A Two-Phase Spectral Bigraph Co-clustering Approach for the "Who Rated What" Task in KDD Cup 2007

Ting Liu
The Institute of Computing Technology, Chinese Academy of Sciences
Beijing 100080, China
tliu@jdl.ac.cn

Yonghong Tian[1]
The Institute of Digital Media, School of EE&CS, Peking University
Beijing 100871, China
yhtian@jdl.ac.cn

Wen Gao[2]
The Institute of Digital Media, School of EE&CS, Peking University
Beijing 100871, China
wgao@jdl.ac.cn

ABSTRACT
This paper describes our approach for the "Who Rated What" task in KDD Cup 2007 competition. Given the Netflix data set that consists of more than 100 million ratings between 1998 and 2005, this task is to predict the probability that each user-movie pair was rated in 2006. Totally 100,000 user-movie pairs are drawn from the Netflix data set as the test set. In our approach, the Netflix data set is modeled as a bipartite graph (or bigraph) with users and movies on either side. In the bigraph, there are only directed edges from user nodes to movie nodes and each directed edge corresponds to a rating event that the user rated the movie at some time. Then the given task can be further formulated as a link existence prediction problem, i.e., whether a directed link exists between a user node and a movie node. Considering the huge size and the sparsity of ratings in the data set, it is important to reveal the hidden class-based correlation between users and movies from the bigraph while keeping relatively low computational complexity. Towards this end, a two-phase spectral bigraph co-clustering approach is used in our approach. The key idea is to simultaneously obtain user and movie neighborhoods via co-clustering and then generate predictions based on the results of co-clustering. Roughly speaking, our approach includes three steps. First, users and movies are coarsely clustered using K-means algorithm respectively. Then the user and movie clusters are further co-clustered using multipartite spectral graph partition algorithm. Based on the results of co-clustering, a probabilistic model is derived to predict the probability of a link existing between a user node and a movie node. Experimental results show that our approach works well in the task.

Categories and Subject Descriptors
H.2.8 Database Applications: *Data mining*

General Terms
Algorithms, Design, Experimentation, Performance

[1] Yonghong Tian is the corresponding author, and he acted as the major coach of this KDD Cup team.
[2] Wen Gao is another coach of the team.

KDDCup'07, August 12, 2007, San Jose, California, USA.

Keywords
Two-Phase Spectral Bigraph Co-clustering, Collaborative Filtering, Link Existence Prediction

1. INTRODUCTION
This paper introduces our two-phase spectral bipartite graph co-clustering approach to the "Who Rated What" task in KDD Cup 2007 competition. Given the Netflix data set that consists of more than 100 million ratings between 1998 and 2005, this task is to predict the probability that each user-movie pair was rated in 2006. Totally 100,000 user-movie pairs are drawn from the Netflix data set as the test set. The detailed description of this task can be found in the KDD Cup 2007 web site. In this section, we highlight several challenging characteristics of the task and some related works, which motivated our approach to the task.

The main challenges of this task includes the lack of user and movie attributes, the huge size (more than 100 million rating records) and the sparsity of ratings in the data set (Over 98% of the possible pairs were not rated). Thus the algorithms designed for this task need to reveal the hidden class-based correlation between users and movies while keeping relatively low computational complexity.

1.1 Related work
A similar problem called rating-based collaborative filtering has been studied for years. This problem is to predict the exact score (generally ranged from 1 to 5) that a user will give for a movie. Latent semantic models are the most favorable model to solve this problem [1], [15]. However, slightly different from this problem, the "Who Rated What" task is to predict the existence of a rating rather than the rating score. Thus we cannot directly apply the rating-based collaborative filtering models to the given task.

