2012 Second International Conference on Cloud and Green Computing

# Book Recommendation Based On Joint Multi-Relational Model

Qiuzi Shangguan
Department of Computer Science and Engineering
Shanghai Jiao Tong University
Shanghai, China
autumn371525@gmail.com

Liang Hu
Department of Computer Science and Engineering
Shanghai Jiao Tong University
Shanghai, China
lianghu@sjtu.edu.cn

Jian Cao
Department of Computer Science and Engineering
Shanghai Jiao Tong University
Shanghai, China
cao-jian@cs.sjtu.edu.cn

Guandong Xu
Center for Applied Informatics
Victoria University
Australia
guandong.xu@vu.edu.au

***Abstract*—Recommender system, which is powerful to deal with the issue of information overload, has been widely investigated by many researchers recently. However, one of the biggest challenges needs to face is the cold start problem. To address this problem, the data source from social network is incorporated into our recommender system in this paper. In a social network, users who tightly connected imply some group-specific interests. Consequently, we may exploit social network information to resolve the cold start problem and improve prediction performance. The main motivation of this paper is to exploit social relationships and other extra data sources to adjust the latent factors learning over the target matrix, namely book rating matrix and a group of auxiliary matrices, typically, the social relationship matrix. Our recommender system is based on coupled matrix factorization in major, and utilizes the random walk and genetic algorithm to learn some special parameters. The data for experiments is crawled from one of the Chinese biggest reading-sharing website, Douban. Finally, the results have proved that our book recommender system incorporating auxiliary data sources has much better performance than traditional methods.**

***Keywords-recommendation; social network; coupled matrix factorization; mult-relational model; genetic algorithm***

## I. INTRODUCTION

With the development of computer technology and extension of the Internet, information overload is becoming a severe problem because it cannot gain target information efficiently in a traditional way such as retrieving data from relational databases. Recommender systems, which are designed to deal with information overload, have been extended from E-commerce to information systems for recommending books, webpages and other media, such as music, movies, and applications. Recommender system is a meaningful tool to help users filter and retrieve target information on the Internet.

At the same time, applications based on social network develop quickly in recent years. A social network system provides the information of interactions among individuals. Recently, social network sites have become popular platforms to retrieve and develop social connections. Since the influence of social network can be subtle but profound, a lot of sociological researches indicate that people in the same circle of a social network will influence each other and finally share certain common interests. To merge the information of social network into recommender system so as to figure out users' tastes and tendency is the starting point of our research.

Traditional recommendation algorithms only analyze one kind of data, namely rating history, so joint multi-relational model is proposed to analyze multiple sources of data, especially social network information. In our model, coupled matrix factorization techniques are applied to obtain latent factors from collective matrices; random walk with restart algorithm is used to explore the strength of associations in social network and genetic algorithm is applied to adjust the impacts from all matrices so as to achieve better results for recommendation. In our experiments, the data are crawled from one of the biggest Chinese reading-sharing website, Douban, and the results demonstrate that our social network information enhanced book recommender system has better performance than traditional approach.

The paper is organized as follows. Section II presented related work. Section III formulated the problem. Section IV introduced our model. The algorithm was presented in Section V and genetic optimization was explained in Section VI. Section VII illustrated the experiment results. Section VIII concluded the whole paper and provided some discussion.

## II. RELATED WORK

Current recommendation algorithms can be divided into two categories: content filtering and collaborative filtering.

Content filtering is based on the similarity of content. Each book can be represented as a content vector: every element in vector stands for a subject term. Each user's interest can also be represented as a content vector. After the calculation of similarity between documents and users'

978-0-7695-4864-7/12 $26.00 © 2012 IEEE
DOI 10.1109/CGC.2012.53

interests, recommendation can be made based on the similarity score [1, 2]. Content filtering is easy to implement. However, the practical experiment shows that content filtering algorithm always recommends the books which users have already read; content filtering cannot recommend new kinds of documents to users.

Collaborative filtering is applied into recommendation system more popularly. Collaborative filtering is based on the assumption that similar users make similar ratings for similar items. Collaborative filtering can be classified into three kinds: user-based, item-based and model-based [3]. User-based collaborative filtering first searches the nearest neighbors and then calculates the recommendation according to the neighbors' rating and similarity. Item-based collaborative filtering algorithm first searches the most similar items and then calculates the predicting score according to the target user's rating history of the similar items [3]. Model-based algorithm is to set up a learning model, which can be trained by data set [4, 5].

All the algorithms above need to face the same problem — cold start. It always needs to impose additional information to solve the cold start problem. For instance, trust-based recommendation [6] mixes a trust network as additional information, where the preferences of cold start users are biased to their trusted users. SoRec [7] proposed the assumption that users are not independent and not identically distributed; it mines relation between users and items and the experiment proved that it had much better performance than compared methods.

Matrix Factorization (MF) is one of the most successful methods to recover missing values. Sometimes, we need to model multiple relations by a group of matrices in which some matrices may share some common dimensions. To handle such correlations between data, Singh and Gordon [13] proposed a framework to factorize collective matrices simultaneously.

