Leveraging tagging and rating for recommendation: RMF meets weighted diffusion on tripartite graphs

Jianguo Li\textsuperscript{a}, Yong Tang\textsuperscript{a}, Jiemin Chen\textsuperscript{a,}\textsuperscript{*}

\textsuperscript{a}School of Computer Science, South China Normal University, Guangzhou, China

Abstract

Recommender systems (RSs) have been a widely exploited approach to solving the information overload problem. However, the performance is still limited due to the extreme sparsity of the rating data. With the popularity of Web 2.0, the social tagging system provides more external information to improve recommendation accuracy. Although some existing approaches combine the matrix factorization models with co-occurrence properties and context of tags, they neglect the issue of tag sparsity without the commonly associated tags problem that would also result in inaccurate recommendations. Consequently, in this paper, we propose a novel hybrid collaborative filtering model named WUDiff\textsubscript{RMF}, which improves Regularized Matrix Factorization (RMF) model by integrating Weighted User-Diffusion-based CF algorithm (WUDiff) that obtains the information of similar users from the weighted tripartite user-item-tag graph. This model aims to capture the degree correlation of the user-item-tag tripartite network to enhance the performance of recommendation. Experiments conducted on four real-world datasets demonstrate that our approach significantly performs better than already widely used methods in the accuracy of recommendation. Moreover, results show that WUDiff\textsubscript{RMF} can alleviate the data sparsity, especially in the circumstance that users have made few ratings and few tags.

Keywords: Regularized matrix factorization, Collaborative filtering, Tag, Diffusion, Tripartite graphs

\textsuperscript{*}Corresponding author

Email address: chenjiemin@m.scnu.edu.cn (Jiemin Chen)
1. Introduction

The recent decade has witnessed the rapidly increasing amount of information available on the Internet, which make people engulf in a wide variety of information. To solve this problem, recommender systems (RSs) have been applied for suggesting products in different domains, such as music recommendation in Last.fm, movie recommendation at Netflix and video recommendation at YouTube. The objective of RSs is to assist users to find out suitable items based on the users’ preference in the past.

As a prominence of recommendation techniques, collaborative filtering (CF) methods [1] recommend items based on the preference of other similar items or like-minded users, including the neighborhood method and the matrix factorization model [2]. The former intends to predict a user’s rating on an item from like-minded users or similar items. By comparison, the latter uses the user-item matrix to map both users and items into a latent factor space for representing their relationship with high accuracy and scalability. Traditional CF methods generally only use the user-item rating matrix for recommendation. However, because the density of available ratings is often less than 1%, traditional CF methods suffer from data sparsity and the cold-start problem that remarkably reduce the system performance [3,4].

With the dramatic development of Web 2.0, social tag systems have emerged and become popular, which allow users to freely assign, organize and share items with tags [5]. For instance, the song “Roll in the Deep” are evaluated by 307,915 users and labeled with 61 tags on Last.fm, where 5091 users tag the song with “amazing voice” and “best of 2011”. User-defined tags are used as a means to represent their preference and evaluations for the items. Therefore, based on the intuition that users’ tagging history can be utilized to improve the performance of recommender systems, the research of tag-based recommendation has become popular, especially tag-based CF methods [6]. Most of them are unified frameworks based on the technique of matrix factorization, which firstly use tagging data to obtain the neighborhood information of each item (user), and then incorporated the information into a probabilistic factor analysis model [6,7]. However, in some real-world scenarios, the user rating data and tagging data are very sparse, so the neighborhood information may be inaccurate. For example, suppose Kent, Mark and Sam love comedy movies. Kent has rated three comedy movies with the high ratings and Mark has rated other five comedy movies with his high ratings. At the same time, Mark has labeled this movie “Dead pool” with two
tags and Sam has labeled the movie with other three tags. Due to the lack of co-rated items, rating-based CF methods may ignore that Kent and Mark share common preference on comedy movies. And without co-occurrence, some tagging-based CF methods also may neglect that Mark and Sam share similar preference. The inaccurate information may result in misleading recommendations.

The preference of users should include tags and ratings. Consequently, inspired by the idea of network-based models from statistical physics, we propose the RMF with WUDiff (WUDiff_RMF) hybrid recommendation approach. Firstly, we represent users and items with a joint latent factor space of dimensionality by RMF. Secondly, we obtain the relationship of similar users by Weight User-Diffusion algorithm (WUDiff), which apply a diffusion process to generate recommendations in a user-item-tag tripartite graph. Finally, we expand the RMF by integrating similar user regularization term which describes that the target user would be influenced by similar users in the weighted tripartite network. A series of experiments are conducted on four popular real-world datasets to validate the effectiveness of WUDiff_RMF. The results confirm that our approach can effectively improve the accuracy of predictions with the other counterparts. And the further analysis indicates it can alleviate the data sparsity problem especially in datasets with each user making sparse ratings and tagging data.

The rest of the paper is organized as follows. Section 2 briefly describes related work. Section 3 describes preliminaries, revisits the RMF model and the Udiff recommendation algorithm. We describe a new, more accurate hybrid recommendation model in Section 4. The experimental results are presented and analyzed in Section 5. Finally, we conclude the paper and give future research directions in Section 6.

2. Related work

Tagging systems provide users with a promising way to freely associate tags with web resources, which are good at describing users’ opinions. Thus, researchers endeavor to improve recommendation accuracies by utilizing tagging data, especially for those CF methods. In this section, we review several major tag-based CF recommendation approaches, which are generally divided into the neighborhood method and the model method.