Another kind of possible solutions can be derived from link prediction problem for relational data [13]. Link prediction has been an important problem in network modeling and has recently been studied in social network, genetic interaction network, and literature citation network contexts [6], [12], [14], [17]. In these studies certain linkage measures are defined to infer the potential for a future link to appear. Various models have been developed to solve the link prediction problem, such as Probabilistic Relational Models (PRMs)[11], Relational Markov Networks (RMNs) [14], relational regression models [12], latent space models [16], [17], or other supervised learning algorithms [6]. In many cases, the links are time-varying [8]. Thus the task becomes a temporal link prediction problem [2], [3]: Given a snapshot of the set of links before time t, the goal is to predict the links at time t+1. To solve

the problem, Madadhain *et al.* [18] propose an EventRank algorithm, and Sarkar and Moore [19] use a dynamic latent model.

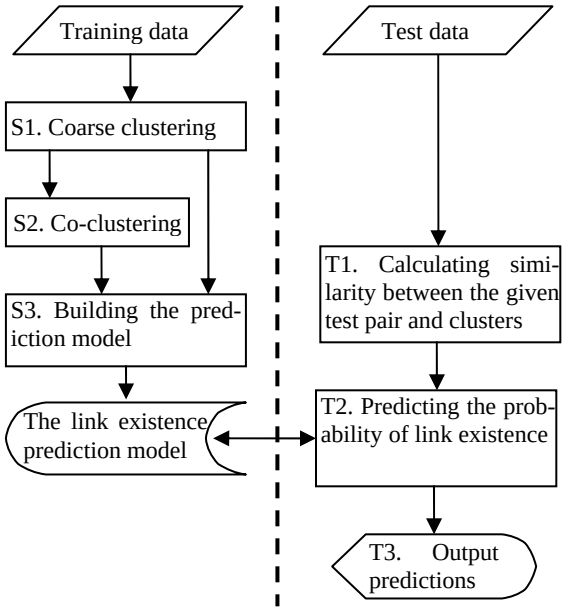

Figure 1. The overall design of our approach. The lines with arrows represent the work flow as well as the flow of data.

1.2 Motivation of our approach

Motivated by the success of link prediction models for relational data, we represent the rating data as links in a bipartite graph containing user and movie nodes and then further formulate the "Who Rated What" task as a link existence prediction problem, i.e., whether a link exists between a user node and a movie node. Particularly,

"A bipartite graph, also called a bigraph, is a set of graph vertices decomposed into two disjoint sets such that no two graph vertices within the same set are adjacent."[3]

Under this bigraph representation, we develop a two-phase spectral bigraph co-clustering approach to solve the given task. The key idea is to simultaneously obtain user and movie neighborhoods via co-clustering and then generate predictions based on the results of co-clustering. Informally, cluster analysis seeks to partition a given data set into compact clusters so that data objects within a cluster are more similar than those in distinct clusters. In general, most clustering algorithms focus on one-way clustering, i.e., cluster one dimension of the table based on similarities along the second dimension where the data is often arranged as a two dimensional table such as user-movie rating table. However, when dealing with sparse and high-dimensional data, it turns out to be beneficial to employ co-clustering or simultaneously clustering both dimensions of the table by exploiting the clear duality between rows and columns [7].

Within our bigraph model, the co-clustering problem can be solved by constructing vertex graph partitions. Finding a globally optimal solution to such a graph partitioning problem is NP-complete; however, the second left and right singular vectors of a suitably normalized user-movie matrix gives an optimal solution to the real relaxation of this discrete optimization problem [4]. More recently, a bipartite isoperimetric graph partitioning approach [10] is also proposed to solve this NP-complete graph partition problem.

Due to the sparsity of ratings in the data set, it is necessary to reveal the hidden class-based correlation between users and movies from the bigraph. Towards this end, the co-clustering algorithms mentioned above apply singular value decomposition (SVD) on the user-movie matrix which turns out to be too large for the matrix calculation in this task. Therefore, akin to [5], users and movies in our model are coarsely clustered respectively to reduce data dimensions before the co-clustering phase.