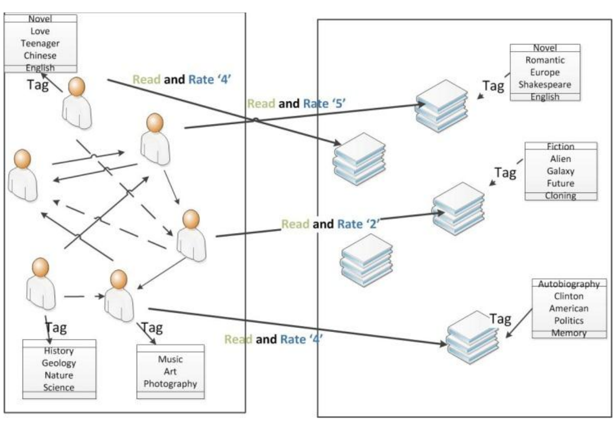


Figure 1. Overview of Multi-Relations between Readers and Books.

## III. PROBLEM FORMULATION

Adding extra social network information and book information into the model can solve the cold start problem and improve the recommendation performance at the same time. Figure 1 shows the multi-type social connections between books and readers. Readers have a social network in which they follow friends' latest moves. Readers may have read several books and made a rating for each finished book. Moreover, readers can label books with any tags, so every reader will have a tag collection which records all tags the reader have labeled; similarly, every book also has a tag collection which records all the tags it has received. Such a mixed model demonstrates clearer relations between a pair of objects, which leads to a better prediction result than a sole type of relation [7, 8].

### A. *Main Relation*

The target relation is reader's ratings assigned for books. The rating history is the most explicit data that recommender system can rely on. We denote the readers as $r = \{r_1, r_2, \dots, r_{N_r}\}$, where $N_r$=$|r|$. The books reader $r_i$ has read can be represented as $b_{r_i}$=$\{b_{r_i,1}, b_{r_i,2}, \dots, b_{r_i,N_p}\}$, where $N_p$=$|b_{r_i}|$. The rating reader $r_i$ has made for books can be denoted by $s_{r_i}$=$\{s_{r_i,1}, s_{r_i,2}, \dots, s_{r_i,N_p}\}$.

### B. *Social Relations*

Every reader has his contacts in the social network. Reader may follow contacts' move or be influenced by contacts' reading history. So social relation is an important relation we can make use of to improve recommendation performance. Reader $r_i$'s social relation can be represented as $c_{r_i}$=$\{c_{r_i,1}, c_{r_i,2}, \dots, c_{r_i,N_f}\}$, the collection contains all the contacts reader $r_i$ follows with. How to use social information to mine deeper relation will be explained in next section.

### C. *Tag Relations*

Besides rating history and social network information, tag history is also an important data source because tags can reflect readers' interests and books' characters. Readers' tags can be represented by two vectors: $t_{r_i} = \{t_{r_i,1}, t_{r_i,2}, \dots, t_{r_i,N_h}\}$, $tc_{r_i} = \{tc_{r_i,1}, tc_{r_i,2}, \dots, tc_{r_i,N_h}\}$ where the vector $t_{r_i}$ contains all the tags reader $r_i$ has labeled and the vector $tc_{r_i}$ records the corresponding times every tag has been used. Books' tags can also be denoted in the same way: $w_{b_i} = \left\{t_{b_i,1}, t_{b_i,2}, \dots, t_{b_i,N_m}\right\}$ and $wc_{b_i} = \{tc_{b_i,1}, tc_{b_i,2}, \dots, tc_{b_i,N_m}\}$, where vector $w_{b_i}$ and $wc_{b_i}$ records all the tags and labeled times book $b_i$ have received.

### D. *Ranking Unread Books*

For each reader, all the books can be categorized into two types: read ones and unread ones. Our goal is to pick up the right books which meet readers' tastes in unread ones. In Figure 2, we have the explicit rating scores for the read books. After the model's calculation, all the unread books will be assigned a predicting score for each user. According to the predicting rating score, we can rank the unread books and choose the top-N as the recommendation result.

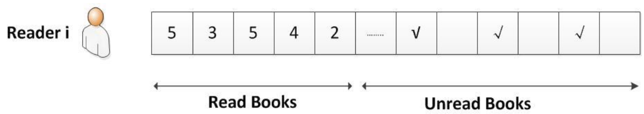


Figure 2. Ranking Unread Books.

## IV. JOINT MULTI-RELATIONAL (JMR) MODEL

In Section III, we have listed all the data sets and their formal representations. In this section, the joint multi-relational model will be introduced. The model consists of five matrices: rating matrix, contact matrix, common contact matrix, book-tag matrix and reader-tag matrix.

First, we will discuss how the data sets are arranged into five matrices and how the five matrices are associated and coupled together. Then, the core algorithm about how the five matrices are factorized and how the missing values are reconstructed will be explained in detail.

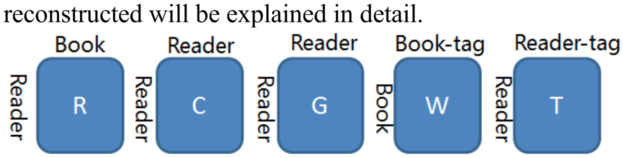


Figure 3. Overview of five matrices.