The measurement of similarity between users or items has played a vital role in the neighborhood-based CF. By taking the relationships among
tags into consideration, the co-occurrence properties and context of tags can be employed to obtain the neighbors of each user and each item. Sen et al.[8] constructed implicit and explicit tag-based recommendation algorithms based on the user-tag rating matrix. Wang et al.[9] improved the tag-based neighborhood method by using tagging data to generate latent topics. Qi et al.[10] computed users’ similarities by utilizing the inferred tag ratings for improving user-based CF method. Gedikli et al.[11] improved the item-based CF by incorporating tag preferences in the context of an item. In addition, from a perspective of graph theory, a user-item-tag-based network can be viewed as a tripartite network[5]. Zhang et al.[12] firstly proposed a tag-aware diffusion-based method(ODiff) that represents tags as nodes in a tripartite graph and utilizes a diffusion process to obtain better recommendations. Shang et al.[13] proposed an user-based hybrid tag algorithm by harnessing diffusion-based method(UDiff). Trinity is introduced in [14], where a random walk with restart model is proposed based on a three-layered object-user-tag heterogeneous network.

As one of the most successful and popular model-based CF methods, Matrix factorization (MF) includes Regularized Matrix Factorization (RMF) [15], Probabilistic Matrix Factorization (PMF) [16], principle component analysis (PCA)[17], latent Dirichlet allocation (LDA) [18] and so on. MF can capture overall structure that associates with most or all items, but overlooks strong associations among a small set of closely related neighbor set[6]. Recently, a trend in the literature is the use unified frameworks that combine neighbor-based methods and MF models to handle the data sparsity problem [7,19]. Zhou et al.[20] proposed a factor analysis approach called TagRec based on a unified probabilistic matrix factorization by utilizing both users tagging information and rating information. Wu et al.[6] built a two-stage recommendation framework, named NHPMF, by using the tagging data to select neighbors of each user and each item. Zhang et al.[21] proposed a recommendation model based on clustering of users (UCMF) by considering the neighbors’ impact on the interest of each user in the same latent factor space. Chen et al.[22] proposed TRCF model that improved recommendation performance by capturing the semantic correlation between users and items. Additionally, from the view of feature level, Zhang et al.[23] presented the feature-centric recommendation approach that utilized user’s feature preferences to improve recommendation of items.

Existing approaches incorporates localized relationships into MF model by using co-occurrence properties and context of tags for finding the set of
neighbors. If the user only associates very few tags with items, these approaches cannot obtain better performance. The feedback information of users should include tags and ratings. Therefore, we propose the WUDiff-based RMF model (WUDiff_RMF), which is a novel unified recommendation framework based on tags and ratings. Different from previous work, based on the information of similar users from the weighted tripartite user-item-tag network, we integrate similar user regularization term into RMF for improving the performance of recommendation.

3. Preliminaries

In a standard setting of CF, we use $U = \{u_1, u_2, ..., u_m\}$ to denote the set of users, the set of items is $I = \{i_1, i_2, ..., i_n\}$ and the set of tags is $T = \{t_1, t_2, ..., t_k\}$. $R|U| \times |I|$ is the user-item rating matrix and $R'|U| \times |T|$ is the user-tag tagging matrix. $r_{ui}$ is the rating by user $u$ of item $i$, which can be binary or integers from a given range (e.g. $r_{ui} \in [1, 5]$). Notations used in this paper are summarized in Table 1.

| symbol | meaning |
|--------|---------|
| $\kappa$ | a set of $r_{ui}$ that are known |
| $f$ | the number of latent factor |
| $\hat{r}_{ui}$ | the predicted value of $r_{ui}$ |
| $p_u, q_i$ | $f$ dimension vectors of user $u$ and item $i$ |
| $R(u)$ | the set of items that are rated by user $u$ |
| $S(u)$ | the set of users who are similar to user $u$ by WUDiff |
| $A$ | user-item adjacent matrix |
| $a_{ij}$ | an edge between user $i$ and item $j$ |
| $A'$ | user-tag adjacent matrix |
| $a'_{ik}$ | an edge between user $i$ and tag $k$ |
| $k(\text{user}_i)$ | the degree of user $i$ |
| $k(\text{item}_j)$ | the degree of item $j$ |
| $k(\text{tag}_k)$ | the degree of tag $k$ |

3.1. Regularized Matrix Factorization

RMF is an efficient and effective approach to RSs among a large number of solutions to RSs [24]. The fundamental principle of RMF maps both users
and items to a low-dimensional feature space of dimensionality $f$, and utilizes the latent factor space to make further missing data prediction. Accordingly, $P \in \mathbb{R}^{U \times f}$ denotes the user feature matrix and each user $u$ is associated with a latent fact vector $p_u$ that represents users personal interests. $Q \in \mathbb{R}^{I \times f}$ represents the item feature matrix. Each item $i$ is associated with a latent fact vector $q_i$ to describe the characteristics. Then the rating approximation of user $u$ on item $i$ could be modeled as inner products as follows:

$$\hat{r}_{ui} = p_u^T q_i$$

(1)

where $\hat{r}_{ui}$ is the estimate of $r_{ui}$, $p_u$ and $q_i$ are column vectors with $f$ values. The layer of $f$-th parameters of all vectors $p_u$ and $q_i$ is called the $f$-th feature. The feature matrix $P$ and $Q$ can be learned by minimizing the following loss (objective) function:

$$\min_{P, Q} \sum_{(u, i) \in \kappa} (r_{ui} - p_u^T q_i)^2 + \frac{\lambda_u}{2} \|P\|_F^2 + \frac{\lambda_i}{2} \|Q\|_F^2$$

(2)

where $\lambda_u, \lambda_i > 0$. In order to avoid over-fitting, two regularization terms are added into the loss function. $\|.\|_F$ denotes the Frobenius norm.

