2014 IEEE International Conference on Data Mining

# Hete-CF: Social-Based Collaborative Filtering Recommendation using Heterogeneous Relations

Chen Luo
Jilin University, China
rackingroll@163.com

Wei Pang
University of Aberdeen, UK
pang.wei@abdn.ac.uk

Zhe Wang
Jilin University, China
wz2000@jlu.edu.cn

Chenghua Lin
University of Aberdeen, UK
chenghua.lin@abdn.ac.uk

***Abstract*—In this paper, we investigate the social-based recommendation algorithms on heterogeneous social networks and proposed Hete-CF, a social collaborative filtering algorithm using heterogeneous relations. Distinct from the exiting methods, Hete-CF can effectively utilise multiple types of relations in a heterogeneous social network. More importantly, Hete-CF is a general approach and can be used in arbitrary social networks, including event based social networks, location based social networks, and any other types of heterogeneous information networks associated with social information. The experimental results on a real-world dataset DBLP (a typical heterogeneous information network) demonstrate the effectiveness of our algorithm.**

## I. Introduction

With the advent of Internet era and the emergence of Big Data, users constantly suffer from information overload. Therefore, recommendation systems, as effective methods to deal with information overload, become a very popular research topic in recent years [1]. In practice, recommendation systems also help e-commerce companies provide personalised services. Many companies, including *YouTube* and *Amazon*, have launched their own personalised recommender systems so that better services can be provided.

Among many recommendation algorithms, collaborative filtering (CF) [1] has been widely used in both social networks and online stores. Most of the collaborative filtering methods aim to provide recommendations or rating predictions based on historical user-item preference records [1]. However, in the real world, users often only rate a limited number of items. For example, in *Amazon*, a user always buys a small fraction of all available commodities, which makes the corresponding user-item information matrix very sparse. Consequently, CF-based recommendation algorithms severely suffer from the cold start and data sparsity problems [1].

In order to deal with data sparsity, many algorithms have been proposed. Social-based recommendation, as one of the efficient and emerging methods, has attracted much attention in recent years [1–3]. Social-based recommendation utilises social information to help improve the recommendation performance. For example, Zhang *et al.* [3] consider the recommendation system on EBSN (Event Based Social Network) [2], which contains both online and offline networks. The algorithm presented in [3] uses the social information extracted from offline networks to help make recommendations in online networks. Another recommendation system research on EBSN also demonstrates the effectiveness of this method [4]. In [4], the author proposed LCARS, a location-content-aware recommender system which considers both personal interests and local preferences to make recommendations. In IJCAI-13, Yang *et al.* proposed the TrustMF recommendation algorithm [1]. TrustMF considers the trust and trustee information between users from the social network. Similar to TrustMF, Xiao Yu *et al.* proposed several recommendation algorithms [5] based on heterogeneous information networks (HINs) by introducing the relationships between items. In [6], the cross-domain knowledge was used to improve the recommendation performance. In this paper, we will introduce the above-related research in detail in Section V.

Previous research has demonstrated that more effective information could lead to better recommendation results [1, 2]. However, most of the above algorithms only use part of the information in social networks (either user-user or item-item information). In order to make better use of social information, in this work we study the collaborative filtering method on heterogeneous social networks. Different from previous research, in this paper we will utilise all three types of relations, that is, not only the user-user and item-item relations, but also the user-item relations.

As in [7], a heterogeneous social network (HSN) contains multi-typed relations and objects, and may contain more semantic meaning. In order to utilise these relations in a heterogeneous network, we can use meta-path [7] to represent each type of relations. Meta-path is an effective way of representing relationships in heterogeneous information networks (HINs) [7]. For example, in a bibliography network, the co-author relationship can be represented as a meta-path "Author−Paper−Author"; the co-conference relationship can be represented as a meta-path "Author−Paper−Conf.−Paper−Author". It is pointed out that the HSNs in our research represent all the HINs [7] with social information, for example, EBSNs (Event-based Social Networks) [2] and LBSNs (Location-based Social Networks) [8]. We will discuss the relations between these networks in Section II.