### A. The Formalization of Matrices

- The rating matrix consists of two dimensions, namely *Reader* and *Book*. Each reader is associated with a unique index $i$. Similarly, each book is also given a unique index $j$. Now, we can build the rating matrix according to $s_{r_i}$. In the rating matrix, the element $m_{ij}$ represents the score reader $r_i$ has rated for the book $b_j$.
- For the social relations, there are two data matrices: contact matrix and common contact matrix. Contact matrix is built directly from the readers' contact vector $c_{r_i}$={$c_{r_{i,1}}, c_{r_{i,2}}, \dots, c_{r_{i,N_f}}$}. The element $m_{ij}$ represent that reader $r_i$ has added user $r_j$ as his contact. In the social network, it is possible that two readers have no direct relation, but they share a big overlapped circle. The common circle may influence their interests because of the circle's same information output. So the amount of common contacts is also a kind of important social information to the recommender system. After comparing each pair of readers' contacts, we can construct the common contact matrix: element $m_{ij}$ represent the amount of common contacts reader $r_i$ and reader $r_j$ sharing.
- Tag relations are formalized as book-tag matrix and reader-tag matrix. The first step is to pick up the top-N (about 10000) most popular tags and associate them with unique indices. Then, we can build the book-tag matrix, where each element $m_{ij}$ stands for the amount that book $b_i$ has been labeled by the $i^{th}$ tag. Likewise, reader-tag matrix is constructed as follows, where each element $m_{ij}$ stands for the amount that reader $r_i$ labeled books with $j^{th}$ tag.

### B. The Integration of Matrices

The above subsection introduces the construction of the five matrices. As shown in Figure 3, rating matrix **R**, contact matrix **C**, common contact matrix **G**, book-tag matrix **W** and reader-tag matrix **T** consist of the original data the model will process. To improve the prediction and handle the cold start problem, we can integrate above introduced five matrices together so as to propagate information among coupled matrices. We can observe from Figure 3 that the matrices in social relations and tag relations all share one dimension with the rating matrix, so it is straightforward to couple these five matrices over the common dimensions as given in Figure 4, that is, coupling auxiliary matrices **C**, **G**, **T** with rating matrix **R** over the *Reader* dimension and coupling auxiliary matrix **W** with rating matrix **R** over the *Book* dimension.

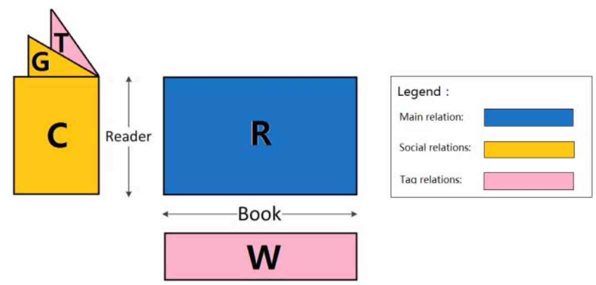


Figure 4. The coupling relations among matrices.

### C. Random Walk with Restart

In the social relations, the social network information is significant for joint multi-relational model. Contact matrix **C** simply represents the relation of every pair of reader. However, we can view the social network as a bipartite graph. A good method to measure the relevance between two nodes on a weighted graph is provided by random walk with restart (RWR) [9, 10]. So we can use RWR algorithm to refine the entries of contact matrix **C**.

Random walk is a mathematical formalization of a trajectory that consists of taking successive random steps with a certain possibility. The migration can reaches steady state under some specific Markov chain condition [9, 10]. The result $p_{ij}$ represents the possibilities from node $i$ to node $j$

$$\sum_j p_{ij} = 1 \tag{1}$$

Given a graph G with N nodes, we can represent G with an $N \times N$ adjacent matrix $\boldsymbol{An}$. We denote $\boldsymbol{s_i}$ as the row vector to measure the relevance score, where the element $s_i(j)$ denotes the relevance score of node $j$ w.r.t node $i$. RWR algorithm can be defined as in Eq. (2):

$$\boldsymbol{s_i} = (1-c)\boldsymbol{P}\boldsymbol{s_i} + c\boldsymbol{v_i} \tag{2}$$

where P is the normalized weighted transition matrix associated with $\boldsymbol{An}$ (e.g. column normalization). $c$ is the probability of restarting the random walk from node $i$ and $\boldsymbol{v_i}$ is a $1 \times N$ starting vector, where the $i^{th}$ element is 1 and 0 for others. The function of restart is to constrain the range of random walk because the very small probability is ignorable. We denote the procedure to compute the steady state of $\boldsymbol{s_i}$ from Eq. (2) [9, 10] as a function:

$$\boldsymbol{s_i} = f_{RWR}(\boldsymbol{An}, i, c) \tag{3}$$

Now, we can set the value of each element of contact matrix **C** by RWR. After calculation, each pair of readers (entry of **C)** will be assigned a score to represent the strength of their relationships in the social network.

## V. ALGORITHM

So far we have built a joint multi-relational model where all the data are arranged into five matrices and the contact matrix **C** has been refined by RWR algorithm. How to factorize the five matrices and predict desired books for each reader will be solved in this section.