The overall design of our approach is shown in Figure 1. Roughly speaking, our approach includes three steps (S1~S3 in Fig. 1). First, users and movies are coarsely clustered using K-means algorithm respectively. Then the user and movie clusters are further co-clustered using multipartite spectral graph partition algorithm. Based on the results of co-clustering, a probabilistic model is derived to predict the link existence probabilities. Experimental results show that our approach works well in the "Who Rated What" task.

The paper is organized as follows: Section 2 introduces the two-phase spectral bigraph co-clustering algorithm, and Section 3 presents our link existence prediction model. In Section 4, we demonstrate our experimental results. Finally, Section 5 concludes the paper.

2. TWO-PHASE SPECTRAL BIGRAPH CO-CLUSTERING

In this section, we present our two-phase spectral bigraph co-clustering approach. As mentioned before, the objective of co-clustering is to simultaneously obtain user and movie neighborhoods so that predictions can be generated based on the results of co-clustering. To reduce data dimensions in the co-clustering process, a coarse clustering step is firstly performed on movies and users. Figure 2 illustrates this two-phase clustering process.

To describe our approach, we begin with some notations. In our solution, the Netflix data set is modeled as a bigraph $\mathcal{G} = (\mathcal{U}, \mathcal{F}, \mathcal{L})$ containing user nodes $\mathcal{U}$ and movie nodes $\mathcal{F}$. A link $u \to f \in \mathcal{L}$ denotes a rating event of a movie $f \in \mathcal{F}$ (where $M = |\mathcal{F}|$) by a user $u \in \mathcal{U}$ (where $N = |\mathcal{U}|$). Without loss of generality, we use a function $T(u, f, t)$ to denote whether a link exists between a user node *u* and a movie node f at time *t*, which is 1 if the user *u* rates the movie *f* at that time and 0 otherwise.

2.1 Coarse clustering

The link structure of the bigraph $\mathcal{G}$ provides a wealth of information for revelation of correlations between users and movies. In this phase, users and movies are classified into coarse clusters to provide approximate but useful information for those who have only several link neighbors, e.g., a user who only rated less than 10 movies. At the same time, the clustering algorithm must also preserve the links and the time information of the bigraph.

[3] From http://mathworld.wolfram.com/BipartiteGraph.html

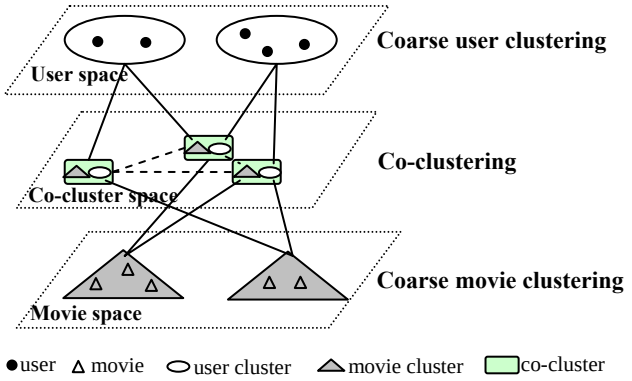

Figure 2. The illustration of two-phase clustering. Users are coarsely clustered in the top layer, and movies are coarsely clustered in the bottom layer. Then both coarse cluster sets are co-clustered in the middle layer.

As provided by Hasan [6], there are some link features that can be extracted from a bigraph for every node with light computation complexity. Similarly, some link features are also used in our approach, e.g., sum of neighbors and clustering index.

In practice, we take snapshots of the graph $\mathcal{G}$ every year $t_1,\ldots,t_s$ from the start time t_0, and calculate link numbers for every node in each sub-graph $\mathcal{G}_{(t)}$. So both users and movies have feature vectors of the link sums of every year. The *i*'th feature for a user $\mathbf{f}_i^u$ and for a movie $\mathbf{f}_i^f$ can be calculated as:

$$\begin{aligned}\mathbf{f}_i^u &= \sum\nolimits_{f\in\mathcal{F}} T(u,f,t_i)\\ \mathbf{f}_i^f &= \sum\nolimits_{u\in\mathcal{U}} T(u,f,t_i)\end{aligned} \qquad (1)$$

We use the K-means algorithm to cluster users into N_c clusters, and movies into M_c clusters. Thus each user is projected into one user cluster $g(u)=c_i^{(\mathrm{U})}$ $(1\le i\le N_c)$, and each movie into one movie cluster $g(f)=c_i^{(\mathrm{M})}$ $(1\le i\le M_c)$. How to choose parameter *K* is presented in the experimental section. Here the key point is to use large numbers so that the clustering results may preserve the diversity of the graph.

