# User-Specific Rating Prediction for Mobile Applications via Weight-based Matrix Factorization

Jingke Meng, Zibin Zheng, Guanhong Tao
School of Data and Computer Science
Sun Yat-sen University
Guangzhou, China
{sysumengjke, zibin.gil, gwinhen}@gmail.com

Xuanzhe Liu
Institute of Software
Peking University
Beijing, China
liuxuanzhe@pku.edu.cn

***Abstract*—With the dramatic growth of mobile application (app) markets, users can find various apps with any functionalities they desire in these markets. However, the huge amounts of apps make it quite a challenge for users to discover good apps efficiently. Previous studies recommend apps by considering all apps equal without capturing the specific interests of each individual user. To address this problem, we propose a model called Weight-based Matrix Factorization (WMF), which can capture user-specific interests and give a more accurate prediction on these apps. WMF views each user as a document and each app as a word, and calculates the weight of each app for target users. The weights are calculated by employing term frequency inverse document frequency (TF-IDF) algorithm, which are then introduced into matrix factorization to predict app ratings. Comprehensive experiments are conducted on a real-world datasets with 5057 users and 4496 apps. The experimental results show that WMF achieves a convincing performance and surpasses other existing prediction models.**

***Index Terms*—Mobile application; Rating prediction; User specific; Matrix factorization;**

## I. Introduction

Mobile Internet is a much more convenient way to share a variety of services comparing to traditional Internet. Mobile apps play a significant role in the environment of Mobile Internet. Google Play[1] (a mobile app marketplace on Android Platform), has more than 1.4 million apps by the time of 2015. The massive amounts of apps cover almost every aspects of people's everyday life, which has become a great challenge for users to find the app that fits their interests. Therefore, an effective app recommender system is in the urgent need to address these problems.

However, not all the apps reflect users' personal interests equally. For instance, if an instant messaging (IM) app and a game app have similar user ratings but with different numbers of users, then the game app with fewer users is more important for the target user as the IM app is installed almost by everyone which can barely show the specific interests of users. The game app, instead, presents the interest of users in a small specific group. Most of the previous work failed to capture such differentiation and consider all apps equally.

In this paper, we view each installed app as a feature for describing the user's personal interests, as shown in Fig. 1. A user is depicted by apps with different weights, the weight of app is expressed by the size of the icon. To clearly explain the problem, we give a definition of user-specific interest as follows.


Fig. 1. An example of TF-IDF

***Definition** 1 (User-Specific Interest):* This represents the apps with high weights for the target user, which can be used to express user's distinctive interest.

The problem is giving a more accurate rating prediction on apps with the user-specific interest. The contributions of this paper are summarized as follows:

- First, we regard each user as a document and each app as a word, and calculate the importance of apps for target users to capture the specific interests of users.
- Second, app weights denoting users' specific interests are introduced into the matrix factorization model[1] to generate much more accurate rating prediction.
- Finally, large-scale experiments with approximate 5057 users and 4496 apps are conducted on a real-world dataset to study the efficiency of our proposed method comparing with other state-of-the-art approaches. The related dataset

[1] $https://play.google.com/store$

978-1-5090-2675-3/16 $31.00 © 2016 IEEE
DOI 10.1109/ICWS.2016.104

is published in our website[2].

The structure of the paper is shown as follows. Section II discusses related work that has been carried out in recent years. Section III illustrates our prediction framework. Section IV proposes our Weight-based Matrix Factorization algorithm. The experiments are presented in Section V. Finally, we conclude our work in Section VI.

## II. Related Work

Functions of apps and interests of users are two aspects to leverage for app recommendation. Small-Crowd model [2] ranked the installed apps to distinguish apps at reflecting users' personal interests by combining the global download information with fine-grained app usage records. Beyond this, each app may have latent factors that influence the decision of users to download apps. Therefore, Yin *et al.* [3] modeled such latent factors into two values: *actual value* and *tempting value*, where the former one stood for the real satisfaction the app brings to the user while the later one was the estimated satisfaction the app may provide. Besides the latent values of apps, the feedbacks generated by users are also valuable information for recommendation. Context-aware recommendation can utilize such feedback including the diverse range of implicit mobile data, and Djinn [4] made full use of such advantages. What's more, how users actually use apps presents their daily usage habits, which indicates their interests and intentions that can be used to improve the recommendation [5][6]. Most of the above methods were based on the usage information of users, which may cause the cold-start problem[7]. To address such situations, Lin *et al.* [8] utilized nascent information culled from Twitter. Authors created pseudo-documents that contained the IDs of Twitter users interested in an app and applied latent Dirichlet allocation [9] to generate latent groups. However, most of these studies failed to capture users' personal interests in mobile application precisely. Yang *et al.* [2] just leveraged apps as features for describing user's personal interests, but they did not give a further study on the user-specific interests, they only proposed a collaborative filtering methods to do personalized recommendation with all the apps.