Typically, a relation modeled by a matrix $\boldsymbol{U}$ can be factorized into two low-rank latent factor matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ [11], that is, $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{V}^T$ where $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$ (latent factor vectors) are the rows of $\boldsymbol{U}, \boldsymbol{V}$ and $d$ is the dimensionality of the latent factor space. In this section, a graphical model based on above constructed matrices is presented and a factorization method is also provided to find factor matrices so as to make prediction.

### A. Graphical Model for Joint Relation

With the rating matrix **R**, social relation matrices **C** and **G**, user-tag matrix **T** and book-tag matrix **W** in hand, the graphical model for these coupled matrices can be given by Figure 5. **R** can be factorized into $F_r$ and $F_i$ for dimensions *Reader* and *Book*. Likewise, the factor matrices of **W** for dimensions books and tags are represented by $F_r$ and $F_w$; $F_i$ and $F_t$ are the factor matrices of **T** for dimensions *Reader* and *Tag*; $F_i$ and $F_g$ are the factor matrices of **G** for dimensions *Reader* and *Contact*; $F_i$ and $F_c$ are the factor matrices of **C** for dimensions *Reader* and *Contact*. It is easy to see **W** shares the common factor matrix $F_r$ with **R**, i.e. $F_r$ serves as the latent factor matrix for both **R** and **W**, so that the tag information can influence the preference for the books. Similarly, **T**, **G**, **C** share the common factor matrix $F_i$ with **R**. Since the shared latent factor matrices are learned through fitting multiple coupled matrices together, so we should carefully to set the weights (here are denoted as $\alpha$ in Figure 5) to scale the loss of fitting each matrix. Well-chosen weights can significantly improve the prediction results [7, 13]. The predicted complete rating matrix $\widetilde{\boldsymbol{R}}$ can be approximated by $F_r F_i^T$, where each row of the reader factor matrix $F_r$ is the reader's latent factor vector which can be interpreted as the amount of reader $r_i$'s personalized interests over the latent factor of some book.

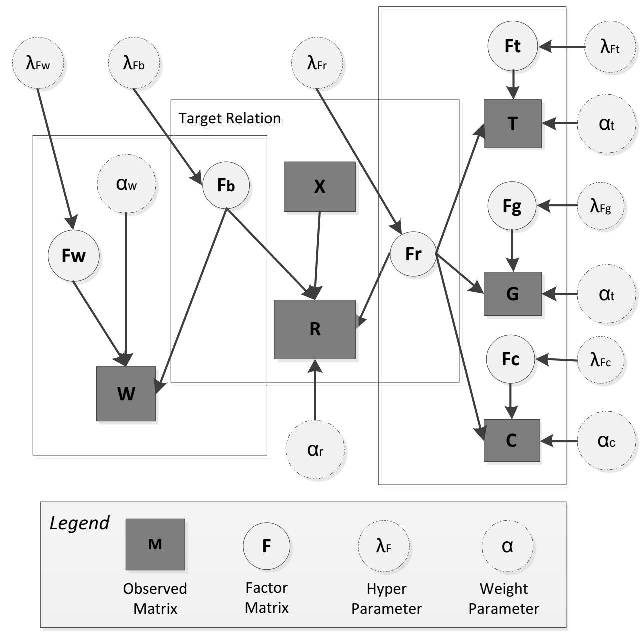


Figure 5. Graphical model for joint relations.

### B. Joint Relational Matrices Factorization

According to the graphical model, it is easy for us to translate it into a loss function for optimization [7, 12] as follows:

$$f = L_M + L_S + L_B + \Omega \tag{4}$$

where $f$ is composed of three parts of losses on all relations and regularization terms to prevent from overfitting [13]. The objective function consists of the losses $L_M$, $L_S$, $L_B$ and the regularization term $\Omega$, where $\|\cdot\|$ denotes the Frobenius norms.

$$L_M = \frac{\alpha_r}{2} \left\| X * (R - F_r F_i^T) \right\|^2 \tag{5}$$

$$L_S = \frac{\alpha_t}{2} \left\| T - F_r F_t^T \right\|^2 + \frac{\alpha_g}{2} \left\| G - F_r F_g^T \right\|^2 + \frac{\alpha_c}{2} \left\| C - F_r F_c^T \right\|^2 \tag{6}$$

$$L_B = \frac{\alpha_w}{2} \left\| (W - F_b F_w^T) \right\|^2 \tag{7}$$

$$\Omega = \sum_{F \in (F_r, F_b, F_w, F_t, F_g, F_c)} \frac{\lambda_F}{2} \|F\|^2 \tag{8}$$

$L_M$ models the loss on fitting the rating matrix, where $X$ denotes the mask matrix in which the entries **0** represent unread ones and **1** stand for read and rated ones. The operator $*$ denotes the element-wise product.

$L_S$ and $L_B$ (Eq. 6 & 7) are the loss functions for social and tag relations. $F_i$ has been introduced as the linear concatenated of three auxiliary latent factor matrices and the weights of the three matrices are determined by the parameter $\eta_i$.

Simply minimizing a loss may lead to overfitting, so the function $\Omega$ is added for regularizing all factor matrices. $\lambda_F$ in Eq. (8) are used to control the penalty of each factor matrix.