In this research, we first model the three types of relations (user-user, user-item, and item-item) individually and then propose a unified model. However, not all the social information is of benefit to the recommendation system. If the historical user-item ratings are not very sparse, the social information

The work described here was funded by the National Natural Science Foundation of China (NSFC) under Grant No. 61373051; the National Science and Technology Pillar Program (Grant No.2013BAH07F05), the Key Laboratory for Symbolic Computation and Knowledge Engineering, Ministry of Education, China, and the UK Economic & Social Research Council (ESRC); award reference: ES/M001628/1

1550-4786/14 $31.00 © 2014 IEEE
DOI 10.1109/ICDM.2014.64

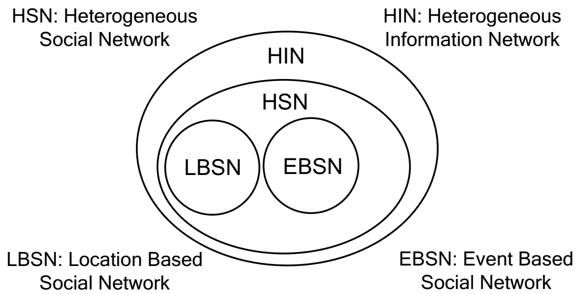

Fig. 1: Relationships between various types of Heterogeneous Networks.

may bias the recommendation results [6]. Therefore, we also proposed a leveraging method to evaluate the weight of the introduced social information. Our approach has been tested on a real-world dataset: DBLP, a typical heterogeneous information network with social information. The experimental results demonstrate the effectiveness of our algorithm.

## II. Preliminary

In this section, we introduce some background knowledge for our research, including HSNs and the measuring method meta-path [7].

As mentioned before, HSNs in our work can be regarded as HINs containing social information, for instance, bibliographic networks, or Facebook relationship network. The relations between HSN, HIN, EBSN and LBSN are illustrated in Fig. 1. From Fig. 1, we can see that LBSN and EBSN are special cases of HSN, and HSN is a special case of HIN. In this research, we consider all kinds of HSNs, including LBSN [8], EBSN [2] and any other types of HIN [7] associated with social information.

As HSN is a special case of HIN, we define HSN by following the concept of HIN. Referring to the definition of HIN [7], the HSN is defined as follows:

*Definition 1 (**Heterogeneous Social Network**):* Suppose we have $m$ types of data objects, denoted by $X_1 = \{x_{11}, ..., x_{1n_1}\}, ..., X_m = \{x_{m1}, ..., x_{mn_m}\}$, a heterogeneous social network is in the form of a graph $G = \langle O, E, W \rangle$, where $O = \bigcup_{i=1}^{m} X_i$ ($m \geq 2$), and in $O$ there exists at least one $X_i$ which is the type of social objects (e.g., Person, Author, or Actor). $E$ is the set of links between any two data objects in $O$. $W$ is the set of weight values on the links. Obviously, $G$ will reduce to a homogeneous network when $m = 1$.

In this paper, we use meta-path [7] to represent the relations in an HSN. As in [7], we use the topological measurement, *PathSim*, to measure the meta-path: given a meta-path, denoted as $P$, the *PathSim* between two objects $s$ and $t$ can be calculated as follows:

$$S_P^{PS}(s,t) = \frac{2 * S_P^{PC}(s,t)}{S_P^{PC}(s,:) + S_{P^{-1}}^{PC}(:,t)} \tag{1}$$

In the above, $S_P^{PC}(s,t)$ is a *Path Count* measure [7] and can be calculated as the number of path instances between $s$ and $t$. $P^{-1}$ denotes the reverse meta-path of $P$. $S_P^{PC}(s,:)$ denotes the path count value following $P$ and starting with $s$; and $S_P^{PC}(s,:)$ denotes the path count value following $P^{-1}$ ending with $t$.

## III. The Hete-CF Model

In this section, we first propose to combine the three types of relations (user-user, user-item, and item-item) in HSNs into a unified model. Then we introduce the learning method of our model. At last, we introduce the complete algorithm and analyse its parameter and time complexity.

### A. The Recommendation Model

*1) Modelling the relations between users:* as in [1], when we recommend items to users, the relationships between users can be used to improve the recommendation performance. This is because it is common sense that similar users are more likely to have similar orientations on a certain range of items. Let us consider an example of recommending conferences to authors for submitting their papers. Given two authors *Tom* and *Peter*, who focus on similar research topics and are both interested in artificial intelligence, if *Tom* frequently publishes his papers in the conference *AAAI*, *Peter* may have a high possibility to publish his paper in *AAAI* as well.