2.2 Co-clustering the coarse clusters

In the co-clustering phase, we use all links without any discrimination of the time. So the $N\times M$ time-varying graph $\mathcal{G}$ is then shifted to a $N_c\times M_c$ static graph $\mathcal{G}^{(\mathrm{C})}$. Similarly, we also use a function $T_c\left(c_i^{(\mathrm{U})},c_j^{(\mathrm{M})}\right)$ to denote how many links exist between a user cluster $c_i^{(\mathrm{U})}$ and a movie cluster $c_j^{(\mathrm{M})}$:

$$T_c\left(c_i^{(\mathrm{U})},c_j^{(\mathrm{M})}\right)=\sum\nolimits_t\sum\nolimits_{g(u)=c_i^{(\mathrm{U})},u\in\mathcal{U}}\sum\nolimits_{g(f)=c_j^{(\mathrm{M})},f\in\mathcal{F}} T(u,f,t). \qquad (2)$$

Clearly, after the coarse clustering phase, the graph $\mathcal{G}^{(\mathrm{C})}$ is still a bigraph with user clusters and movie clusters as the two parts. To further reveal the correlations between user clusters and movie clusters, a co-clustering step is then performed on the results of the coarse clustering phase. As mentioned before, the co-clustering problem can often be solved by constructing vertex bigraph partitions. Ideally we want to get global optimum partition of the graph $\mathcal{G}^{(\mathrm{C})}$, which has minimum sum of links between partitions and maximum sum of links within partitions.

One possible strategy to partition a bigraph is to iteratively cluster the two parts of the bigraph on each other. But this two-way iterative clustering can only get local optimum solution. So we use the spectral graph partitioning algorithm which co-clusters users and movies simultaneously to get a global optimum solution.

To do so, we firstly construct a Laplacian matrix $\mathbf{\Gamma}=\lfloor\tau_{ij}\rfloor$ for the bigraph $\mathcal{G}^{(\mathrm{C})}$. Generally speaking, the Laplacian matrix of a graph is an $n\times n$ symmetric matrix, with one row and column for each vertex, such that $\tau_{ij}=\sum_k\omega_{ik}$ if *i=j*, and $\tau_{ij}=-\omega_{ij}$ otherwise. ω_{ik} is the edge weight between the node *i* and the node *j*, and is 0 if the edge is not existent. In this bipartite case, the Laplacian matrix is defined as

$$\mathbf{\Gamma}=\begin{bmatrix}\mathbf{D}_1 & -\mathbf{T}_c\\ -\mathbf{T}_c' & \mathbf{D}_2\end{bmatrix},\quad \mathbf{D}=\begin{bmatrix}\mathbf{D}_1 & 0\\ 0 & \mathbf{D}_2\end{bmatrix}$$

