)$ should be a positive sample. 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 10 tags. Because of this preprocessing, the matrix $X = (x_{ij})_{m \times n}$ to be factorized has much fewer rows than the original matrix X'.

Similarly, **MIRFLICKR-25K** contains $25,000$ tagged images with 38 different tags. We generate a matrix representation to this dataset as described above. Again, we apply preprocessing to this datasets to exclude the images which provide insufficient tag information. On **MIRFLICKR-25K**, we only use images that have more than 12 tags. 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
NUS_WIDE TAGGED	89	81	40
MIRFLICKR-25K	104	38	20

We compare RPDMF with the competing methods PDMF and weighted nonnegative matrix factorization (WNMF).

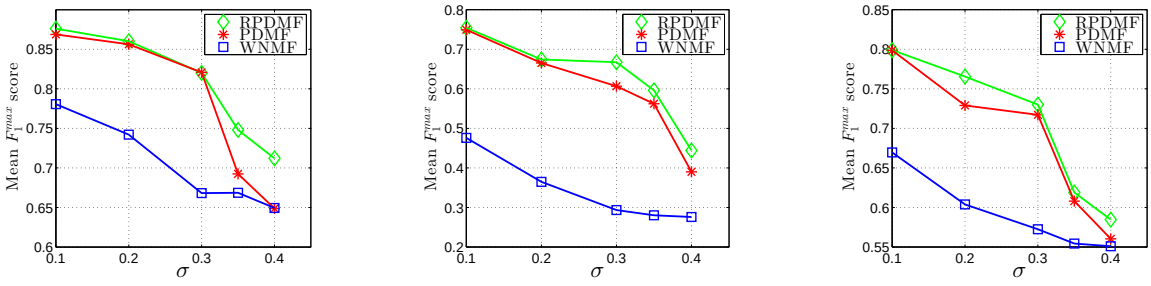
Now we discuss the parameters and performance measures for these three methods. On all three datasets, we label the positive samples in X as 1. However, due to the different nature of WNMF, PDMF, and RPDMF, we need to label negative samples in X differently depending on which method is applied. For WNMF, we label negative samples as 0, since X_{ij} should be nonnegative value. For PDMF and RPDMF, we should label negative samples as -1. After that, we randomly mask 20% of elements in X, and use them as the testing set while the rest are used as the training set. For PDMF and RPDMF, the masked elements are labeled as 0. These are expected to be recovered. For WNMF, we can achieve masking by specifying the weight matrix M in the objective cost function:

$$O_{wnmf} = ||M \odot (X - WH)||_2^F, \tag{6}$$

where $m_{ij} = 0$ if this element is masked for testing, while $m_{ij} = 1$ if it is in the training set. In order to test the robustness of these three methods, we randomly flip a portion of σ elements in the training set. This process introduces noise into our training set as discussed in Sec. 2. A larger value of σ means that the training data is more heavily polluted.

For faster convergence of RPDMF, we initialize $W^0 = (w^0_{ij})_{m \times p}$, $H^0 = (h^0_{ij})_{n \times p}$ as $w^0_{ij} \sim N(0, 1)$, $h^0_{ij} \sim N(0, 1)$, and $I_{ij} = 1$. On all the datasets, our experiments suggest that 30 iterations are sufficient for RPDMF to reach convergence. In every iteration k, we sequentially optimize all rows of W^k followed by all rows of H^k, and finally I. To update each row $W^k_{i.}$, all the rows of H^k which correspond to non-zero entries in X are viewed as samples. And we update each row $H^k_{j.}$ symmetrically. We use *BFGS* Quasi-Newton approach [22] in every iteration to update from W^k, H^k to W^{k+1}, H^{k+1}. Our experiment show that BFGS has the fastest convergence compared to other gradient methods such as *DFP* or *Conjugate Gradient* [22]. After 30 iterations, we calculate $\hat{X}' = W^{30} \times (H^{30})^T$. For the WNMF method, we simply update W^k and H^k in very iteration as in [18]. Our experiments suggest that 40 iterations are sufficient for WNMF to reach convergence. So $\hat{X}' = W^{40} \times (H^{40})^T$ for WNMF. For all the three methods, $\hat{X}'$ is binarized at threshold value T to obtain the recovered binary matrix $\hat{X}$. Finally, we evaluate the performance of factorization in the testing set, which are those masked elements that we choose at the beginning.

For both PDMF and RPDMF, different values for box constraint C and threshold T will generate different estimates



(a) Synthetic (b) NUS-WIDE TAGGED (c) MIRFLICKR-25K

Fig. 1: Testing results on three datasets for three methods, as a function of noise level

Table 2: mF_1^{max} scores on three datasets for RPDMF, PDMF, and WNMF

Dataset	Synthetic			NUS-WIDE TAGGED			MIRFLICKR-25K		
σ	WNMF	PDMF	RPDMF	WNMF	PDMF	RPDMF	WNMF	PDMF	RPDMF
0.1	0.7807	0.8688	**0.8761**	0.4760	0.7493	**0.7544**	0.6696	0.7987	**0.7990**
0.2	0.7421	0.8563	**0.8602**	0.3649	0.6654	**0.6743**	0.6039	0.7290	**0.7655**
0.3	0.6682	**0.8208**	0.8201	0.2936	0.6067	**0.6672**	0.5725	0.7171	**0.7301**
0.4	0.6686	0.6925	**0.7481**	0.2804	0.5623	**0.5962**	0.5544	0.6081	**0.6190**
0.5	0.6494	0.6487	**0.7121**	0.2761	0.3902	**0.4442**	0.5510	0.5604	**0.5847**

$\hat{X}$. 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 five times for each method, at every level of σ, to obtain five 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 at different noise levels $\sigma = 0.1$, 0.2, 0.3, 0.35, 0.4 are given in Table 2. As a visualization of Table 2, we also present our testing results in Fig. 1.