We model the above common sense by using graph regularization [9], and the objective function is shown as follows:

$$\min_{U,A} \sum_{k=0}^{N_A} \alpha_k \sum_{i=0}^{n} \sum_{j=0}^{n} S_A^k(i,j) \, \|U_i - U_j\|_F^2 \tag{2}$$

In the above, $\|*\|_F^2$ is the Frobenius norm [1]; $\alpha_k$ denotes the importance of the $k$-th meta-path between users and $A = [\alpha_1, \alpha_2, ..., \alpha_{N_A}]$; $N_A$ is the number of meta-paths between users. $U = [U_1, U_2, ..., U_n]$ denotes the low rank representation of users [1]. $S_A^k$ is the similarity matrix for users under the $k$-th meta-path relation, and it is calculated as follows (recall *PathSim* described in Section II): $S_A^k(i,j) = S_{P_k^A}^{PS}(i,j)$, where $P_k^A$ is the $k$-th meta-path between User $i$ and User $j$.

*2) Modelling the relations between items:* as in [5, 10], we can see that by introducing the relations between items the recommendation performance can be improved. It is common sense that a user may be interested in similar items. For example, if an author is interested in publishing his/her papers in *ICML*, this author may also be interested in publishing his/her papers in similar conferences, for instance, ***NIPS*** (***NIPS*** and ***ICML*** are both top conferences in the field of machine learning). Here we also employ graph regularization [9] to model this common sense as follows:

$$\min_{V,B} \sum_{k=0}^{N_B} \beta_k \sum_{i=0}^{m} \sum_{j=0}^{m} S_B^k(i,j) \, \|V_i - V_j\|_F^2 \tag{3}$$

In the above, $\beta_k$ denotes the importance of the $k$-th meta-path between items and $B = [\beta_1, \beta_2, ..., \beta_{N_B}]$; $N_B$ is the number of meta-paths between items; $V = [V_1, V_2, ..., V_m]$ denotes the low rank representation of items [1]. $S_B^k$ is the similarity matrix for items; $S_B^k(i,j)$ is calculated as: $S_B^k(i,j) = S_{P_k^B}^{PS}(i,j)$, where $P_k^B$ is the $k$-th meta-path between Item $i$ and Item $j$.

*3) Modelling the relations between users and items:* The collaborative filtering algorithm uses historical user-item ratings to make recommendations. In a heterogeneous social network, there are also many other types of relations between users and items, and these relations may be used to further improve the recommendation performance.

For instance, if an author often cites papers from a particular conference, he will also be very likely to submit his papers to that conference. The common sense here is that users may be highly interested in the items which are "close" to them. The term "close" here in the context of HSNs can be represented as the larger similarity of the meta-path between a user and an item, and this distance can again be calculated by using *PathSim*, as show in Equation (1). Same as the historical user-item ratings, these relationships are also between users and items. So we can use the collaborating filtering method to model such relationships, and the model is shown below:

$$\min_{U,V,W} \sum_{k=0}^{N_W} w_k \sum_{i=0}^{n} \sum_{j=0}^{m} (U_i^T V_j - \mathbb{R}_{i,j}^k)^2 \tag{4}$$

Here, $m$ and $n$ are the numbers of items and users, respectively; $w_k$ denotes the importance of the $k$-th meta-path between a user and an item, and $W = [w_1, w_2, ..., w_{N_W}]$; $N_W$ is the number of meta-paths between users and items. $\mathbb{R}^k$ is the relation graph for the $k$-th meta-path between users and items, and it can be calculated as: $\mathbb{R}^k(i,j) = S_{P_k^W}^{PS}(i,j)$, where $P_k^W$ is the $k$-th meta-path between User $i$ and Item $j$. $U$ and $V$ have the same meaning as in Equations (2) and (3).

*4) A Unified Model:* Having proposed the modelling approaches in Sections III-A1 $\sim$ III-A3, we intend to combine all the three factors together. And the unified model for recommendation in a HSN is proposed as follows:

$$\begin{aligned}\min_{U,V,A,B,W} & \sum_{i=0}^{m} \sum_{j=0}^{n} (U_i^T V_j - \mathbb{R}_{i,j})^2 \\ & + \sum_{k=0}^{N_A} \alpha_k \sum_{i=0}^{n} \sum_{j=0}^{n} S_A^k(i,j) \left\| U_i - U_j \right\|_F^2 \\ & + \sum_{k=0}^{N_B} \beta_k \sum_{i=0}^{m} \sum_{j=0}^{m} S_B^k(i,j) \left\| V_i - V_j \right\|_F^2 \\ & + \mu \sum_{k=0}^{N_W} w_k \sum_{i=0}^{n} \sum_{j=0}^{m} (U_i^T V_j - \mathbb{R}_{i,j}^k)^2 \\ & + \lambda(\left\| U \right\|_F^2 + \left\| V \right\|_F^2 + \left\| A \right\|_F^2 + \left\| B \right\|_F^2 + \left\| W \right\|_F^2)\end{aligned} \tag{5}$$

In the above, the symbols have the same meanings as introduced in previous sections. $\mu$ and $\lambda$ are important parameters which capture the importance of each term, and we will discuss these parameters in section III-C1.