where $\mathbf{T}_c=\left[T_c\left(c_i^{(\mathrm{U})},c_j^{(\mathrm{M})}\right)\right]_{N_c\times M_c}$, $\mathbf{T}_c'$ is the transpose of $\mathbf{T}_c$, and $\mathbf{D}_1$ and $\mathbf{D}_2$ are two diagonal matrices, such that $D_1(i,i)=\sum_j T_c\left(c_i^{(\mathrm{U})},c_j^{(\mathrm{M})}\right)$ and $D_2(j,j)=\sum_i T_c\left(c_i^{(\mathrm{U})},c_j^{(\mathrm{M})}\right)$. According to [4], the second eigenvector $\mathbf{x}$ of the generalized eigenvalue problem $\mathbf{\Gamma x}=\lambda\mathbf{D x}$ provides a real relaxation to the discrete optimization problem of finding the minimum normalized cut. Letting $\mathbf{x}=\left[\mathbf{x}_1,\mathbf{x}_2\right]'$, $\mathbf{y}^{(\mathrm{L})}=\mathbf{D}_1^{-1/2}\mathbf{x}_1$ and $\mathbf{y}^{(\mathrm{R})}=\mathbf{D}_2^{-1/2}\mathbf{x}_2$, and after a little algebraic manipulation, we get $\mathbf{D}_1^{-1/2}\mathbf{T}_c\mathbf{D}_2^{-1/2}\mathbf{y}^{(\mathrm{L})}=(1-\lambda)\mathbf{y}^{(\mathrm{L})}$ and $\mathbf{D}_2^{-1/2}\mathbf{T}_c\mathbf{D}_1^{-1/2}\mathbf{y}^{(\mathrm{R})}=(1-\lambda)\mathbf{y}^{(\mathrm{R})}$. These are precisely the equations that define the singular value decomposition (SVD) of the normalized matrix $\mathbf{A}=\mathbf{D}_1^{-1/2}\mathbf{T}_c\mathbf{D}_2^{-1/2}$. Here $\mathbf{y}^{(\mathrm{L})}$ and $\mathbf{y}^{(\mathrm{R})}$ are the left and right singular vectors respectively, while $(1-\lambda)$ is the corresponding singular value. Thus we can compute the left and right singular vectors corresponding to the second (largest) singular value of $\mathbf{A}$. Computationally, working on $\mathbf{A}$ is much better since $\mathbf{A}$ is of size $N_c\times M_c$ while the matrix $\mathbf{\Gamma}$ of the larger size $(N_c+M_c)\times(N_c+M_c)$.

The singular vectors $\mathbf{y}_2^{(\mathrm{L})}$ and $\mathbf{y}_2^{(\mathrm{R})}$ of $\mathbf{A}$ give a real approximation to the discrete optimization problem of minimizing the normalized cut. That is, the right singular vector $\mathbf{y}_2^{(\mathrm{R})}$ will give us a bi-partitioning of user clusters while the left singular vector $\mathbf{y}_2^{(\mathrm{L})}$ will give us a bi-partitioning of movie clusters. On this basis, the literature [4] proposes the following multi-partition algorithm:

Step 1. Given $\mathbf{T}_c$, form $\mathbf{A}=\mathbf{D}_1^{-1/2}\mathbf{T}_c\mathbf{D}_2^{-1/2}$.

Step 2. Compute $l=\lceil\log_2(k)\rceil$ singular vectors of $\mathbf{A}$, $\mathbf{y}_2^{(\mathrm{L})},\cdots,\mathbf{y}_{l+1}^{(\mathrm{L})}$ and $\mathbf{y}_2^{(\mathrm{R})},\cdots,\mathbf{y}_{l+1}^{(\mathrm{R})}$, and form the matrix

$$\mathbf{Z}=\left[\mathbf{D}_1^{-1/2}\mathbf{Y}^{(\mathrm{L})}\quad \mathbf{D}_2^{-1/2}\mathbf{Y}^{(\mathrm{R})}\right]'$$

where *k* is the desired cocluster number.

Step 3. Run the *k*-means algorithm on the *l*-dimension data $\mathbf{Z}$ to obtain the desired *k*-way multi-partitioning.

By using this algorithm, users and movies can be finally co-clustered into a set of co-clusters $\mathbf{O}$. Intuitively, coarse user clusters and coarse movie clusters in the same co-cluster may have relatively strong correlation. In other words, there is a high probability that a user may rate a movie in the same co-cluster. On the other hand, coarse user clusters and coarse movie clusters in different co-clusters may also have some rating links between them, but the rating probability is relatively small. To measure this difference, we use a probability function s based on the distances of the corresponding features in Z for coarse clusters and co-cluster centers.