3.2. UDiff algorithm

UDiff algorithm [8] utilizes user-item-tag tripartite graph with resource allocation to recommend new users. The tripartite graph consists of two bipartite graphs: the user-item graph and the user-tag graph, which can be presented by two adjacent matrices $A$ and $A'$. If user $u$ has collected an item $i$, there is an edge between $u$ and $i$, and we set $\alpha_{ui} = 1$, otherwise $\alpha_{ui} = 0$. Analogously, if user $u$ has used a tag $t$, we set $\alpha_{ut}' = 1$, otherwise $\alpha_{ut}' = 0$. An example of the process of resource reallocation is illustrated in Figure 1.
In the user-item bipartite graph, the resource-allocation process consists of two steps: first from users to items, then back to users. Each user distributes his/her initial resource equally to all the items he/she has collected, and then each item distributes its resource equally to all the users having collected it. Thus, based on user-item bipartite network, the similarity between $u$ and $v$ is:

$$s_{uv} = \frac{1}{k(v)} \sum_{i=1}^{n} \frac{\alpha_{ui} \alpha_{vi}}{k(i)}$$ (3)

where $k(i)$ is the degree of item $i$ and $k(v)$ is the degree of user $v$ in the user-item bipartite graph. Analogously, considering the diffusion on the user-tag bipartite graph, the tag-based similarity between user $u$ and $v$ is:

$$s'_{uv} = \frac{1}{k'(v)} \sum_{t=1}^{r} \frac{\alpha'_{u't} \alpha'_{vt}}{k'(t)}$$ (4)

where $k'(t)$ and $k'(v)$ are respectively the degrees of tag $t$ and user $v$ in the user-tag bipartite graph.

Finally, the above two diffusion-based similarities are integrated by linear superposition to obtain a final redistribution matrix as:

$$s^*_{uv} = \lambda s_{uv} + (1 - \lambda) s'_{uv}$$ (5)

where $\lambda \in [0, 1]$ is a tunable parameter.
4. WUDiff_RMRF model

In this section, we present a novel hybrid collaborative filtering model named WUDiff_RMRF, which utilizes the information of similar users from WUDiff algorithm to further improve RMF. And we systematically describe how to model the neighborhood information of users based on tags and ratings as regularization terms to constrain the matrix factorization framework.

4.1. Implicit neighborhood information

As shown in Fig. 2(a), one user $u_1$ has collected item $i_1$ with tag $t_1$, user $u_2$ has collected item $i_1$ with tag $t_2$ and tag $t_3$, and item $i_2$ also has been collected by user $u_3$ with tag $t_3$. If we adopt traditional neighborhood-based CF methods to measure the similarity between users $u_2$ and $u_3$, they will not be similar at all, for the reason that there is co-rated items. Analogously, due to without co-occurrence tags, user $u_1$ and $u_2$ also will be dissimilar by the traditional tag-based CF methods. However, user $u_1$ and $u_2$ have labeled item $i_1$ with different tags. In Fig. 2(b), when user Joe and Sam give high ratings to “Someone like you ” and “Rolling in the deep ” respectively, a good recommender system should consider that they have similar preference due to two songs belonged to Adele and the co-occurrence tag. Therefore, compared to traditional similarity measures, UDiff algorithm can be directly applied in extracting the implicit information of similar users based on the degree correlation of the user-item-tag tripartite graph.

Fig. 2. The correlation of users in the tripartite graph
However, the UDiff method focuses on the unweighted graphs, and a refined method should take into account the weights of edges in the tripartite network[]. Thus, we improve UDiff algorithm by introducing user-item rating as the weights of edges on the user-item bipartite graph and user-tag labeling as the weights on the user-tag relations. Firstly, different users towards various items may differ, so it is necessary to convert individual ratings to a universal scale. Z-score normalization[25] is the most popular rating normalization scheme and considers the spread in the individual rating scales[2]. In user-based methods, the z-score normalization of $r_{ui}$ divides the user-mean-centered rating by the standard deviation $\delta_u$ the ratings given by item $u$[2]:

$$h(r_{ui}) = \frac{r_{ui} - \bar{r}_u}{\delta_u}$$ (6)

Furthermore, to compute the weight of user $u$ to tag $k$ conveniently, we adopt an adaptation of Okapi BM25 weighting scheme based on the idea that each user assigns each tag with a score[26,27].

$$w(u,t) = \log \frac{M}{n_u(t)} \cdot \frac{tf(u,t) \cdot (k_1 + 1)}{tf(u,t) + k_1 \cdot (1 - b + b \cdot \frac{|u|}{avg(U)})}$$ (7)

where $M$ is the number of users, $n_u(t)$ is the number of users who has used tag $t$, $tf(u,t)$ is the number of times user $u$ has annotated items with tag $t$. $|u| = \sum_{t=1}^{T} n_u(t)$ represents user profile size, $avg(U)$ is the average of all users’ profile size, $b$ and $k_1$ are set to the standard values of 0.75 and 2, respectively.

