Kernelized Probabilistic Matrix Factorization for Collaborative Filtering: Exploiting Projected User and Item Graph

Bithika Pal
Indian Institute of Technology Kharagpur
West Bengal, India
bithikapal@iitkgp.ac.in

Mamata Jenamani
Indian Institute of Technology Kharagpur
West Bengal, India
mj@iem.iitkgp.ac.in

ABSTRACT

Matrix Factorization (MF) techniques have already shown its strong foundation in collaborative filtering (CF), particularly for rating prediction problem. In the basic MF model, the use of additional information such as social network, item tags along with rating has become popular and effective, which results in making the model more complex. However, there are very few studies in recent years, which only use the users rating information for the recommendation. In this paper, we present a new finding on exploiting *Projected User* and *Item Graph* in the setting of Kernelized Probabilistic Matrix Factorization (KPMF), which uses different graph kernels from the projected graphs. KPMF works with its latent vector spanning over all users (and items) with Gaussian process priors and tries to capture the covariance structure across users and items from their respective projected graphs. We also explore the ways of building these projected graphs to maximize the prediction accuracy. We implement the model in five real-world datasets and achieve significant performance improvement in terms of RMSE with state-of-the-art MF techniques.

CCS CONCEPTS

• **Computing methodologies** → **Gaussian processes**; **Factorization methods**; • **Information systems** → **Recommender systems**;

KEYWORDS

Recommendation; Collaborative Filtering; Graph Kernel; Matrix Factorization; Projected User and Item Graph

ACM Reference Format:
Bithika Pal and Mamata Jenamani. 2018. Kernelized Probabilistic Matrix Factorization for Collaborative Filtering: Exploiting Projected User and Item Graph. In *Twelfth ACM Conference on Recommender Systems (RecSys '18), October 2–7, 2018, Vancouver, BC, Canada.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3240323.3240402

1 INTRODUCTION

Rating prediction and personalized item recommendation are two fundamental problems in the recommender system. In solving these problems, factorization based algorithms have achieved wide popularity and success [3–8]. The basic working principle lies on the assumption of low-rank rating matrix and factorizing the users and items into low dimensional latent space. MF techniques solve the rating prediction problem as a matrix completion problem and find its latent factor matrices by minimizing the prediction error. Among several variants of MF model, Kernelized MF is a different one where latent space can be non-linear and also can capture high dimensional feature space by considering the distance-matrix [5, 8]. In [8], the authors have proposed KPMF using the additional social or item information and used the graph kernel for capturing the variance of users from the social network. Though a lot of current studies have the focus on the use of social network and item context, recently, in [5], the authors have proposed one direction of forming dictionary based kernel without using extra side information. However, MKMF does not take the covariance factors of user and items. In this paper, we propose a method to capture the covariance in terms of graph kernels using only rating information. For this, we create a projected user graph and an item graph from the rating matrix based on certain criteria and use the generated graph kernels in kernelized probabilistic matrix factorization setting. We also try to analyze the property of the projected graphs which can produce the best result. We apply the model to the existing social network based model and observe better result in using projected graphs. The experimentation of the model is done on five real-world datasets using seven key MF techniques and significant improvement in accuracy is achieved. Below, we describe the key related works used in performance comparison and the organization of the paper.

Key Literatures. In [2], the rating matrix $R \in \mathbb{R}^{M \times N}$ is considered as multiplication of user latent matrice $P \in \mathbb{R}^{M \times D}$ and item latent matrix $Q \in \mathbb{R}^{N \times D}$ $[R = PQ^T]$ and optimize it by iteratively updating the P and Q. Each row of P signifies the interest of a user in each of the D latent features and each row of Q tells that how much an item contributes in each of the D latent features. The issue in this model is its inability of predicting the rating for new users and capturing users biases. In [3, 4], the users bias (b_u) and item bias (b_i) is also considered and captured the predicted rating for an user u to an item i as $r_{u,i} = \mu + b_u + b_i + P_{u,:}.{Q_{i,:}}^T$ where μ is global mean rating ($u \in [M]$, $i \in [N]$). This model is further extended by incorporating the neighborhood information [3]. Among several variants of MF models, one well-established model is Probabilistic Matrix Factorization (PMF) [6]. Here, each row features of P and Q (each user latent feature $P_{u,:}$ and each item latent feature $Q_{i,:}$) is generated from Gaussian distribution (used as prior), and P , Q are estimated using MAP inference that maximizes log-likelihood with respect to P and Q. Extension of PMF is done using fully Bayesian treatment where hyperparameters for the priors are considered with Gaussian-Wishart conjugate [7]. In [8], KPMF is introduced