Thus link existence predictions can be generated based on the results of this two-phase co-clustering. The details of the link existence prediction model will be further discussed in the following section.

3. LINK EXISTENCE PREDICTION

To predict the link existence probability for a given user-movie pair (u, f), we first compute the *affinity* (or similarity) of a user *u* and a movie *f*, and then derive the probabilistic link existence model using the affinity model.

3.1 Measuring the affinity

In our solution, a probabilistic model is employed to measure the affinity between a user *u* and a movie *f*. As shown in Figure 3(a), a similarity propagation chain is formed by exploiting the two-phase co-clustering. Following this similarity propagation chain, we can easily obtain the affinity between a user *u* and a movie *f* as follows:

$$s_{\text{Coclustering}}(u, f) = \sum_{o_i \in \mathbf{O}} \sum_{o_j \in \mathbf{O}} s(c_u^{(\mathrm{U})}, o_i)\, s(o_i, o_j)\, s(o_j, c_f^{(\mathrm{M})}) \qquad (3)$$

For the coarse clustering, each user *u* (or movie *f*) corresponds to only one cluster while each cluster may have many members. For simplicity, let $c_u^{(\mathrm{U})}$ denote the cluster of user *u*, $c_f^{(\mathrm{M})}$ denote the cluster of movie *f*. On the other hand, each coarse cluster may correspond to several co-clusters. As illustrated by Figure 2, correlation may exist among different co-clusters. Here we use $s(o_i, o_j)$ to denote the affinity among co-clusters $o_i \in \mathbf{O}$ and $o_j \in \mathbf{O}$. In our approach, the score of $s_{\text{Coclustering}}(u, f)$ is used to measure the affinity between a user *u* and a movie *f*.

3.2 Link existence prediction model

Similar to the existence uncertainty of PRMs [11], we introduce a link existence variable $E_{u \to f}$ for link $u \to f$ (i.e., a rating event of a movie $f \in \mathcal{F}$ by a user $u \in \mathcal{U}$). From the affinity model, we have

$$P(E_{u \to f} \mid u, f) = P(E_{u \to f} \mid c_u^{(\mathrm{U})}, c_f^{(\mathrm{M})}) \qquad (4)$$

where $P(E_{u \to f} \mid u, f)$ denotes the probability of a link $u \to f$ existing between a user node *u* and a movie node *f*, while $P(E_{u \to f} \mid c_u^{(\mathrm{U})}, c_f^{(\mathrm{M})})$ denotes the link existence probability due to the hidden correlation between a coarse user cluster and a coarse movie cluster revealed by the two-phase co-clustering. The user

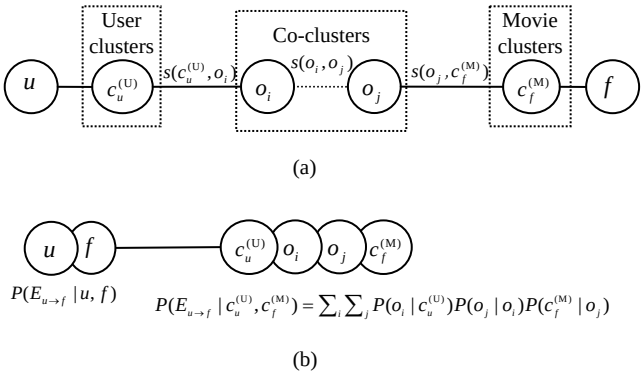

Figure 3. (a) The affinity model based on co-clustering. (b) The link existence prediction model.

node *u* belongs to the coarse user cluster $c_u^{(\mathrm{U})}$ and the movie node *f* belongs to coarse user cluster $c_f^{(\mathrm{M})}$.