## III. The Prediction Framework

The prediction framework is illustrated in Fig. 2. As shown in the figure, users' downloads and uninstall app logs are obtained from mobile app markets, and uploaded to our prediction server. Then, by analyzing user usage logs, we find out user-specific apps and predict ratings for these apps.

## IV. Weight-based Matrix Factorization

In this section, we describe our WMF model in detail. Firstly, we present the weight computation of apps. And then we discuss our WMF algorithm.

### A. Weight Computation

We employ the Term Frequency-Inverse Document Frequency (TF-IDF) [10][11] algorithm to calculate the weights of apps for each given user in order to find user-specific apps. TF-IDF is a numerical statistic method that intends to reflect how important a word is to a document in a collection or corpus. It is widely used as a weighting factor in information retrieval and text mining.

[2] $www.inpluslab.com/icws2016$

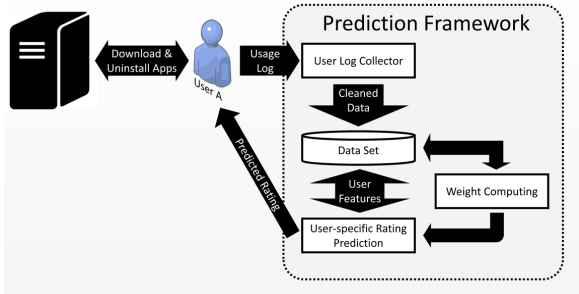


Fig. 2. Weight-based rating prediction framework

The original TF-IDF is calculated as:

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D), \tag{1}$$

where $t$ is a word, $d$ is a document, and $D$ is the collection of documents. $tf(\cdot)$ is the number of times that word $t$ occurs in the document $d$. And

$$idf(t,D) = log(\frac{N}{1+|\{d \in D : t \in d\}|}), \tag{2}$$

where $N = |D|$ is the total number of documents in the corpus. $|\{d \in D : t \in d\}|$ is the number of documents where the word $t$ appears. In order to avoid division-by-zero, we adjust the denominator to $1 + |\{d \in D : t \in d\}|$.

Learning from the idea of TF-IDF, we view a user as a document, an app as a word, and all apps as the corpus. The definition of app weight $w_{i,j}$ is shown as follows:

$$w_{i,j} = t_{i,j} \times d_{i,j}, \tag{3}$$

$$d_{i,j} = 1 + log(\frac{|U|}{|\{n : a_j \in u_i\}|}), \tag{4}$$

where $t_{i,j}$ denotes the ratings of user $u_i$ over $a_j$, which is calculated by analyzing users' usage logs and the details are discussed in Table II of Section V(A). $d_{i,j}$ is the reversed app download rate. $|U|$ is the number of all the users and $|\{n : a_j \in u_i\}|$ is the number of the user $u_i$ who download app $a_j$.

After computing the weights of apps, the apps with different weights can be used to describe user's interest, as shown in Fig. 1.

### B. WMF Algorithm

Before describing our WMF algorithm, we first discuss the PMF model.

Suppose we have $m$ users, $n$ apps, and the integer rating values range from 1 to 5. Let $q_{i,j}$ represents the rating of user $u_i$ for app $a_j$, $U \in R^{l \times m}$ and $V \in R^{l \times n}$ be user and app latent feature matrices, where the column vectors $U_i$ and $V_j$ represent user and app latent feature vectors, respectively. As shown in

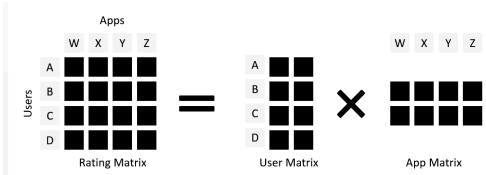


Fig. 3. An example of MF

Fig. 3, in order to predict missing user-app rating values, the rating matrix is decomposed into two low-rank matrices, and further predictions are based on such decomposition.