RecSys '18, October 2–7, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5901-6/18/10…$15.00
https://doi.org/10.1145/3240323.3240402

using side information, where instead of each row of P and Q, the columns of P and Q are captured from a generative process; i.e., instead of each user, the individual latent feature from all over the user is generated from the Gaussian process. Same is done for the item latent feature as well. The user's covariance and item's covariance of this process are chosen from graph kernel. Similar to PMF, MAP estimate is taken for P and Q in KPMF. In [5], dictionary-based kernel is introduced. A random dictionary is generated for each latent feature mapping the feature in a higher dimension. Then, from the dictionary, multiple kernels are built as a gram-matrix. The authors have also given a model (MKMF) for blending multiple kernels with determining the contribution of each individual kernel, similar to the mixture model. All the mentioned models are referred for comparison.

The following paper is organized as follows. Section 2 describes the proposed methodology. Section 3 and Section 4 discuss about the experiment and results analysis. Finally, Section 5 concludes the work.

2 METHODOLOGY

First, we describe the KPMF model using projected graphs. Next, we formally define the projected user and item graph and their creation strategies.

2.1 Model

As mentioned earlier, the prior distribution of the columns are generated from a Gaussian Process (GP) which is defined by mean function and covariance function. Let, $K_P \in \mathbb{R}^{M\times M}$ and $K_Q \in \mathbb{R}^{N\times N}$ denote the full covariance matrix for all users and items. These K_P and K_Q generation process will be discussed later.

The generation process of KPMF is mentioned below.

(1) For each column of P, $P_{:,d} \sim GP(0, K_P)$.
(2) For each column of Q, $Q_{:,d} \sim GP(0, K_Q)$.
(3) For each existing rating of R, $r_{u,i} \sim \mathcal{N}(P_{u,:}Q^T_{i,:}, \sigma^2)$, σ is a constant.

The priors over P and Q are given in Equation 1 and 2 respectively.

$$p(P|K_P) = \prod_{d=1}^{D} GP(P_{:,d}|0, K_P) \tag{1}$$

$$p(Q|K_Q) = \prod_{d=1}^{D} GP(Q_{:,d}|0, K_Q) \tag{2}$$

The likelihood over the observed entries of the rating matrix R given P and Q is mentioned in Equation 3 and $\delta_{u,i}$ is 1 if $r_{u,i}$ is present, else 0.

$$p(R|P, Q, \sigma^2) = \prod_{u=1}^{M}\prod_{i=1}^{N}[\mathcal{N}(r_{u,i}|\mathcal{N}(P_{u,:}Q^T_{i,:}, \sigma^2))]^{\delta_{u,i}} \tag{3}$$

The log-posterior over P and Q is given in Equation 4.

$$\begin{aligned}\log p(P, Q|R, \sigma^2, K_P, K_Q) = &-\frac{1}{2\sigma^2}\sum_{i=1}^{M}\sum_{j=1}^{N}\delta_{u,i}(r_{u,i} - P_{u,:}Q^T_{i,:})^2 \\ &-\frac{1}{2}\sum_{d=1}^{D}P^T_{:,d}K_P^{-1}P_{:,d} - \frac{1}{2}\sum_{d=1}^{D}Q^T_{:,d}K_Q^{-1}Q_{:,d} \\ &-(\sum_{u=1}^{M}\sum_{i=1}^{n}\delta_{u,i})\log\sigma^2 \\ &-\frac{D}{2}(\log|K_P| + \log|K_Q|) + C\end{aligned} \tag{4}$$

where $|K|$ is the determinant of K and C is a constant.