The first term of the model incorporates the collaborating filtering component, which keeps the $U^T V$ closer to the *user-item* rating matrix $\mathbb{R}$. The second and third terms consider the *user-user* and *item-item* relations, respectively. The fourth term of the model is the *user-item* relationship component. The last term of the model is used for smoothing. After minimising this model, we can obtain $U$ and $V$, and then the predicted ratings can be obtained as $\hat{R_{ij}} = U_i^T V_j$.

In order to avoid over-fitting during the learning process, we introduce weighted-$\lambda$-regularization [11] in our algorithm. This method penalises the feature vectors which involve more ratings. Thus the last term of our model becomes:

$$\lambda(\sum_i n_{user_i} \left\| U \right\|_F^2 + \sum_j n_{item_j} \left\| V \right\|_F^2 + \left\| A \right\|_F^2 + \left\| B \right\|_F^2 + \left\| W \right\|_F^2),$$

where $n_{user_i}$ and $n_{item_i}$ denote the number of ratings given by User $i$ and the number of ratings given to Item $j$, respectively.

On the other hand, in a HSN the similarity values calculated by *PathSim* are between 0 and 1 (recall *PathSim* described in section II). So, as suggested in [12], in order to fit the data more conveniently, we adopt a logistic function to bound the inner product of the latent feature vectors into the interval $[0, 1]$. As in [12], we use the logistic function $f(x) = 1/(1 + \exp(-x))$ in our model. Then the model to be optimised is shown below:

$$\begin{aligned}\min_{U,V,A,B,W} & \sum_{i=0}^{m} \sum_{j=0}^{n} (f(U_i^T V_j) - \mathbb{R}_{i,j})^2 \\ & + \sum_{k=0}^{N_A} \alpha_k \sum_{i=0}^{n} \sum_{j=0}^{n} S_A^k(i,j) \left\| U_i - U_j \right\|_F^2 \\ & + \sum_{k=0}^{N_B} \beta_k \sum_{i=0}^{m} \sum_{j=0}^{m} S_B^k(i,j) \left\| V_i - V_j \right\|_F^2 \\ & + \mu \sum_{k=0}^{N_W} w_k \sum_{i=0}^{n} \sum_{j=0}^{m} (f(U_i^T V_j) - \mathbb{R}_{i,j}^k)^2 \\ & + \lambda(\sum_i n_{user_i} \left\| U \right\|_F^2 + \sum_i n_{item_i} \left\| V \right\|_F^2 \\ & + \left\| A \right\|_F^2 + \left\| B \right\|_F^2 + \left\| W \right\|_F^2)\end{aligned} \tag{6}$$

Thus the predicted rating becomes $\hat{R_{ij}} = f(U_i^T V_j)$. However, this model cannot be directly optimised. In order to optimise it, we rewrite the graph regularizing terms into their trace forms as in [5]. The graph regularizing terms are derived as follows:

$$\begin{aligned}& \sum_{k=0}^{N_A} \alpha_k \sum_{i=0}^{n} \sum_{j=0}^{n} S_A^k(i,j) \left\| U_i - U_j \right\|_F^2 \\ & \qquad = Tr(U^T (\sum_{k=0}^{N_A} \alpha_k L_A^k) U),\end{aligned} \tag{7}$$

and

$$\begin{aligned}& \sum_{k=0}^{N_B} \beta_k \sum_{i=0}^{m} \sum_{j=0}^{m} S_B^k(i,j) \left\| V_i - V_j \right\|_F^2 \\ & \qquad = Tr(V^T (\sum_{k=0}^{N_V} \beta_k \ L_B^k) V).\end{aligned} \tag{8}$$