In order to minimizing the object function, we need to compute the gradient w.r.t each factor matrix and then we can use any first-order optimization algorithm. The partial derivatives of $\Omega$ are easy to be derived with respect to each factor matrix as follows.

$$\frac{\partial \Omega}{\partial F} = \lambda_F F$$

The partial derivatives of $L_M$ w.r.t. $F_r$ and $F_b$ are given as follows:

$$\frac{\partial L_M}{\partial F_r} = [X * (F_r F_b^T - R)]F_b$$

$$\frac{\partial L_M}{\partial F_b} = [X * (F_b F_r^T - R)]F_r$$

More details of the derivation can be found in [14, 15]. The partial derivatives w.r.t other factor matrices $F_w, F_t, F_g, F_c$ are zeros.

Similarly, the partial derivatives of $L_S$ and $L_B$ are given by:

$$\frac{\partial L_S}{\partial F_t} = \alpha_t \left(F_t F_r^T - T\right)F_r$$

$$\frac{\partial L_S}{\partial F_r} = \alpha_t (F_r F_t^T - T)F_t$$

$$\frac{\partial L_S}{\partial F_g} = \alpha_g (F_g F_r^T - T)F_r$$

$$\frac{\partial L_S}{\partial F_r} = \alpha_g (F_r F_g^T - T)F_g$$

$$\frac{\partial L_S}{\partial F_c} = \alpha_c (F_c F_r^T - T)F_r$$

$$\frac{\partial L_S}{\partial F_r} = \alpha_c (F_r F_g^T - T)F_g$$

$$\frac{\partial L_B}{\partial F_b} = \alpha_w (F_b F_w^T - W)F_w$$

$$\frac{\partial L_B}{\partial F_w} = \alpha_w (F_w F_b^T - W)F_b$$

From Eq. (4), we can easily write the partial derivatives of $f$ w.r.t $F \in \left\{F_r, F_b, F_w, F_t, F_g, F_c\right\}$

$$\frac{\partial f}{\partial F} = \frac{\partial L_M}{\partial F} + \frac{\partial L_S}{\partial F} + \frac{\partial L_B}{\partial F} + \frac{\partial \Omega}{\partial F}$$

The gradient of $f$ can be written by vectorizing $\frac{\partial f}{\partial F}$ and then we can obtain Eq. (9)

$$\nabla f = \left[\mathrm{vec}(\tfrac{\partial f}{\partial F})\right]_{F \in \{F_r, F_b, F_w, F_t, F_g, F_c\}} \quad (9)$$

Now, we have built the objective function and derived its gradient, we can use on kind of gradient-based optimization algorithm such as the Nonlinear Conjugate Gradient (NCG) [17] or Limited-Memory BFGS method (L-BFGS) to compute the factor matrices. As shown in Algorithm 1, we use an iterative algorithm to approximate the factor matrices.

**Algorithm 1: Factorization by Iterative Gradient Descent**

1: Initialize $\{F_r, F_b, F_w, F_t, F_g, F_c\}$ randomly
2: While not converged
3: Compute the objective function by Eq. (4);
4: Compute the gradients $\nabla f$ by Eq. (9);
5: Update $\{F_r, F_b, F_w, F_t, F_g, F_c\}$ by gradient descent algorithm given $f$ and $\nabla f$;
6: End While
7: Output $\{F_r, F_b, F_w, F_t, F_g, F_c\}$;

### C. Recommendation by Factor Matrices

Through algorithm 1, we can obtain all the low rank latent factor matrices. In order to infer the potential preferences for unread books for readers, we can construct the missing data straightforward from the latent factor matrices $F_i$ and $F_r$ of rating matrix **R**.

$$\widetilde{\boldsymbol{R}} = F_i F_r^T \quad (10)$$

Given a reader $r_i$, the recommendation ranks for the unread books can be given by sorting the reconstructed values in a descending order.

$$rec(r_i) = \widetilde{\boldsymbol{R}}(r_i) \downarrow \quad (11)$$

## VI. Learning Weights by Genetic Algorithm

In last section, we have derived the algorithm to learn latent factor matrices and briefly discussed the importance of setting proper weights on the losses of fitting matrices. In fact, each auxiliary matrix should impose different degree of influence on learning the shared factor matrix with target matrix, which can be controlled by the weight. With the number of weights needed to be tuned increasing, manually tuning all the weights becomes infeasible. Therefore, in this section we employ the genetic algorithm to automatically tune the weights so as to achieve better performance.

### A. Genetic Algorithm

Genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. The general procedure of GA can be given as following:

*1)* Choose the initial population of individuals

*2)* Evaluate the fitness of each indicidual in that population

*3)* Repeat on this generation until termination (time limit, sufficient fitness achieved, etc):

*a)* Select the best-fit indicidual s for reproduction *b)* Breed new individuals through crossover and mutation operation to give birth to offspring

*c)* Evaluate the individual fitness of new individuals

*d)* Replace least-fit population with new individuals

### *B. Area under The ROC*

Area under the ROC curve (AUC): AUC measures the probability that a system ranks a positive instance higher than a negative one [18]. This metric has been widely used to evaluate the result of link prediction [19] and recommendation [20].