For maximizing the log-posterior, to learn P and Q, we have the minimization function as shown in Equation 5 which is to be optimized.

$$\begin{aligned}E = &\frac{1}{2\sigma^2}\sum_{i=1}^{M}\sum_{j=1}^{N}\delta_{u,i}(r_{u,i} - P_{u,:}Q^T_{i,:})^2 \\ &+\frac{1}{2}\sum_{d=1}^{D}P^T_{:,d}K_P^{-1}P_{:,d} \;+\; \frac{1}{2}\sum_{d=1}^{D}Q^T_{:,d}K_Q^{-1}Q_{:,d}\end{aligned} \tag{5}$$

Equation 5 can be rewritten as

$$E = \frac{1}{\sigma^2}\sum_{i=1}^{M}\sum_{j=1}^{N}\delta_{u,i}(r_{u,i} - P_{u,:}Q^T_{i,:})^2 + \mathrm{Tr}(P^TK_P^{-1}P) + \mathrm{Tr}(Q^TK_Q^{-1}Q) \tag{6}$$

We perform gradient descent for minimizing E and the gradients are calculated on P and Q seperately. Here, the parameters to learn are $\Theta = \{P, Q\}$. In each iteration $t+1$, the algorithm updates Θ as below.

$$\Theta^{(t+1)} = \Theta^{(t)} - \eta\frac{\partial E}{\partial \Theta} \tag{7}$$

2.2 Building K_P and K_Q

User covariance and item covariance are captured in K_P and K_Q. The projected user graph and item graph are created for building K_P and K_Q respectively.

Projected User Graph $\mathcal{G}_P$: The vertex set $\mathcal{U}$ of $\mathcal{G}_P$ is the set of M users in rating. The edge set of $\mathcal{G}_P$ is created based on function $f_{\mathcal{G}_P} : \mathcal{U}\times\mathcal{U} \rightarrow \{0, 1\}$. There exist an edge between two vertices if $f_{\mathcal{G}_P}$ is 1.

Projected Item Graph $\mathcal{G}_Q$: The vertex set $\mathcal{I}$ of $\mathcal{G}_Q$ is the set of N items in rating. The edge set of $\mathcal{G}_Q$ is created based on function $f_{\mathcal{G}_Q} : \mathcal{I}\times\mathcal{I} \rightarrow \{0, 1\}$. There exist an edge betweem two vertices if $f_{\mathcal{G}_Q}$ is 1.

Here, the $f_{\mathcal{G}_P}$ is chosen based the number of common items rated by two users and if the count is greater than some threshold value, then $f_{\mathcal{G}_P} = 1$. Similarly, the $f_{\mathcal{G}_Q}$ is chosen based on the number of common users who have rated the item. If the count is greater than some threshold value, then $f_{\mathcal{G}_Q} = 1$.

The threshold value of this count plays a vital role in building $\mathcal{G}_P$ and $\mathcal{G}_Q$. For example, if the threshold is chosen as 1, i.e., if two

users have purchased at least one common item, an edge will be created between two of them. In reality, this event is very likely to happen, which results in the $\mathcal{G}_P$ as a very dense graph. Same is true for $\mathcal{G}_Q$ as well. Now in reverse, if the threshold is very high the graph will become extremely sparse, then it will result in missing covariance (discussed later).

2.2.1 Graph Kernel. The K_P and K_Q is chosen as the graph kernels from the projected graphs $\mathcal{G}_P$ and $\mathcal{G}_Q$. We consider the adjacency matrix of a graph as A where $A_{i,j} = 1$ if there is an edge between vertex i and j. The Laplacian of a graph is described as $L = D - A$ where D is a diagonal matrix and each of it represents the degree of that node. Graph kernel gives a way for capturing the underlying structure of the graph. The connectivity among nodes along with the degree of a node, the number of reachability between two nodes can be captured in this kernel. Due to the inherent ability of the laplacian to capture the neighborhoodness and similarity [1], in this paper, we consider regularized Laplacian (RL) kernel. It is given in Equation 8 and 9, where $L_{\mathcal{G}_P}$ and $L_{\mathcal{G}_Q}$ symbolizes Laplacian of the respective projected user and item graph and $\gamma > 0$ is constant.

$$K_P^{RL} = (I + \gamma L_{\mathcal{G}_P})^{-1} \tag{8}$$

$$K_Q^{RL} = (I + \gamma L_{\mathcal{G}_Q})^{-1} \tag{9}$$

There is a relation between this kernel and covariance and it has the following properties.

- Sum of each row or each column is 1.
- Between any two nodes, if the connecting path length increases, then the covariance decreases.
- If the connecting node is of relatively higher degree, then their covariance is less.
- Variance of each individual vertex is also dependent on the degree of that vertex. Higher Degree signifies low variance.

The other graph kernels like Diffusion Kernel, Commute Time Kernel can also be exploited.

3 EXPERIMENTS

We perform the experiments for three objectives.