According to the above analysis, the resource allocation process of the WUDiff is formulated as

$$ws_{uv} = \frac{1}{k(v)} \sum_{i=1}^{n} \alpha_{ui} \alpha_{vi} h(r_{ui}) h(r_{vi}) \frac{h(r_{ui}) h(r_{vi})}{k(i)}$$ (8)

$$ws'_{uv} = \frac{1}{k'(v)} \sum_{t=1}^{r} \alpha'_{ut} \alpha'_{vt} w(u,t) w(v,t) \frac{w(u,t) w(v,t)}{k'(t)}$$ (9)

$$ws^*_{uv} = \lambda \cdot ws_{uv} + (1 - \lambda) \cdot ws'_{uv}$$ (10)
4.2. Regularization and optimization

In this section, we will incorporate additional sources of information from the user-item-tag tripartite network to improve the prediction accuracy of the RMF approach. WUDiff can obtain the neighborhood information by utilizing the information of ratings and tags, and thus can capture the implicit information of similar users based on the degree correlation of the tripartite network, not just ratings or tags. Since the behavior of users always resembles their neighbors in taste, we integrate similar user regularization term to the loss function in our model, which measures the difference between the latent feature vector of a user and those of his/her similar users from WUDiff. The mechanism of WUDiff_RMF is shown in Fig.3.

![Fig. 3. The mechanism of WUDiff_RMF](image)

There are a set I of items and a set U of users. An user group denoted by $S(u)$ is a “k-nearest-neighbors” set of user $u$ obtained by WUDiff_RMF approach, and $S(u)$ is represented as following:

$$S(u) = \{u_1^{w_{u,1}}, u_2^{w_{u,2}}, \ldots, u_k^{w_{u,k}}\}$$

(11)

where $k$ is the number of users in $S(u)$; $u_k$ is a user, and $w_{u,k}$ is the similarity user $u$ and user $u_k$.

A user-factors vector $p_u$ of the user $u$ is a vector of real numbers in the closed interval $[0,1]$, defined by

$$p_u = (p_{u,1}, \ldots, p_{u,t}, \ldots, p_{u,f})$$

(12)
where $f$ is the number of feature vectors, and $p_{u,t}$ is the typicality value of user $u$ in the feature vector $t$. If user $u$ is similar to user $v$, the ideal property is to minimize the distance between latent vectors $p_u$ and $p_v$. Hence, to improve recommender systems, we utilize regularization term $\|p_u - p_v\|_F^2$ employed in Equation (2) to utilize implicit user information from WUDiff.

The range of datasets is different, for example, the movie rating is range from 0 to 5. Thus, to learn parameters in a more convenient way, we change the raw rating $R_{ui}$ by adopting the function $f(x) = x/R_{\text{max}}$, where $R_{\text{max}}$ is the maximum of ratings in the RSs. In MovieLens, due to $R_{\text{max}} = 5$, an interval $[0,5]$ will be changed into $[0,1]$. In addition, the logistic function $g(x) = 1/(1 + \exp(-x))$ is employed to map the inner product of $p_u$ and $q_i$ into the interval $[0,1]$. Accordingly, the objective function of this approach can be formulated as:

$$
L = \min_{P,Q} \sum_{(u,i) \in \kappa} (r_{ui} - g(p_u^T q_i))^2 + \frac{\alpha}{2} \sum_{u=1}^{m} \sum_{v \in S(u)} w_{uv}^* \cdot \|p_u - p_v\|_F^2 \\
+ \frac{\lambda_u}{2} \|P\|_F^2 + \frac{\lambda_i}{2} \|Q\|_F^2
$$

(13)

where $\alpha > 0$, $\alpha$ is the regularization parameter, which balances the contribution of neighborhood information of users from WUDiff to the recommendation performance of the model. $w_{uv}^*$ indicates the similarity between user $u$ and user $v$. $S(u)$ is the set of users who are similar to user $u$ by WUDiff.

For each observed rating $r_{ui}$, we can minimize the above objective function by performing Stochastic Gradient Descent (SGD) to learn the latent parameters:

$$
p_u \leftarrow p_u + \gamma_1 (g'(p_u^T q_i)e_{ui}q_i - \sum_{v \in S(u)} w_{uv}^* \cdot (p_u - p_v)) \\
+ \alpha \sum_{v \in S(u)} w_{uv}^* \cdot (p_v - p_u) - \lambda_u p_u,
q_i \leftarrow q_i + \gamma_2 (g'(p_u^T q_i)e_{ui}p_u - \lambda_i q_i)
$$

(14)

where $e_{ui} = r_{ui} - g(p_u^T q_i)$, and the derivative of logistic function $g(x)$ is $g'(x) = \exp(-x)/(1 + \exp(-x))^2$. $\gamma_1$ and $\gamma_2$ are the learning rates.