The objective function of PMF with quadratic regularization terms is the following optimization problem [12]:

$$\min_{U,V} L(U,V) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{i,j}(q_{i,j} - U_i^T V_j)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2, \quad (5)$$

where $I_{i,j}$ is the indicator function which is equal to 1 if user $u_i$ rated app $a_j$ and 0 otherwise. $\|U\|_F^2$ and $\|V\|_F^2$ denote the Frobenius norm[3], and $\lambda_U$ and $\lambda_V$ are two regularization parameters. The optimization problem in Equation (5) minimizes the sum-of-squared-errors objective function with quadratic regularization terms by adopting Gaussian observation noise.

However, PMF predicts the rating values of apps with the same weight, which fails to capture the most important apps. In our WMF algorithm, we introduce $w_{i,j}$ as the weight of app $a_j$ for user $u_i$ calculated by Equation (3). Thus, the missing rating values are predicted by optimizing the following objective function:

$$\min_{U,V} L(U,V) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} w_{i,j} I_{i,j}(q_{i,j} - U_i^T V_j)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2, \quad (6)$$

From Equation (5) and (6), we can observe the following:

- We introduce $w_{i,j}$ into Equation (6), which indicates app's weight calculated by Equation (3). In Equation (6), $w_{i,j}$ can be seen as the penalty term of deviation between predicted values and actual values.
- If $w_{i,j}$ is large, the value of $(q_{i,j} - U_i^T V_j)^2$ will be strengthened in Equation (6). This result means that the apps with high weights are given more attention, and the predicted values based on high-weight apps will be more accurate.
- If $w_{i,j}$ is small, the value of $(q_{i,j} - U_i^T V_j)^2$ will be weakened. This result means that the apps with small weights are given less attention, and the predicted values based on low-weight apps will lead to large deviations.

To get a minimum of the objective function in equation (6), we apply the gradient descent algorithm to iteratively recover user and app latent space on both $U_i$ and $V_i$. The gradients can be computed by:

$$\frac{\partial L}{\partial U_i} = \lambda_U U_i - \sum_{j=1}^{n} w_{i,j} I_{i,j}(q_{i,j} - U_i^T V_j)V_j, \quad (7)$$

$$\frac{\partial L}{\partial V_i} = \lambda_V V_i - \sum_{i=1}^{m} w_{i,j} I_{i,j}(q_{i,j} - U_i^T V_j)U_i^T. \quad (8)$$

The WMF algorithm procedure is summarized in Algorithm 1.

**Algorithm 1** WMF algorithm

**Require:** Training matrix $Q$; Weight matrix $W$;
**Ensure:** $U$, $V$;

```
Initialize U and V with small random numbers; I = 0;
Calculate the objective function L;
Update ∂L/∂U_i;
Update ∂L/∂V_i;
while I < MaxI do
    for i = 1 → m do
        U'_i ← U_i − α∂L/∂U_i;
    end for
    for j = 1 → n do
        V'_i ← V_i − α∂L/∂V_i;
    end for
end while
```

$I$ is the iteration number in the gradient descent algorithm, and $MaxI$ is the number of the max iteration. The parameter $\alpha$ is the iteration step which is used to control the speed of iterations.

## V. Experiments

In this section, we conduct experiments on the dataset with 5,057 users and 4,496 apps to evaluate the efficiency of our WMF approach and compare with other state-of-the-art collaborative filtering methods.

### A. Dataset Description

Our dataset is from Wandoujia[4], a leading Android app marketplace in China. Similar to Google Play, Wandoujia provides functionalities for users to manage app including search, browse, download, install, update and uninstall. This dataset is the log of users' download and uninstall records ranging from May 1, 2014 to July 31, 2014. which includes 688,198 log records of 5,057 users and 4,496 apps. The detailed log sample is given in Table I.

TABLE I
Dataset Information

| Type | Status | UserID | AppID | Timestamp |
|---|---|---|---|---|
| download | success | 1 | com.sina.weibo | 2014-5-2 |
| uninstall | failure | 5 | com.taobao.taobao | 2014-6-9 |
| download | failure | 9 | com.tencent.mm | 2014-5-24 |
| uninstall | success | 5 | com.whatsapp | 2014-7-9 |

[3] $http://mathworld.wolfram.com/FrobeniusNorm.html$

[4] $http://www.wandoujia.com$

TABLE II
QUANTIFICATION OF RATINGS

| Download | Update | Uninstall | Rating |
|---|---|---|---|
| failure | no | no | 1 |
| success | no | yes | 2 |
| success | yes | yes | 3 |
| success | no | no | 4 |
| success | yes | no | 5 |

In order to quantify these statuses of users' attitudes towards apps, we map various combinations into $[1, 5]$ discrete score domain. Table II shows the full combinations of different statuses.