According to Table. 2 and Fig. 1, we can conclude that RPDMF achieves the highest mF_1^{max} score when compared to WNMF and PDMF on all three datasets in very case but one. Even for that particular case, its score is very close to that of the best result. Our method significantly outperforms WNMF in terms of mF_1^{max} score on all three datasets. We can also see that the performance of PDMF degrades more rapidly compared to RPDMF, as we increase the number of polluted (flipped) elements. This points to the robustness of our algorithm. In Fig. 1, we see that the mF_1^{max} scores of RPDMF drop faster in the tails compared to WNMF as σ increases, primarily because the performance of WNMF is already at a very low level. Since our original matrix is binary, it should be noted that any result close to $P(X_{ij} = p_s)$ means that the factorization provides little practical value. That is because the probability of a random guess for any element in this matrix should be $P(X_{ij} = p_s)$.

However, RPDMF is more computationally expensive compared to WNMF and PDMF, according to our experiment. Even though we avoid joint optimization and apply the Quasi-Newton approach BFGS, we still need to do intensive convex optimization in every iteration when updating $W_{i\cdot}^k$ or $H_{j\cdot}^k$ based on 'samples' in the other matrix. So the computational complexity of RPDMF grows exponentially with the size of the original matrix X. In contrast, WNMF utilizes simple gradient-based method to reach convergence for both W^k and H^k. In every iteration of WNMF, W^k and H^k update once respectively, and the updates involve only simple matrix multiplication. In terms of computational complexity, RPDMF is almost equivalent to multiple (≈ 10) iterations of PDMF, because we need to tune one more parameter q in (2).

4. CONCLUSION

In this paper, we presented a new method RPDMF, which can be used to recover missing data in a noisy binary matrix. This method is motivated by the real-word application: image tag completion. In RPDMF, we introduce the logistic loss into the cost function and optimize two base matrices W and H alternately to reach convergence. We evaluate RPDMF on three datasets, and compare it with the competing methods WNMF and PDMF. According to our experiment, we see that RPDMF has a significant advantage over WNMF and PDMF in terms of the F_1 measure when dealing with noisy matrices.

5. REFERENCES

[1] Zechao Li, Jing Liu, Xiaobin Zhu, Tinglin Liu, and Hanqing Lu, "Image annotation using multi-correlation probabilistic matrix factorization," in *International Conference on Multimedia*. ACM, 2010, pp. 1187–1190.

[2] Lei Wu, Rong Jin, and Anil K Jain, "Tag completion for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 716–727, 2013.

[3] Qifan Wang, Bin Shen, Shumiao Wang, Liang Li, and Luo Si, "Binary codes embedding for fast image tagging with incomplete labels," in *European Conference on Computer Vision*, 2014, pp. 425–439.

[4] Zijia Lin, Guiguang Ding, Mingqing Hu, Jianmin Wang, and Xiaojun Ye, "Image tag completion via image-specific and tag-specific linear sparse reconstructions," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 1618–1625.

[5] Yehuda Koren, Robert Bell, and Chris Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[6] Chris HQ Ding, Xiaofeng He, and Horst D Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering.," in *SIAM International Conference on Data Mining*, 2005, vol. 5, pp. 606–610.

[7] Deepak Agarwal and Bee-Chung Chen, "flda: matrix factorization through latent dirichlet allocation," in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 91–100.

[8] Chao Liu, Hung-chih Yang, Jinliang Fan, Li-Wei He, and Yi-Min Wang, "Distributed nonnegative matrix factorization for web-scale dyadic data analysis on mapreduce," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 681–690.

[9] Wen-Yen Chen, Jon-Chyuan Chu, Junyi Luan, Hongjie Bai, Yi Wang, and Edward Y Chang, "Collaborative filtering for orkut communities: discovery of user latent behavior," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 681–690.

[10] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001, pp. 285–295.

[11] John S Breese, David Heckerman, and Carl Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.

[12] Prem Melville, Raymond J Mooney, and Ramadass Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *AAAI/IAAI*, 2002, pp. 187–192.

[13] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King, "Sorec: social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 931–940.

[14] Daniel D. Lee and H. Sebastian Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*. 2000, pp. 556–562, MIT Press.

[15] Nathan Srebro, Jason D. M. Rennie, and Tommi S. Jaakola, "Maximum-margin matrix factorization," in *Advances in Neural Information Processing Systems*. 2005, pp. 1329–1336, MIT Press.

[16] Andriy Mnih and Ruslan Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2007, pp. 1257–1264.

[17] Daniel D. Lee and H. Sebastian Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[18] Quanquan Gu, Jie Zhou, and Chris HQ Ding, "Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs.," in *SDM*. SIAM, 2010, pp. 199–210.

[19] Maximilian Nickel and Volker Tresp, "Logistic tensor factorization for multi-relational data," *arXiv preprint arXiv:1306.2084*, 2013.

[20] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng, "Nus-wide: a real-world web image database from national university of singapore," in *Proceedings of the ACM international conference on image and video retrieval*. ACM, 2009, p. 48.

[21] Guangyu Zhu, Shuicheng Yan, and Yi Ma, "Image tag refinement towards low-rank, content-tag prior and error sparsity," in *Proceedings of the international conference on Multimedia*. ACM, 2010, pp. 461–470.

[22] Edwin KP Chong and Stanislaw H Zak, *An introduction to optimization*, vol. 76, John Wiley & Sons, 2013.