Different from previous approaches that incorporate localized relationships into RMF model, WUDiff_RMF combines neighborhood information
of users on the degree correlation of the weighted tripartite graph based on
ratings and tags, instead of using co-occurrence properties of tags or rat-
ings for finding the set of neighbors. The learning algorithm of our model is
described in algorithm 1:

\textbf{Algorithm 1} \textsc{WUDiff\_RMF}

\textbf{Input}: \(D_T\), Test set \(D_V\), factor dimensionality \(f\), user\_num, \(R(u)\), \(S(u)\)

\textbf{Output}: Learned \textsc{WUDiff\_RMF} model parameters

1: Randomly initialize feature parameters;
2: \textbf{repeat}
3: \hspace{1em} \textbf{for} \(u = 1; u <= \text{user\_num}; u + + \) \textbf{do}
4: \hspace{2em} \textbf{for} \(i \in R(u)\) \textbf{do}
5: \hspace{3em} \(p_u \leftarrow p_u + \gamma_1 (g'(p_u^T q_i) e_{ui} q_i - \alpha \sum_{v \in S(u)} w_{uv}^* \cdot (p_u - p_v)) \)
6: \hspace{3em} + \alpha \sum_{v \in S(u)} w_{uv}^* \cdot (p_v - p_u) - \lambda_u p_u \)
7: \hspace{3em} \(q_i \leftarrow q_i + \gamma_2 (g'(p_u^T q_i) e_{ui} p_u - \lambda_i q_i) \)
8: \hspace{2em} \textbf{end for}
9: \hspace{1em} \textbf{end for}
10: \hspace{1em} update;
11: \hspace{1em} calculate RMSE on \(D_V\);
12: \textbf{until} RMSE on \(D_V\) does not improve.

5. Experiments

In this section, we report the results from experiments conducted to com-
pare the \textsc{WUDiff\_RMF} with seven existing recommendation methods by us-
ing well-known benchmark datasets. Our experiments aim to answer the
following questions:

1. How does the parameter \(\lambda\) of \textsc{WUDiff} algorithm affect the \textsc{WUDiff\_RMF} method?
2. How does the number of nearest neighbor \(k\) affect the recommendation
quality?
3. How does the parameter \(\alpha\) affect the recommendation results?
4. Is CogTime RMF more efficient than seven existing recommendation
methods?
5. Can our method obtain good performance even if users have very few ratings and few tags?

5.1. Description of Datasets

We adopt four real-world datasets, Delicious, Last.fm, DBLP, and MovieLens, to evaluate the effectiveness of our recommendation algorithm. The first two datasets are released in the conference of HetRec 2011[28]. Delicious consists of 1867 users, 69,926 URLs and 53,388 tags. Last.fm consists of 1892 users, 17,632 artists and 11,946 tags. The third data set DBLP contains 6815 authors’ ratings on 78475 papers with 81858 unique venues and authors of all papers from an academic network[23,29]. The three datasets have binary ratings. The fourth dataset Movielens, which has 1857 users’ ratings on 4721 items(scale from 0.5 to 5) with 8288 unique tags[23,28], is also released in the framework of HetRec 2011.

5.2. Evaluation metrics

We employ two widely used metrics, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), to investigate the prediction quality of our proposed WUDiff_RMF model in comparison with other collaborative filtering methods.

MAE is defined as:

\[ MAE = \frac{1}{|T_E|} \sum_{(u,i,r_{ui}) \in T_E} |r_{ui} - \hat{r}_{ui}| \]  

(15)

where \( T_E \) denotes the number of tested ratings.

RMSE is computed as follows:

\[ RMSE = \sqrt{\frac{1}{|T_E|} \sum_{(u,i,r_{ui}) \in T_E} (r_{ui} - \hat{r}_{ui})^2} \]  

(16)

Observably lower MAE and RMSE correspond to higher prediction accuracy.

5.3. Comparisons

In the experiments, to show the performance of our proposed WUDiff_RMF, we adopt two matrix factorization recommendation methods as baselines for the comparison, which include RMF and PMF without tagging data. Moreover, we compare our algorithm with five state-of-the-art algorithms with tagging: CTR, FM, RLFM, SIM and FCR-r.
• RMF: It has been described in Section 3.1 and uses only the user-rating matrix to generate recommendations.

• PMF [14]: It models user-item rating matrix by matrix factorization, including adaptive priors over the item and user feature vectors that can be used to control model complexity automatically.

• Collaborative topic regression (CTR) [30]: It combines matrix factorization and probabilistic topic modeling for the recommendation.

• Factorization machine (FM) [31]: It combines the flexibility of feature engineering with the superiority of factorization models.

• Regression latent factor model (RLFM) [32]: It is a regression based latent factor model by incorporating past interactions and the features of users and items.

• Similarity based method (SIM) [33]: It incorporates tag preferences in the context of an item by adopting the SVM regression for recommendation.

• Feature-Centric Recommendation (FCR-r) [23]: It transforms the item ratings into feature ratings and learns the global weighting for features by a regression model.

The results of PMF, CTR, FM, RLFM, SIM and FCR-r are cited from [23]. To a fair comparison, we follow the experimental settings as [23], and the results of RMF and WUDiff_RMF are obtained by our careful implementations.

5.4. Experimental Results and analysis

We conduct a series of experiments on four datasets to demonstrate the advantage of our method relative to others. To assess the performance of WUDiff_RMF, we apply the standard 10-fold cross-validation for all datasets and methods. The random selection is carried out 10 times independently, then mean of MAE and RMSE are calculated as the final results. For RMF, we use the setting $\lambda_u = \lambda_v = 0.01$. 
5.4.1. Impact of the parameter $\lambda$

In WUDiff, as a tunable parameter, $\lambda$ determines how much the information of tag should be incorporated into diffusion-based similarity in the user-item-tag tripartite network. $\lambda = 1$ and $\lambda = 0$ correspond to the cases for pure user-item and user-tag diffusions. To test the effect of the parameter on the recommendation accuracy of WUDiff, we conduct experiments by setting $\lambda$ from 0 to 1 on different datasets.