The AUC of the recommended papers for a reader $r_i$ is defined as follows:

$$AUC(u) = \frac{\sum_{i \in pa(r)} \sum_{k \in u \backslash pa(r)} \delta(rk(i) < rk(k))}{|pa(r)||u \backslash pa(r)|} \qquad (12)$$

where $rk(i)$ is a function to retrieve the rank of book *i.* $\delta(rk(i) < rk(k))$ is the delta function which returns 1 if $rk(i) < rk(k)$ and 0 otherwise. $pa(r)$ is the validation data set and the set $u$ refers to the unread books.

In this section and following experiments, we use AUC as the metric to evaluate the recommendation performance.

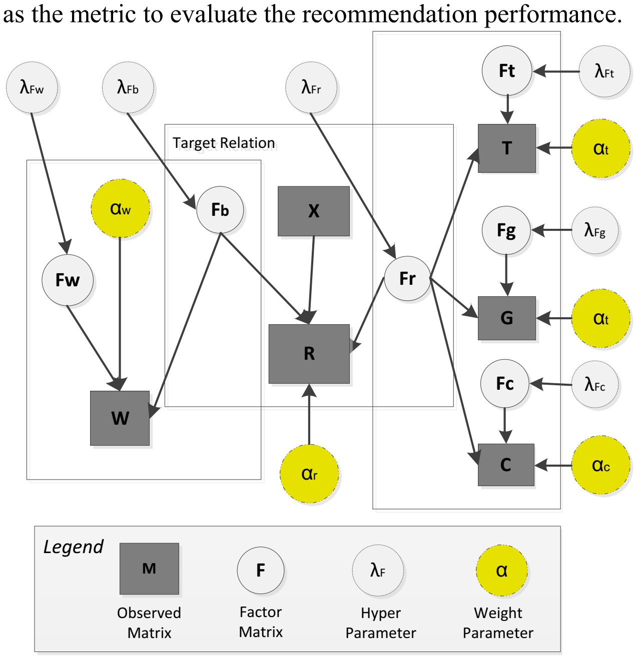


Figure 6. Weights imposed on losses need to be learned

### *C. Modify The Joint Multi-Relational Model*

As shown in Figure 6, the highlighted circles in the graphical model are the weights that need to apply genetic algorithm for tuning. Each weight is used to control the penalty of the approximated relation diverging from the true relation. For example, the larger is $\alpha_w$, the more divergence is scaled between $F_b {F_w}^T$ and the data matrix **W** which lead to increasing the loss of objective function. So we should make more effort to minimize $L_B$, i.e. the shared factor matrix $F_r$ is more determined by the relation **W**. In contrary, a small $\alpha_w$ suppress the loss of fitting **W**, that is, $F_r$ may be more determined by the rating matrix **R** with some relative larger $\alpha_r$. Similarly, $\alpha_g, \alpha_t, \alpha_c$ control influence coming from matrices **T**, **G** and **C**.

The genetic algorithm is a method to obtain approximate optimal solution. The result will be nearer to the best solution with more generations' calculation. Before applying our genetic algorithm, we stack the weight parameters as a vector $\vec{p}$ given by Eq. 12:

$$\vec{p} = \begin{bmatrix} \alpha_r \\ \alpha_t \\ \alpha_g \\ \alpha_c \\ \alpha_w \end{bmatrix} \qquad (12)$$

Figure 7 shows the procedures of our GA based optimization. First, we initialize a start group of vector $\vec{p}$, and then evaluate the recommendation results using the above model and algorithm for each weight vector. After selection, crossover, mutation, we can obtain the offspring. The procedure repeat until the termination condition reaches. Here, we choose the $100^{th}$ generation's result as the optimal weight vector.

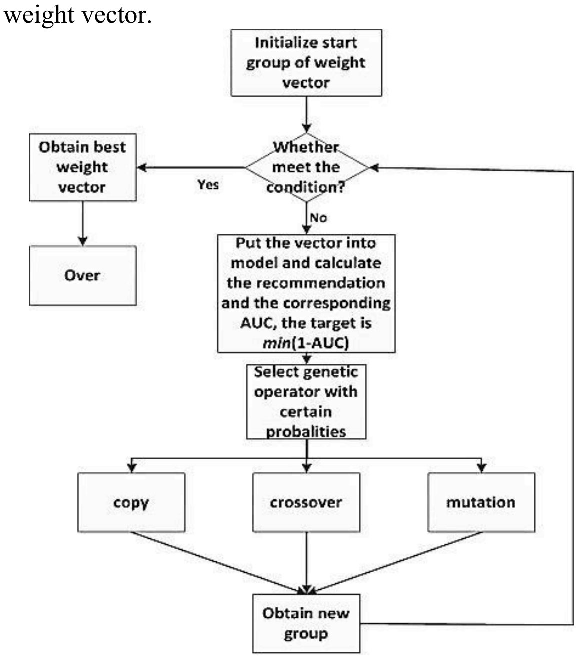


Figure 7. Procedure of genetic algorithm

A heuristic set configuration, such as the equal weights $\{0.2, 0.2, 0.2, 0.2, 0.2\}$, may degrade the recommendation performance whereas the genetic optimization of weight vector makes the model and algorithm more robust and automatic. In addition, we can analyze the influence from each matrix according to the weight assigned through the genetic optimization.