In the above, $L_A^k = D_A^k - S_A^k$, where $L_A^k$ is a diagonal matrix and $D_A^k(i,i) = \sum_{j=0}^{n} S_A^k(i,j)$. Similarly, $L_B^k = D_B^k - S_B^k$, where $L_B^k$ is a diagonal matrix and $D_B^k(i,i) = \sum_{j=0}^{m} S_B^k(i,j)$. Finally, based on Equations (6), (7), and (8), the unified model, denoted as $J$, can be rewritten as:

$$\begin{aligned} J = &\sum_{i=0}^{m}\sum_{j=0}^{n}(f(U_i^T V_j) - \mathbb{R}_{i,j})^2 \\ &+Tr(U^T(\sum_{k=0}^{N_A}\alpha_k L_A^k)U) + Tr(V^T(\sum_{k=0}^{N_V}\beta_k\ L_B^k)V) \\ &+\mu\sum_{k=0}^{N_W} w_k \sum_{i=0}^{n}\sum_{j=0}^{m}(f(U_i^T V_j) - \mathbb{R}_{i,j}^k)^2 \\ &+\lambda(\sum_i n_{user_i}\|U\|_F^2 + \sum_i n_{item_i}\|V\|_F^2 \\ &+\|A\|_F^2 + \|B\|_F^2 + \|W\|_F^2) \end{aligned} \tag{9}$$

### *B. The Learning Algorithm*

In this subsection, we introduce the learning algorithm of our model presented in Equation (9). The learning method of our model is a two-step iteration one, where the predicted rating matrices $U, V$ and the weight for each meta-path $A, B, W$ mutually enhance each other. In the first step, we fix the weight vectors $A, B, W$ and learn the best predicted rating matrices $U, V$. In the second step, we fix the predicted rating matrices $U, V$ and learn the best weight vectors $A, B, W$.

*1) Optimise $U, V$ Given $A, B, W$:* When $A, B, W$ are fixed, the model becomes a traditional collaborative filtering model. Therefore, as in [1], we can use SGD (Stochastic Gradient Descent) to solve such problem.

*2) Optimise $A, B, W$ Given $U, V$:* When $U, V$ are fixed, the terms involving only $U, V$ can be discarded, and the objective function is reduced to:

$$\begin{aligned} J_1 = &Tr(U^T(\sum_{k=0}^{N_A}\alpha_k L_A^k)U) + Tr(V^T(\sum_{k=0}^{N_V}\beta_k\ L_B^k)V) \\ &+\mu\sum_{k=0}^{N_W} w_k \sum_{i=0}^{n}\sum_{j=0}^{m}(f(U_i^T V_j) - \mathbb{R}_{i,j}^k)^2 \\ &+\lambda(\|A\|_F^2 + \|B\|_F^2 + \|W\|_F^2) \end{aligned} \tag{10}$$

We can see that $J_1$ becomes a linear model for each $A, B$, and $W$. Therefore, we can also use SGD to obtain $A, B, W$.

### *C. The Complete Algorithm*

After presenting the calculation method for each relevant variable in Section III-B1 and Section III-B2, the detailed steps of our recommendation algorithm are given in Algorithm 1.

*1) Parameter Settings:* There are two parameters, $\lambda$ and $\mu$, in our model. As in [1, 11, 12], $\lambda$ is always assigned manually based on the experiments and experience. Therefore, we only discuss the assignment of parameter $\mu$.

$\mu$ is an important parameter, and it can directly affect the performance of our algorithm. When $\mathbb{R}$ is sparse, a larger $\mu$ can improve the recommendation results, because more information can be added to the training process. On the other hand, when $\mathbb{R}$ is not sparse, a larger $\mu$ will bias the recommendation results.

**Input**: A HSN $G = \langle O, E, W\rangle$. Three sets of meta-paths between user and item, users, and items; the user-item rating matrix $\mathbb{R}$; parameters $\mu$ and $\lambda$.
**Output**: The predicted rating Matrix $\hat{R}$;
Initialise $U, V, A, B, W$ randomly;
**while** *not reaching the inner $U, V, A, B, W$ difference threshold* **do**
    **while** *after updating the difference of $U, V$ is bigger than the predefined threshold* **do**
        Update $U, V$ using SGD;
    **end**
    **while** *after updating the difference of $A, B, W$ is bigger than the predefined threshold* **do**
        Update $A, B, W$ using SGD;
    **end**
**end**
The predicted rating is $\hat{R_{ij}} = f(U_i^T V_j)$;
return $\hat{R_{ij}}$.