As shown in Fig.4, to compare the $\lambda = 0$ or $\lambda = 1$ case, we find that the balanced relation between taggings and ratings can help improving the algorithmic accuracy. As $\lambda$ increases, the RMSE value decreases at first, but
when $\lambda$ goes up a certain threshold like 0.6 on Delicious dataset, the RMSE value increases steeply with further increase of the value of $\lambda$. The best parameter $\lambda$ setting for other three datasets are 0.5, 0.6, and 0.3 respectively.

5.4.2. Impact of the parameter $k$

We analyze the performance of our model when varying the number of nearest neighbor $k$ on four datasets. Different $k$ will lead different recommendation accuracies. Fig.5 shows following observations. With the increment of $k$, the prediction accuracy improves at first, but when $k$ goes up a certain threshold like 40 on Delicious dataset, the prediction accuracy decreases. Accordingly, the thresholds of parameter $k$ are 40 on Last.fm, 30 on DBLP and 40 on Movielens.

![Fig. 5. Impact of Parameter $k$](image-url)
Although \( k \) increases, recommendation accuracy would be improved in KNN-based CF methods [2,6]. However, in WUDiff_RMF, when \( k \) is too small, the set of “neighbors” is not enough to improve recommendation accuracy. As \( k \) increase, there are not enough qualified “neighbors” for each user that will affect the performance of RMF in capturing overall structure.

### 5.4.3. Impact of the parameter \( \alpha \)

In WUDiff_RMF, the parameter \( \alpha \) controls the proportion of effect between rating preference and neighborhood information from the user-item-tag tripartite network when training model. In order to check the effect of the parameter \( \alpha \) for recommending, we compare the RMSE of our model for the different ranges of parameter \( \alpha \) on all datasets.

![Graphs showing the impact of \( \alpha \) on RMSE for different datasets](image)

(a) \( \alpha \) vs. RMSE, Delicious  
(b) \( \alpha \) vs. RMSE, Last.fm  
(c) \( \alpha \) vs. RMSE, DBLP  
(d) \( \alpha \) vs. RMSE, Movielens

Fig. 6. Impact of Parameter \( \alpha \)
From the results presented in Fig.6, we can observe that incorporating neighborhood information of users can actually enhance the performance of recommendation. If we set $\alpha$ as 0, it means we just employ the user-item rating matrix for making the recommendation. As $\alpha$ increases, the RMSE value decreases at first, but when $\alpha$ goes below a certain threshold like 0.04 on Delicious dataset, the RMSE value goes up with further increase of the value of $\alpha$. The best value of $\alpha$ on Last.fm, DBLP, Movielens is 0.01,0.06, and 0.01.

| Dataset   | Metrics | RMF   | PMF   | CTR   | FM    | RLFM  | SIM   | FCR-r | WUDiff |
|-----------|---------|-------|-------|-------|-------|-------|-------|-------|--------|
| Delicious | MAE     | 0.1891| 0.8081| 0.7431| 0.2906| 0.3978| 0.3872| 0.1513| 0.1482 |
|           | RMSE    | ±0.0225| ±0.0048| ±0.0005| ±0.0020| ±0.0010| ±0.0011| ±0.0009| ±0.0008 |
| Last.fm   | MAE     | 0.185 | 0.3179| 0.4084| 0.2534| 0.2235| 0.2941| 0.1066| 0.0762 |
|           | RMSE    | ±0.0024| ±0.0022| ±0.0025| ±0.0030| ±0.0015| ±0.0013| ±0.0016| ±0.0022 |
| DBLP      | MAE     | 0.1827| 0.3800| 0.3653| 0.1167| 0.1930| 0.3032| 0.0841| 0.0367 |
|           | RMSE    | ±0.0040| ±0.0020| ±0.0018| ±0.0020| ±0.0007| ±0.0003| ±0.0003| ±0.0004 |
| Movielens | MAE     | 1.0748| 0.8622| 0.8315| 0.9467| 0.8056| 0.7616| 0.7208| 0.7280 |
|           | RMSE    | ±0.0187| ±0.0197| ±0.0156| ±0.0194| ±0.0145| ±0.0123| ±0.0125| ±0.0045 |
|           |         | ±0.0182| ±0.0247| ±0.0191| ±0.0229| ±0.0215| ±0.0190| ±0.0201| ±0.0027 |