- Compare the recommendation accuracy with existing standard MF techniques.
- Compare the results obtained using social information and without using social information.
- Analyze the fair choice of threshold in $f_{\mathcal{G}_P}$ and $f_{\mathcal{G}_Q}$.

Dataset Description. We use five datasets ciaoDVD, FilmTrust, Epinions, ML-100k and ML-1m (Movielens) for experimentation. CiaoDVD and FilmTrust datasets are taken from librec data[1]. Epinions dataset is taken from [8]. Movielens data is downloaded from Grouplens[2]. Movielens does not have social information so taken all datapoints. For FilmTrust and CiaoDVD, the users present in both social and rating data is selected and item set is construted with the item which has at least 2 ratings. The statistics of the dataset used for the experiment is given in Table 1.

[1]https://www.librec.net/datasets.html
[2]https://grouplens.org/datasets/movielens/100k/

Table 1: Dataset Description

dataset	#users	#items	#rating	#social tie
CiaoDVD	2068	3174	21338	17308
FilmTrust	529	1357	13741	1439
Epinions	1000	1500	26398	77722
ML-100K	943	1682	100000	-
ML-1M	6040	3706	1000209	-

Experimental Setup. In the experimental setup, we select 80% of the rating dataset as training data and 20% as test data. From training dataset, we take 20% ratings as validation data. We run this for 5 times and report the average results. For training in KPMF, rating value is normalized as rating/max(rating) and denormalized for evaluation. For comparison, we use six MF techniques as mentioned earlier, named as FunkSVD [2], Korens et al. SVD [4], SVD++ [3], PMF [6], BPMF [7], MKMF [5] and one social relation based KPMF-Soc [8].

For our proposed methods, KPMF-U denotes the use of $\mathcal{G}_P$ for building K_P and diagonal matrix with $\gamma = 0.2$ as K_Q. Similarly, in KPMF-I, K_P is used as diagonal matrix with $\gamma = 0.2$ and K_Q is constructed using $\mathcal{G}_Q$. The KPMF-UI denotes use of both $\mathcal{G}_P$ and $\mathcal{G}_Q$. In preperation of $\mathcal{G}_P$, for movielens, the threshold is chosen from $[1, 5, 10, 20, 30, 40]$ and for others from $[1, 2, 3, 4, 5, 10]$. Similary, for $\mathcal{G}_Q$, for movielens, threshold is chosen from $[10, 20, 30, 40]$ and for others from $[1, 2, 3, 4, 5]$.

For all the models, latent Factor is tried with D from $[5, 10, 20, 30]$. For MKMF, dictionary size is taken from $[50, 100, 300]$ as mentioned in the paper. All the results are reported with either best parameter settings or the parameter setting mentioned in the paper.

Evaluation Metric. As we are minimizing the mean error in all the models, here, we report the *root mean squared error* (RMSE) for performance comparison (Equation 10).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(r_i - \hat{r}_i)^2}{n}} \tag{10}$$

where r_i is the actual rating, $\hat{r}_i$ is the predicted rating and n is the total number of test data set entry.

4 RESULTS AND DISCUSSION

In this section, we discuss the results based on our experiments. The results for comparison of RMSE over all the methods are given in Table 2. The boldly faced values denote the best result and the underlined values show the best result from existing methods in each dataset and Improve(%) column tells about the improvement of the proposed model from the best existing methods. First, we will analyze the results technique wise and then dataset wise.

From the existing methods, SVD has given the best results in all the datasets, (except filmtrust and ML-1m), which uses the user and item bias and replace the mean rating if it is not able to find from PQ^T. On the other hand, FunKSVD and MKMF perform worst in most of the cases except MKMF in movielens. Also, for all the datasets, SVD++ could not perform better than SVD. Another observation is that BPMF always outperforms over PMF in all datasets which has been found in literature as well. From the results, it can