## VII. EXPERIMENT

Because the model and the algorithm are based on reading books and social network, we collect data from Douban Reading website [21] in China as the experiment data. Douban is one of the biggest reading and sharing website in China. Users can record their reading history, label tags to books, assign ratings to books and write reading reviews. Moreover, users can become contacts or friends with others and set up activity groups, which means Douban also provides a platform to readers to manage their social circle.

### A. Data Preparation

Douban provides APIs for the development of related applications. With such APIs, we can easily obtain users' data where all the data is organized in JSON format. To analyze the JSON data, we use Jackson processor.

In this experiment, we randomly retrieve 1200 readers' profile and the related 31080 books. Apart from the users' ratings for each book, the retrieved dataset contains the tags information associated with each book and contacts for each user. This data set was processed into matrices and applied into the model as discussed in previous sections.

### B. Result on Joint Multi-Relational Model

Following the iterative algorithm and genetic optimization described above, we obtain the performance of the AUC 0.6619 average over all readers. And the AUC for each reader is shown in Figure 8.

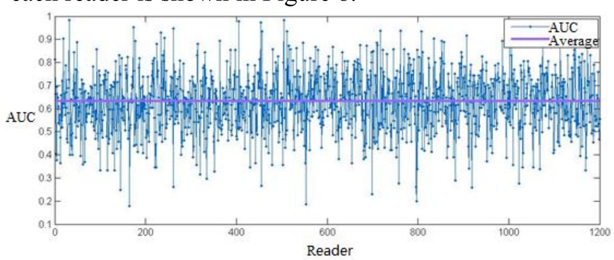


Figure 8. Result of JME Model

Because of the special application, we cannot simply judge whether the performance of our model is good. So, controlled experiments are conducted to compare the performance with other models.

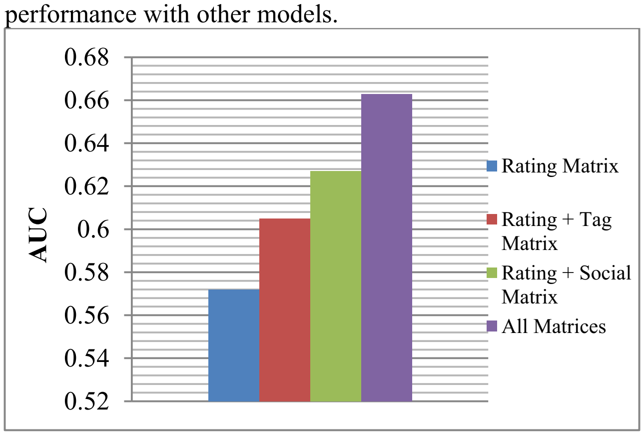


Figure 9. AUC of comparative models

The following comparative trials use different models supplied with different data sources to compare with main model (the five coupled matrices model shown in Figure 4). The case of single rating matrix based model is the most frequently studied in many recent works [11], so we select it as the base line. Rating matrix + Social matrix and Rating matrix + Tag matrix are two compared models with partial data sources based on our approach. Our main model is the five matrices coupled model that is constructed with all the data sources.

From Figure 9, we can find that the baseline model, namely single rating matrix, achieve the worst performance in four compared model. Rating matrix + Social matrix and Rating matrix + Tag matrix achieve higher AUC than the baseline model which proves that appropriately imposing auxiliary information do imporve prediction performance. Our main model is the best one in four, which indicates that the incorporation of social network information and tag information can siginificantly improve the performance comparing with the baseline model.

### C. Genetic Optimization

Since the performance is highly dependent on the weight for fitting each matrix, we need to test whether our genetic algorithm can automatically tune the weight vector to achieve better performance.

Figure 10 illustrates the comparison of AUC results from 1$^{st}$ generation to 200$^{th}$ generation, from which we can conclude that more generations of optimization leads to higher AUC. We use 100 generation as the default stop condition. In order to verify the convergence within 100 generations, we continue the iterations until the 200$^{th}$ generation. From Figure 10, we can find that the 200$^{th}$ generation's result is almost same with the 100$^{th}$ generation.

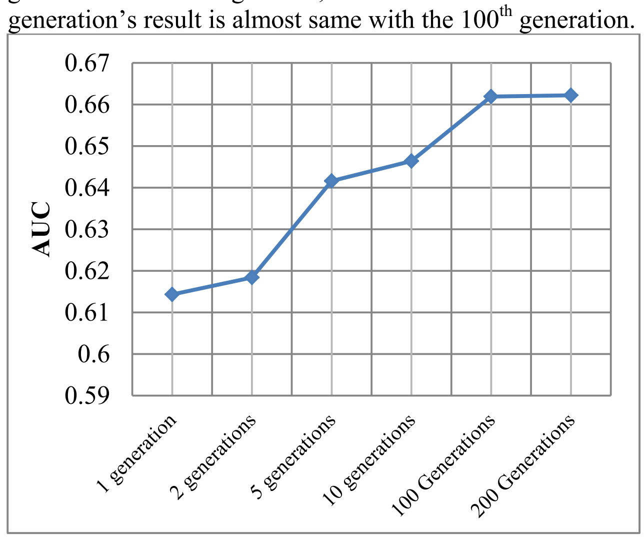


Figure 10. AUC over increasing generations

We tracked the changes of weight vector during the optimization process from the first generation to the 100$^{th}$ generation to demonstrate how genetic algorithm adjusts weights. Table I depicts the 1$^{st}$, 2$^{nd}$, 5$^{th}$, 10$^{th}$ and 100$^{th}$ weights used to fit each matrix. From the final results (the 100$^{th}$ generation), we can find that the weights assigned to fit matrix **C** and **G**, (the matrices modeling the social networks), is much larger than other weights. That is, the social network information plays a very importance role in rating and recommending books.