5.4.4. Prediction performance of WUDiff_RMF

As mentioned in Section 5.3, we adopt two baseline methods and five state-of-the-art models to compare with WUDiff_RMF on four datasets in terms of RMSE and MAE, in which we set $f=20$. Note that the results of PMF, CTR, FM, RLFM, SIM and FCR-r are directly cited from [13]. From Table 2, we can notice that our method outperforms other approaches on all datasets. For example, on average, WUDiff_RMF improves FCR-r by 18.43%, 7.45%, 7.89%, and 2.28% respectively, in terms of RMSE, on the four datasets. Although FCR-r performs better than our method in Movielens in terms of MAE, WUDiff_RMF is more powerful in terms of RMSE. Since RMSE is more indicative than MAE [34], our method still owns best performance.
Table 3. Statistical significance of prediction accuracy improvements

| Method                                      | t-Test  | Delicious | Last.fm | DBLP   | MovieLens |
|---------------------------------------------|---------|-----------|---------|--------|-----------|
| RMF with user-item bipartite network        | MAE     | t=22.24   | t=41.78 | t=31.78| t=12.63   |
|                                             | p=1.71E-6| p=7.41E-8| p=2.92E-6| p=p=2.76E-5 |
|                                             | RMSE    | t=5.29    | t=4.23  | t=19.71| t=5.37    |
|                                             | p=0.0016| p=0.0041  | p=2.37E-5| p=0.0015 |
| RMF with user-tag bipartite network         | MAE     | t=23.94   | t=21.83 | t=19.71| t=4.89    |
|                                             | p=1.18E-6| p=1.87E-6| p=1.95E-5| p=0.0022 |
|                                             | RMSE    | t=4.28    | t=7.02  | t=23.53| t=5.42    |
|                                             | p=0.0039| p=4.52E-4| p=9.67E-6| p=0.0014 |

*Significance at 95%.

In order to further validate the statistical significance of our experiments, we perform the paired t-test (2-tail) on RMF with the user-item bipartite network, RMF with user-tag bipartite network and WUDiff_RMF over the MAE and RMSE. The gains in accuracy are statistically significant with 95% confidence level as displayed in Table 3. The results show that the improvements of WUDiff_RMF over the other two methods are statistically significant ($p < 0.01$) on four datasets.

In conclusion, the experimental results meet our expectation that utilizing the rating information and tagging information together to find the neighbour of users is better than only utilizing users’ ratings or only utilizing users’ taggings. And hence, by incorporating the neighborhood information from WUDiff into RMF, our method obtains significant promotion on prediction accuracy.

5.4.5. Performance on Users with Different Number of Ratings and tags

When users offer few ratings and few tags or even have no rating and tag, it is an critical challenge of the RSs to make an accurate prediction. In order to analyze the capabilities of our model thoroughly, we first group all the users based on the number of the observed ratings and the number of tag assignments in the training datasets, then make predictions and evaluate the prediction accuracies on different user groups. Additionally, to interpret the results more intuitively, we include the baseline method RMF for comparison since it does not incorporate any the neighborhood information.
Based on ratings and tag information, users are grouped into different classes. In MovieLens dataset, users are grouped into 13 classes: (5,10), (10,10), (15,20), (15,30), (20,20), (20,30), (25,30), (30,30), (30,40), (35,40), (50,50) and (> = 65, > = 100). (5,10) denotes that the users have 0
to 5 ratings and 0 to 10 tags. There are 48, 55 and 46 categories in Delicious, Last.fm and DBLP datasets, separately. Fig.7 summarizes the user group distributions of the testing data in different datasets. For clarity, the results are illustrated by two-dimensional graph and three-dimensional histogram in Fig.8 and Fig.9. Since RMSE is more indicative than MAE, we only present the performance in RMSE when \( f = 20 \).

![Graphs showing performance comparison on different user groups](image)

Fig. 8. Performance comparison on different user groups (2-D graph)
Fig. 9. Performance comparison on different user groups (3-D histogram)

From these results, the performance of two compared methods changed with different user groups. The comparison demonstrates that our model can perform better than the pure RMF approach in all user groups. Even when the user groups have few ratings and few tags, especially the group
with no more than 5 ratings or 10 tags, WUDiff_RMFi also improves prediction accuracy. Actually, when little ratings are observed, pure RMF methods can not work well by only relying on the sparse rating matrix. Thus, fusing the implicit information of neighbors based on the degree correlation of the tripartite network is a more efficient way to enhance accurate recommendations.

6. Conclusion and Future Work

In this paper, based on the intuition that the users’ ratings and tags have already reflected their interests, we present a hybrid recommendation model called WUDiff_RMFi. A distinct feature of our model obtains the implicit information of similar users by utilizing the degree correlation of the user-item-tag tripartite network, and expands the RMFi by integrating similar user regularization term. The experimental analysis of four datasets suggests that our proposed approach has better prediction accuracy over the existing seven methods, and the further analysis indicates it can alleviate the data sparsity, especially when active users have very few ratings and few tags.

There still exist several possible future extensions to our method. We will combine our model with other information of users’ behaviour, such as time information, social trust network information and the semantic information of tags. Moreover, aiming at working well with larger datasets, it is necessary to adopt some parallel computing methods to speed up the calculation.

Acknowledgements

The authors would like to thank Chenyi Zhang for his valuable datasets: DBLP and Movielens, and express their gratitude to Jing He for helpful suggestions. This work was partially supported by the following projects: National High-Technology Research and Development Program (863 Program) of China (Grant No. 2013AA01A212); National Natural Science Foundation of China (Grant No. 61370229, 61502180, 61272067); Natural Science Foundation of Guangdong Province of China (Grant No. S201203006242); Natural Science Foundation of Guangdong Province of China (Grant No. 2014A030310238, 2016A030313441); Science and Technology Program of Guangzhou, China (Grant No. 201508010067).
References

[1] L, L., Medo, M., Yeung, C.H., Zhang, Y.-C., Zhang, Z.-K., Zhou, T.: Recommender systems. Physics Reports 519(1), 1-49 (2012)

[2] Kantor, P.B., Rokach, L., Ricci, F., Shapira, B.: Recommender systems handbook. Springer, (2011)

[3] Chen J M, Tang F, Xiao J, et al. CogTime_RM: regularized matrix factorization with drifting cognition degree for collaborative filtering. Cluster Computing, 2016: 1-15.