According to the similarity propagation chain shown in Figure 3, the probability $P(E_{u \to f} \mid c_u^{(\mathrm{U})}, c_f^{(\mathrm{M})})$ can be computed by:

$$P(E_{u \to f} \mid c_u^{(\mathrm{U})}, c_f^{(\mathrm{M})}) = P(c_u^{(\mathrm{U})} \mid u) P(c_f^{(\mathrm{M})} \mid c_u^{(\mathrm{U})}) P(f \mid c_f^{(\mathrm{M})}) \qquad (5)$$

where $P(c_u^{(\mathrm{U})} \mid u) = 1$ for each user corresponds to only one coarse user cluster, and $P(f \mid c_f^{(\mathrm{M})}) = 1/|c_f^{(\mathrm{M})}|$ to approximate the probability of choosing a movie *f* given a coarse movie cluster. The link probability between the user cluster $c_u^{(\mathrm{U})}$ and the movie cluster $c_f^{(\mathrm{M})}$, i.e., $P(c_f^{(\mathrm{M})} \mid c_u^{(\mathrm{U})})$ depends on their distribution over co-clusters and the relationships of co-clusters that they are related to. Thus we have

$$P(c_f^{(\mathrm{M})} \mid c_u^{(\mathrm{U})}) = \sum_{o_i \in \mathrm{O}} \sum_{o_f \in \mathrm{O}} P(o_i \mid c_u^{(\mathrm{U})}) P(o_j \mid o_i) P(c_f^{(\mathrm{M})} \mid o_j)\,. \qquad (6)$$

The three probabilities in Equation (6) have similar calculation formulas for all of $c_u^{(\mathrm{U})}, c_f^{(\mathrm{M})}, o_i, o_j$ are projected in the *l*-dimension space. Take $P(o_i \mid c_u^{(\mathrm{U})})$ for example, which can be calculated by

$$P(o_i \mid c_u^{(\mathrm{U})}) = \frac{s(o_i \mid c_u^{(\mathrm{U})})}{\sum_{o_j \in \mathrm{O}} s(o_j \mid c_u^{(\mathrm{U})})} \qquad (7)$$

Here the similarity *s* can be expressed as a function of Euclidean distance in the *l*-dimension space.

Finally, we can generate the existence probability of any directed link in the bigraph based on Equation (4) to (7). Our link existence prediction model is very efficient. On a notebook with a Pentium 1.6GHz CPU and 768MB RAM, it takes only 0.551 second to predict the existence probabilities of the 100 thousand user-movie pairs in the test data set.

4. EXPERIMENTAL RESULTS

We evaluate our model on the Netflix data set, which is also employed by the KDD CUP Competition of 2007. This data set consists of more than 100 million ratings from over 480189 randomly-chosen, anonymous customers on nearly 17770 movie titles. The data were collected between October, 1998 and De-

cember, 2005 and reflect the distribution of all ratings received by Netflix during this period. The test data set is a list of 100,000 user-movie pairs where the users and movies are drawn from the Netflix data set. None of the pairs were rated in the training set.

RMSE. The root mean squared error (RMSE) is used to evaluate the prediction results, as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_i (x_i - y_i)^2} \quad (8)$$

where n is the size of the test list, y_i is the prediction for the i'th user-movie pair, and x_i is the correct value of this pair which 1 means existence and 0 otherwise. Models with the lower RMSE are considered the better.

4.1 Parameters selection

There are two kinds of important parameters that need to be carefully chosen: the number of coarse cluster centers when using K-means, the number k of co-clusters when using K-means in the l-dimension space Z.

As shown in Table 1, the number K of coarse user clusters is chosen by comparing RMSE with co-cluster number of 100 and 200. The number of coarse movie clusters in K-means is set as 500. We can see that the performance is relatively better when there are 5000 coarse user clusters. Thus we choose 5000 as the number K of coarse user clusters in the following experiments.

In our experiments, we get 500 coarse movie clusters and 5000 coarse user clusters by K-means. Using these settings, we experiment to choose the number k of co-clusters. The results are shown in Table 2. In the following experiments, we choose 100 as the number k of co-clusters which gets best RMSE.