Table 2: Results of RMSE for Performace Comparison of Recommendation Accuracy

Dataset / Paper	FunkSVD [2]	SVD [4]	SVD++ [3]	PMF [6]	BPMF [7]	KPMF-Soc [8]	MKMF [5]	KPMF-U Our	KPMF-I Our	KPMF-UI Our	Improve(%)
CiaoDVD	1.3692	0.9207	0.9326	1.2122	1.1480	1.007	1.8103	**0.8787**	1.0126	0.9162	4.56
FilmTrust	1.3105	0.8145	0.8302	0.9913	0.9744	0.7582	1.3978	0.7474	0.7466	**0.7371**	2.78
Epinions	1.6923	**1.0892**	1.0907	1.4160	1.359	1.1469	1.4511	1.1077	1.1056	1.0899	-0.06
ML-100k	1.4297	0.9315	0.9375	1.1274	1.1214	-	0.9689	0.9290	0.9348	**0.9232**	0.89
ML-1m	1.4306	0.8877	0.9231	1.0024	0.9987	-	0.8755	0.8613	0.8617	0.8554	2.3

be observed that MKMF can perform well if the rating information is dense and each item has a large number of co-purchased users, which holds in both the movielense datasets (also reported same in literature). Now, KPMF-Soc performs fair in comparison with others but can not outperform the proposed ones. From the three proposed models, KPMF-UI performs best, except in CiaoDVD.

For, CiaoDVD dataset, RMSE in KPMF-I is high, which causes KPMF-UI to get more RMSE than KPMF-U. For FilmTrust dataset, KPMF-UI outperforms all the models and delivers the highest improvement compared to SVD. Also, all the kernelized methods perform well compared to other MF techniques in FilmTrust. In case of Epinions dataset, SVD performs best but with very slight improvement. For movielens dataset, ML-1m achieves a significant improvement compared to ML-100k using our proposed methods. In three of our proposed models, we report the best result achieved from different projected graphs. We also observe the heatmap of the adjacency matrix of the projected graphs (not reported here). It shows that high density projected graphs are incapable of finding covariance. As discussed earlier in the properties of the projected graph, that higher degree of vertices cause very less variance and diminishes the covariance after a certain length.

Table 3: RMSE Comparison with Different Graph Creation Threshold for ML-100k

Threshold ($\mathcal{G}_P$, $\mathcal{G}_Q$)	KPMF-U	KPMF-I	KPMF-UI
30, -	0.9290	-	-
40, -	0.9455	-	-
-, 10	-	0.9451	-
-, 20	-	0.9356	-
-, 30	-	0.9348	-
-, 40	-	0.9433	-
5, 5	-	-	1.003
10, 10	-	-	0.9648
20, 20	-	-	0.9388
30, 20	-	-	0.9359
40, 30	-	-	0.9232

In Table 3, we show a sample result for selecting the best value from the thresholds. Results are shown for the ML-100k dataset, with some of the run results. Same is performed for the all the datasets with different combinations mentioned in the experimental setup. For decreasing the running time, we first take the SGD results and choose that threshold parameter going for GD version. The results in Table 3, shows that after increasing the threshold up to a certain limit, RMSE increases. This is caused as the entry of the adjacency matrix diminishes. So, K_P and K_Q could not capture the intrinsic structure of the graph. The process for selecting the right threshold can become more expensive when the number of users and items are very large. In that case, if the rating matrix is extremely sparse the threshold can be selected as one.

5 CONCLUSION

In this paper, we have built different projected user and item graphs from the rating information and used the projected graphs in kernelized probabilistic matrix factorization. We have used a simple threshold-based approach to build the projected graphs. The proposed model has shown significant performance improvement compared to the baseline methods. In results, we have also shown that the use of the projected graph dominates the social counterpart in KPMF. However, the model suffers from the scalability issue as the building of the projected graphs consumes a significant amount of time. This issue can be further investigated. In the future, we also want to analyze the performance of the model using different graph kernels.

ACKNOWLEDGMENTS

The work has been financially supported by the project *E-business Center of Excellence* funded by Ministry of Human Resource and Development (MHRD), Government of India under the scheme of *Center for Training and Research in Frontier Areas of Science and Technology (FAST)*, Grant No. F.No.5-5/2014-TS.VII.

REFERENCES

[1] Mikhail Belkin and Partha Niyogi. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*. 585–591.
[2] Simon Funk. 2006. Netflix update: Try this at home.
[3] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.
[4] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).
[5] Xinyue Liu, Chara Aggarwal, Yu-Feng Li, Xiaugnan Kong, Xinyuan Sun, and Saket Sathe. 2016. Kernelized matrix factorization for collaborative filtering. In *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 378–386.
[6] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.
[7] Ruslan Salakhutdinov and Andriy Mnih. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th international conference on Machine learning*. ACM, 880–887.
[8] Tinghui Zhou, Hanhuai Shan, Arindam Banerjee, and Guillermo Sapiro. 2012. Kernelized probabilistic matrix factorization: Exploiting graphs and side information. In *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 403–414.