[4] Bobadilla J, Ortega F, Hernando A, et al. Recommender systems survey. Knowledge-Based Systems, 2013, 46: 109-132.

[5] Zhang Z K, Zhou T, Zhang Y C. Tag-aware recommender systems: a state-of-the-art survey. Journal of computer science and technology, 2011, 26(5): 767-777.

[6] Wu L, Chen E, Liu Q, et al. Leveraging tagging for neighborhood-aware probabilistic matrix factorization. Proceedings of the 21st ACM international conference on Information and knowledge management. ACM, 2012: 1854-1858.

[7] Ma H. An experimental study on implicit social recommendation. Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2013: 73-82.

[8] Sen S, Vig J, Riedl J. Tagommenders: connecting users to items through tags. Proceedings of the 18th international conference on World wide web. ACM, 2009: 671-680.

[9] Wang Z, Wang Y, Wu H. Tags meet ratings: Improving collaborative filtering with tag-based neighborhood method. Proceedings of the Workshop on Social Recommender Systems. 2010: 15-23.

[10] Qi Q, Chen Z, Liu J, et al. Using inferred tag ratings to improve user-based collaborative filtering. Proceedings of the 27th Annual ACM Symposium on Applied Computing. ACM, 2012: 2008-2013.
[11] Gedikli F, Jannach D. Improving recommendation accuracy based on item-specific tag preferences. ACM Transactions on Intelligent Systems and Technology (TIST), 2013, 4(1): 11.

[12] Zhang Z K, Zhou T, Zhang Y C. Personalized recommendation via integrated diffusion on userCitemCtag tripartite graphs. Physica A: Statistical Mechanics and its Applications, 2010, 389(1): 179-186.

[13] Shang M S, Zhang Z K, Zhou T, et al. Collaborative filtering with diffusion-based similarity on tripartite graphs. Physica A: Statistical Mechanics and its Applications, 2010, 389(6): 1259-1264.

[14] Gan M X, Sun L, Jiang R. Trinity: Walking on a User-Object-Tag Heterogeneous Network for Personalised Recommendations. Journal of Computer Science and Technology, 2016, 31(3): 577-594.

[15] Paterek A. Improving regularized singular value decomposition for collaborative filtering. Proceedings of KDD cup and workshop. 2007, 2007: 5-8.

[16] Salakhutdinov R, Mnih A. Probabilistic matrix factorization. NIPS. 2011, 20: 1-8.

[17] Kim D, Yum B J. Collaborative filtering based on iterative principal component analysis. Expert Systems with Applications, 2005, 28(4): 823-830.

[18] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation. Journal of machine Learning research, 2003, 3(Jan): 993-1022.

[19] Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008: 426-434.

[20] Zhou T C, Ma H, King I, et al. Tagrec: Leveraging tagging wisdom for recommendation. Computational Science and Engineering, 2009. CSE’09. International Conference on. IEEE, 2009, 4: 194-199.

[21] Zhang C X, Zhang Z K, Yu L, et al. Information filtering via collaborative user clustering modeling. Physica A: Statistical Mechanics and its Applications, 2014, 396: 195-203.
[22] Chen C, Zheng X, Wang Y, et al. Capturing Semantic Correlation for Item Recommendation in Tagging Systems. Thirtieth AAAI Conference on Artificial Intelligence. 2016.

[23] Zhang C, Wang K, Lim E, et al. Are Features Equally Representative? A Feature-Centric Recommendation. Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015.

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

[25] Ekstrand M D, Riedl J T, Konstan J A. Collaborative filtering recommender systems. Foundations and Trends in Human-Computer Interaction, 2011, 4(2): 81-173.

[26] Cantador I, Bellogn A, Vallet D. Content-based recommendation in social tagging systems. Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010: 237-240.

[27] Wu H, Pei Y, Li B, et al. Item recommendation in collaborative tagging systems via heuristic data fusion[J]. Knowledge-Based Systems, 2015, 75: 124-140.

[28] I. Cantador, P. Brusilovsky, T. Kuflik, 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011), in: Proceedings of the 5th ACM conference on Recommender systems, RecSys 2011, ACM, New York, NY, USA, 2011.

[29] Jie Tang, A.C.M. Fong, Bo Wang, and Jing Zhang. A Unified Probabilistic Framework for Name Disambiguation in Digital Library. IEEE Transaction on Knowledge and Data Engineering (TKDE), Volume 24, Issue 6, 2012, Pages 975-987.

[30] Wang C, Blei D M. Collaborative topic modeling for recommending scientific articles. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011: 448-456.

[31] Rendle S. Factorization machines with libfm. ACM Transactions on Intelligent Systems and Technology (TIST), 2012, 3(3): 57.
[32] Agarwal D, Chen B C. Regression-based latent factor models. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009: 19-28.

[33] Gedikli F, Jannach D. Improving recommendation accuracy based on item-specific tag preferences. ACM Transactions on Intelligent Systems and Technology (TIST), 2013, 4(1): 11.

[34] Guo G, Zhang J, Yorke-Smith N. TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. AAAI Press, 2015: 123-129.