TABLE I. WEIGHTS OF DIFFERENT GENERATIONS

| | Matrix | | | | |
|---|---|---|---|---|---|
| **Generation** | **R** | **C** | **G** | **T** | **W** |
| 1 | 0.107 | 0.454 | 0.137 | 0.159 | 0.143 |
| 2 | 0.257 | 0.086 | 0.316 | 0.209 | 0.132 |
| 5 | 0.272 | 0.367 | 0.29 | 0.021 | 0.05 |
| 10 | 0.239 | 0.343 | 0.356 | 0.025 | 0.037 |
| 100 | 0.166 | **0.317** | **0.456** | 0.023 | 0.038 |

From above experiments, we can conclude that our multi-relational book recommender system which incorporates social networks and tag information can achieve better performance than the rating matrix only model.

## VIII. CONLUSION

In this paper, we propose a multi-relational model which incorporates social network information with the user ratings to improve recommendation and deal with cold start problem. Such auxiliary information is very helpful to predict the unobserved values in the main relations. Correspondingly, we devise a coupled matrix factorization algorithm to learn latent factors for each matrix. Finally, the experiments were conducted using the data crawled from Douban reading website. The results of comparative trials both have proved that our joint multi-relational model achieves much better recommendation performance than currently popular single matrix based model.

## ACKNOWLEDGMENT

The authors appreciate the anonymous reviewers for their extensive and informative comments for the improvement of this paper. Thanks for the support from our laboratory and Douban platform.

## REFERENCES

[1] Marko,B. and Yoav,S., Fab: content-based, collaborative recommendation. Communications of the ACM, Volume 40 Issue 3, March 1997, 66-72.

[2] Basu,C.,Hirsh,H., and Cohen, W., Recommendation as classifacation: Using social and content-based information in recommendation. In Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98), 714-720.

[3] Sarwar, B., Karypis, G., Konstan, J., Riedl, J. Item-based collaborative filtering recommendation algorithms. In proceeding of the 10th international conference on Worlk Wide Web (New York, USA, 2001)

[4] Sarwar, B., Karypis, G., Konstan, J., Riedl, J. Application of Dimensionality Reduction in Recommender System. 2000

[5] Tang, T.Y and McCalla, G. 2003. Smart recommendations for an evolving e-learning system. In Workshop on Technologies for Electronic Documents for Supporting Learning. AIED' 2003

[6] Jamali, M. and Ester, M. TrustWalker: a random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (Paris, France, 2009).

[7] Ma, H., Yang, H., Lyu, M. R. and King, I. SoRec: social recommendation using probabilistic matrix factorization. In *Proceeding of the 17th ACM conference on Information and knowledge management* (Napa Valley, California, USA, 2008).

[8] Dunlavy, D. M., Kolda, T. G. and Kegelmeyer, W. P. *Multilinear algebra for analyzing data with multiple linkages*. SIAM, Philadelphia, 2011.

[9] Pan, J.-Y., Yang, H.-J., Faloutsos, C. and Duygulu, P. Automatic multimedia cross-modal correlation discovery. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (Seattle, WA, USA, 2004).

[10] Tong, H., Faloutsos, C. and Pan, J. Y. Random walk with restart: fast solutions and applications. *Knowledge and Information Systems*, 14, 3 2008), 327-346.

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

[12] Salakhutdinov, R. and Mnih, A. Probabilistic matrix factorization. In *Advances in neural information processing systems* (2008).

[13] Singh, A. P. and Gordon, G. J. Relational learning via collective matrix factorization. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (Las Vegas, Nevada, USA, 2008).

[14] Acar, E., Kolda, T. G. and Dunlavy, D. M. All-at-once Optimization for Coupled Matrix and Tensor Factorizations. In *Proceedings of Mining and Learning with Graphs* (2011).

[15] Hu, Y., Koren, Y. and Volinsky, C. *Collaborative filtering for implicit feedback datasets*. IEEE, City, 2008.

[16] Acar, E., Dunlavy, D. M., Kolda, T. G. and Morup, M. Scalable tensor factorizations with missing data. In *Proceedings of the 2010 SIAM International Conference on Data Mining* (Philadelphia, 2010).

[17] Nocedal, J. and Wright, S. J. *Numerical optimization*. Springer verlag, 1999.

[18] Fawcett, T. An introduction to ROC analysis. *Pattern recognition letters*, 27, 8 2006), 861-874.

[19] Backstrom, L. and Leskovec, J. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining* (Hong Kong, China, 2011).

[20] Rendle, S., Freudenthaler, C. and Schmidt-Thieme, L. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web* (Raleigh, North Carolina, USA, 2010).

[21] Douban, http://www.douban.com/