**Algorithm 1:** Hete-CF

Therefore, the value of $\mu$ depends on how sparse $\mathbb{R}$ is. In this sense, we can use the proportion of non-zero elements in matrix $\mathbb{R}$ to calculate $\mu$ as follows:

$$\mu = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} I_{i,j}}{m \times n}, \tag{11}$$

where $I_{i,j}$ is calculated as:

$$I_{i,j} = \begin{cases} 0 & \text{if } \mathbb{R}_{i,j} = 0 \\ 1 & \text{if } \mathbb{R}_{i,j} \neq 0 \end{cases} \tag{12}$$

## IV. EXPERIMENTAL RESULTS

In this section, we first report the experimental evaluation of our algorithm by performing a set of experiments on a real dataset. Then, we study the parameter of our algorithm.

### *A. Datasets*

In this research, we use a real world HSN datasets for our experiments: DBLP [1]. As the links in DBLP are always sparse, we can test the ability of our algorithm on mitigating the cold start and data sparsity problems. On the other hand, our algorithm is proposed for HSNs, and DBLP is a typical HSN.

The DBLP dataset is widely used for heterogeneous network analysis [7, 10]. In this research, we extract a sub dataset from DBLP. The sub dataset contains the papers published in 261 computer journals and 313 computer conferences. The schema [7, 10] of DBLP is shown in Fig.2, in which Term is extracted from the titles of the papers. For the DBLP data, our recommendation problem becomes recommending conferences to authors. Therefore, we model the historical *user-item* relationship as the "Author−Paper−Conference" meta-path, and the rating that an author gives to a conference is calculated as the *PathSim* (see Section II) of the meta-path "Author−Paper−Conference".

As in previous research on DBLP as a HSN [7], we choose several meta-paths as follow: Author−Author: $A-P-A$, $A-$

[1]http://www.informatik.uni-trier.de/∼ley/db/

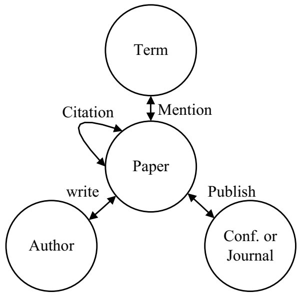


Fig. 2: Network Schema

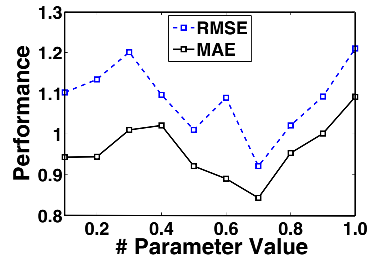


Fig. 3: Parameter investigation on the DBLP $\mu$ ranges from 0.1 to 1.0. The value of $\mu$ calculated by our method is: $\mu = 0.7$ in DBLP

$P-C-P-A$, $A-P-T-P-A$; Conf. − Conf.: $C-P-A-P-C$, $C-P-P-C$, $C-P-T-P-C$; Author − Conf: $A-P-T-P-C$, $A-P-P-C$. In the above, $A$ stands for "Author"; $P$ stands for "Paper"; $C$ means "Conference or Journal"; and $T$ is for "Term".

### *B. Experimental Setup*

As in [1], we use 5-fold cross-validation for learning and testing. We randomly select $40\%$ $(60\%)$ of the data as the training set and the rest $60\%$ $(40\%)$ as the testing set. Each result discussed below is averaged over ten trials.

There are three baselines and two state-of-the-art methods used in our methods for comparison. The three baselines are listed as follows: (1)**UserMean:** prediction rate equals the mean value of the users. (2)**ItemMean:** prediction rate equals the mean value of the items. (3)**NMF:** non-negative matrix factorization function [5], with $d = 5$, and $d = 10$, where $d$ denotes the dimension of the feature vector. The three baselines cannot make use of the heterogeneous relations in a HSN, and they only consider the target recommendation relation (the $A-P-C$ relation in DBLP). So in this experiment, we only consider the target recommendation relations in each dataset for these three baseline algorithms.

The two state-of-the-art methods selected in our experiment are described below: (1)**Trust-MF[1]**: this is described in Section I. In the DBLP dataset used in our experiment, in terms of the relations between users we use the co-author relationship. As in [1], the parameters of this algorithm is set as follows: $\lambda = 0.001$ and $\lambda_t = 1$. (2)**Hete-MF[5]**: Hete-MF considers the relations between 'items'. In this paper, for the relations between 'items', we add all the selected relations: $C-P-P-C$, $C-P-A-P-C$, $C-P-T-P-C$ in DBLP. Finally, as in [1, 5], MAE and RMSE are used as the evaluation methods in our experiment.