### B. Evaluation Metrics

The mean absolute error (MAE) is widely used to measure the difference between the estimated values and observed values. In this paper, we adopt a measurement metric that is similar to ordinary MAE with top-k selection process.

*Theorem 1 (TMAE):* Given a set of estimated values $\widetilde{q}_{i,j} \in \mathcal{S}$, the objective of TMAE is to calculates the MAE value of estimated values at the top k with the interest of target objects:

$$TMAE = \frac{1}{k} \sum_{\widetilde{q}_{i,j} \in \mathcal{T}(i)} | q_{i,j} - \widetilde{q}_{i,j} |, \tag{9}$$

$$\mathcal{T}(i) = \{\widetilde{q}_{i,j} \,|j \leq k, \widetilde{q}_{i,j} \in \underset{S}{sort}(w_{i,j} |\, \widetilde{q}_{i,j} \in S)\}, \tag{10}$$

where $sort(\cdot)$ arranges estimated values in descending order based on the weight $w_{i,j}$ defined in Equation (3).

### C. Performance Comparison

To prove the effectiveness of our WMF method, we conduct extensive experiments on state-of-the-art method PMF to compare with our method.

In our experiment, data proportion is defined as the proportion of the training data with respect to the whole dataset. In the evaluation process, each rating prediction method is run on 4 different test cases whose proportion are 60%, 70%, 80% and 90%, respectively. In specific, a data proportion of 60% means that 60% of the known rating entries in the matrix are used for predicting missing rating values. The known rating entries occupy 3% of the total rating matrix.

Table III shows the MAE results of different methods with different proportions from 60% to 90%, where KPMF gives the TMAE values of PMF and WMF also uses the TMAE metric. In all the experiments, we set $k = 10$. We can observe that our WMF approach obtains smaller MAE values consistently under different proportions of test data. The experiment results demonstrate that our approach is realistic and reasonable and outweighs other existing state-of-the-art methods.

TABLE III
MAE ACCURACY COMPARISON ON RATING PREDICTION

| Methods | 60% | 70% | 80% | 90% |
|---|---|---|---|---|
| PMF | 0.4551 | 0.4349 | 0.4233 | 0.4490 |
| KPMF | 0.4717 | 0.4316 | 0.4107 | 0.4389 |
| WMF | 0.4421 | 0.4171 | 0.4044 | 0.4338 |
| Improvement | 2.85% | 3.36% | 1.53% | 1.13% |

## VI. CONCLUSION

Based on the intuition that not all the apps reflect users' personal interests equally, we view each installed app as a feature to describe the users' personal interests and predict more accurate ratings on apps with respect to user-specific interests. We propose a WMF model and the experiments conducted on a real-world dataset show that WMF achieves a convincing performance and surpasses other existing prediction models.


## ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Project No. 61472338), Guangdong Natural Science Foundation (Project No. 2014A030313151), and the Fundamental Research Funds for the Central Universities.


## REFERENCE


[1] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[2] C. Yang, T. Wang, G. Yin, H. Wang, M. Wu, and M. Xiao, "Personalized mobile application discovery," in *Proceedings of the 1st International Workshop on Crowd-based Software Development Methods and Technologies*, 2014, pp. 49–54.

[3] P. Yin, P. Luo, W.-C. Lee, and M. Wang, "App recommendation: a contest between satisfaction and temptation," in *Proceedings of the sixth ACM international conference on Web search and data mining*, 2013, pp. 395–404.

[4] A. Karatzoglou, L. Baltrunas, K. Church, and M. Böhmer, "Climbing the app wall: enabling mobile app discovery through context-aware recommendations," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 2527–2530.

[5] B. Yan and G. Chen, "Appjoy: personalized mobile application discovery," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, 2011, pp. 113–126.

[6] Z.-X. Liao, P.-R. Lei, T.-J. Shen, S.-C. Li, and W.-C. Peng, "Mining temporal profiles of mobile applications for usage prediction," in *Proc. 12th Int'l Conf. Data Mining Workshops (ICDMW'12)*, 2012, pp. 890–893.

[7] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, "Addressing cold-start problem in recommendation systems," in *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, 2008, pp. 208–211.

[8] J. Lin, K. Sugiyama, M.-Y. Kan, and T.-S. Chua, "Addressing cold-start in app recommendation: latent user models constructed from twitter followers," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, 2013, pp. 283–292.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[10] H. P. Luhn, "A statistical approach to mechanized encoding and searching of literary information," *IBM Journal of research and development*, vol. 1, no. 4, pp. 309–317, 1957.

[11] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[12] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2007, pp. 1257–1264.