### *C. Result and Analysis*

The experimental results are reported in Table I. From these results, we can see that the performance of our Hete-CF algorithm was improved with the increase of the amount of training data. This is because more training data can provide more information, and more importantly, our model can avoid over-fitting. On the other hand, we can see that when the dimension of feature vector $d = 5$, the performance is always better. This is because our datasets extract nearly five areas of data from the DBLP dataset (see Section IV-A). Compared with the other algorithms, Hete-CF always performs better, except that when d=5 with 60% training data, trust-MF performed slightly better than Hete-CF. From this aspect, our algorithm has significant advantages when performing the recommendation task on HSNs.

### *D. Parameter Study*

In this section, we study the impact of the parameter $\mu$ in Equation (9), which is used for leveraging the importance of the introduced meta-paths between users and items in our model. The result is shown in Fig. 3.

In Fig. 3, we can see that when the value of $\mu$ is too small or too large, the result will not be good enough. Because $\mu$ is used for leveraging the *user-item* meta-path term (the fourth term) in our model shown in Equation (9), an extreme value of $\mu$ (too large or too small) will bias the result. On the other hand, the $\mu$ value calculated by Equation (11) is $\mu = 0.7$ in DBLP and the best result just appeared when $\mu = 0.7$ as shown in Fig. 3. This demonstrates that our proposed method of evaluating the $\mu$ value is effective, and it can deal with the parameter pre-assignment issue.

## V. Related Work

In this section, we introduce some work related to our research.

**Collaborative Filtering and Social-based Recommendation** Social based recommendation is an emerging research topic which combines the recommendation algorithms and social media mining algorithms. In [3], the author proposed a group recommendation method on EBSN. This method considers location features, social features, and implicit patterns in a unified model. In [13], the authors proposed to realise location recommendation services. The recommendation method in [13] considered both the friend relationship and the geographic information. The algorithm proposed in [6] utilised the knowledge from other domains to improve the recommendation performance. In [10], the authors proposed a recommendation algorithm on HINs and utilise part of the heterogeneous relations. In this research, we aim to bridge the gap between CF-based recommendation and social-based recommendation. Different from the above algorithms, our proposed method considers the recommendation in HSNs (e.g., LBSN, EBSN and other HINs with social information). In addition, our method utilises all types of relations in HSNs while the above algorithms utilise only part of the relations.

TABLE I: Algorithm Performance Comparison in DBLP

| %Training | Feature | Evaluation | UserMean | ItemMean | NMF | Trust-MF | Hete-MF | Hete-CF |
|---|---|---|---|---|---|---|---|---|
| 40% | $d = 5$ | MAE | $0.942 \pm 0.02$ | $1.065 \pm 0.02$ | $2.156 \pm 0.02$ | $0.831 \pm 0.01$ | $0.931 \pm 0.02$ | $\mathbf{0.831 \pm 0.02}$ |
| | | RMSE | $1.216 \pm 0.01$ | $1.123 \pm 0.02$ | $2.394 \pm 0.01$ | $1.013 \pm 0.02$ | $1.105 \pm 0.01$ | $\mathbf{1.002 \pm 0.03}$ |
| | $d = 10$ | MAE | $0.943 \pm 0.03$ | $0.948 \pm 0.01$ | $2.194 \pm 0.03$ | $0.887 \pm 0.01$ | $0.901 \pm 0.01$ | $\mathbf{0.859 \pm 0.01}$ |
| | | RMSE | $1.138 \pm 0.02$ | $1.256 \pm 0.04$ | $2.292 \pm 0.02$ | $1.083 \pm 0.03$ | $1.114 \pm 0.03$ | $\mathbf{1.056 \pm 0.02}$ |
| 60% | $d = 5$ | MAE | $0.948 \pm 0.02$ | $0.919 \pm 0.01$ | $2.131 \pm 0.04$ | $\mathbf{0.812 \pm 0.02}$ | $0.891 \pm 0.02$ | $0.831 \pm 0.02$ |
| | | RMSE | $1.132 \pm 0.02$ | $1.157 \pm 0.01$ | $2.385 \pm 0.01$ | $\mathbf{0.907 \pm 0.02}$ | $1.010 \pm 0.03$ | $0.938 \pm 0.02$ |
| | $d = 10$ | MAE | $0.932 \pm 0.03$ | $0.978 \pm 0.03$ | $2.184 \pm 0.02$ | $0.873 \pm 0.03$ | $0.881 \pm 0.01$ | $\mathbf{0.856 \pm 0.02}$ |
| | | RMSE | $1.154 \pm 0.02$ | $1.143 \pm 0.02$ | $2.275 \pm 0.01$ | $1.051 \pm 0.01$ | $1.013 \pm 0.02$ | $\mathbf{0.994 \pm 0.03}$ |

**Mining HSNs** As introduced before, HSNs are special cases of HINs. HIN has multi-typed objects and relations, and it may contain more meaningful information. The concept of HIN is first proposed by Sun *et. al* in [7, 14]. The work carried out by them demonstrated that by mining HINs, one can obtain more meaningful results. It also attracted us to carry out further research on mining HINs [15, 16]. In this research, we consider all types of HINs together, and our recommendation method can be used in all types of networks introduced above.

## VI. Conclusion and Future Work

In this paper, we focus on the recommendation problem using heterogeneous social relations, and we proposed Hete-CF, a collaborative filtering recommendation method on HSNs. Different from the previous social based recommendation methods, we propose to effectively incorporate all the social relations, including the relations between users, items and user-item. In addition, since Hete-CF is a network structure based model, it can be used in many types of social networks (e.g. , LBSN, EBSN, and other HINs with social information). For instance, Hete-CF can be used in recommending off-line events in EBSN or recommending locations (or hotels) in LBSN. The experiments on a real-world HSN DBLP demonstrate the effectiveness of Hete-CF [2].

In the future, we intend to apply Hete-CF to more real-world recommendation problems. In addition, another direction of our future research is to explore the potential of Hete-CF on big data problems, such as problems involving massive amounts social media data.

## References


[1] B. Yang, Y. Lei, D. Liu, and J. Liu, “Social collaborative filtering by trust,” in *IJCAI’13*. AAAI Press, 2013, pp. 2747–2753.

[2] X. Liu, Q. He, Y. Tian, W.-C. Lee, J. McPherson, and J. Han, “Event-based social networks: linking the online and offline social worlds,” in *KDD’12*. ACM, 2012, pp. 1032–1040.

[3] W. Zhang, J. Wang, and W. Feng, “Combining latent factor model with location features for event-based group recommendation,” in *KDD’13*. ACM, 2013, pp. 910–918.

[4] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen, “LCARS: a location-content-aware recommender system,” in *KDD’13*. ACM, 2013, pp. 221–229.

[5] X. Yu, X. Ren, Q. Gu, Y. Sun, and J. Han, “Collaborative filtering with entity similarity regularization in heterogeneous information networks,” *IJCAI HINA*, 2013.

[6] S. Gao, H. Luo, D. Chen, S. Li, P. Gallinari, and J. Guo, “Cross-domain recommendation via cluster-level latent factor model,” in *MLKDD*. Springer, 2013, pp. 161–176.

[7] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, “Pathsim: Meta path-based top-k similarity search in heterogeneous information networks,” *VLDB’11*, vol. 4, no. 11, pp. 992–1003, 2011.

[8] Y. Zheng, Y. Chen, X. Xie, and W.-Y. Ma, “Geolife2. 0: a location-based social networking service,” in *MDM’09*. IEEE, 2009, pp. 357–358.

[9] M. Wu and B. Schölkopf, “Transductive classification via local learning regularization,” in *ICAIS’07*, 2007, pp. 628–635.

[10] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, Q. Gu, B. Norick, and J. Han, “Recommendation in heterogeneous information networks with implicit user feedback,” in *RecSys’13*. ACM, 2013, pp. 347–350.

[11] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, “Large-scale parallel collaborative filtering for the netflix prize,” in *Algorithmic Aspects in Information and Management*. Springer, 2008, pp. 337–348.

[12] A. Salakhutdinov Ruslan, Mnih, “Probabilistic matrix factorization.” in *NIPS*, 2007, pp. 1257–1264.

[13] B. J. et. al, “Location-based and preference-aware recommendation using sparse geo-social networking data,” in *GIS’12*. ACM, 2012, pp. 199–208.

[14] Y. Sun, Y. Yu, and J. Han, “Ranking-based clustering of heterogeneous information networks with star network schema,” in *KDD’09*. ACM, 2009, pp. 797–806.

[15] C. Luo, W. Pang, and Z. Wang, “Semi-supervised clustering on heterogeneous information networks,” in *PAKDD’14*. Springer, 2014, pp. 548–559.

[16] C. Luo, R. Guan, Z. Wang, and C. Lin, “Hetpathmine: A novel transductive classification algorithm on heterogeneous information networks,” in *ECIR’14*. Springer, 2014, pp. 210–221.


[2] The data and code used in this research can be accessed through: https://github.com/rackingroll/Hete-CF