Table 1. RMSE for different numbers of coarse user cluster centers with 100 and 200 co-clusters

Co-cluster num \ User cluster num	1000	5000	10000
100	0.275269	0.267418	0.278148
200	0.276187	0.275619	0.275657

Table 2. RMSE using different numbers of co-cluster centers

k	10	50	100	200
RMSE	0.271499	0.271161	0.267418	0.275619

4.2 Evaluation

In this sub-section, we evaluate our model against some other prediction models. First we compare our model with some constant link probability models that assign equal constants to each link pair as link existence probability. Shown in Table 3, we compare RMSE of our model with those of the constant link probability models with constants c=0, 0.08 and 1, where 0.08 is an approximate proportion of rating links against all possible links in the training data set. We also compare the results of our model with the model which only use one-phase clustering but don't use co-clustering.

In Table 3, our model performs better than all other link probability models. However, the RMSE using 0.08 is very near to that of our model. There are many possible reasons. One important reason is that the size of test set (100 thousand) is too small compared with all possible rating links 480189*17770 (over 8 billion). Another possible reason is the sparsity of rating data in the training data set, which makes the prediction probabilities from statistical learning very small. But the ground truth is 1 for each existing links, so the RMSE cannot be very high.

Another advantage of our model is that it assigns different probability values to different pairs according to the correlation of the user and the movie in each pair. The larger the probability value is, the more likely the user rated the movie. The histogram of rating probability values in our model is illustrated in Figure 3. Different from the constant model which uses 0.08 for every pair, our model generates values raging from 0 to 0.7.

Table 3. The evaluation of the proposed model against some other models in test set

RMSE \ Models	c=0	c=0.08	c=1	One-phase clustering	Best of our model
all test pairs	0.279	0.268	0.960	0.282576	0.265575

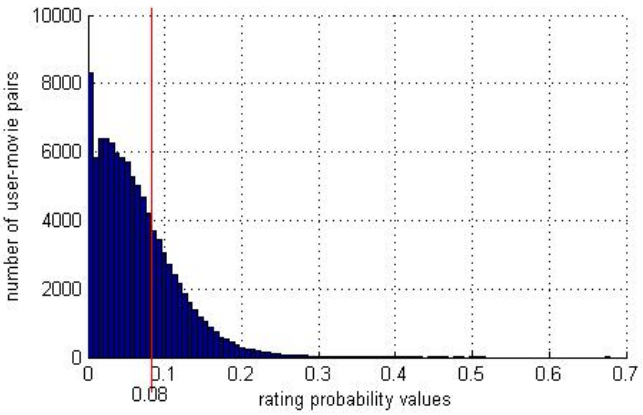

Figure 3. The histogram of the rating probability values.

5. CONCLUSIONS

In this paper, we have introduced our approach of solving the "Who Rated What" task in KDD CUP 2007 competition. By modeling the user-movie rating collection as a bigraph, we use a two-phase bigraph co-clustering strategy. We derive our models to predict the link existence probabilities based on the results of co-clustering. Experimental results show that our approach works well in the task.

6. ACKNOWLEDGEMENTS

This work is supported by grants from Chinese NSF under contract No. 60605020, National Hi-Tech R&D Program (863) of China under contract No. 2006AA01Z320 and 2006AA010105, and National Key Technology R&D Program under contract No. 2006BAH02A10.

7. REFERENCES

[1] Marlin, B. Modeling User Rating Profiles for Collaborative Filtering. In *Proceedings of Neural Information Processing Systems (NIPS2003.)* 627-634, Cambridge, MA. MIT Press.

[2] Getoor, L. Link mining: a new data mining challenge. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. *SIGKDD Explorations* 5(1): 84-89 (2003).

[3] Getoor, L. and Diehl, C. P. Link mining: a survey. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2005), SIGKDD Explorations* 7(2): 3-12 (2005).

[4] Dhillon, I. S. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 26 - 29, 2001, San Francisco, California, USA.* KDD 2001: 269-274.

[5] Fern, X. Z. and Brodley,C. E. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the Twenty-First International Conference on Machine Learning (ICML 2004) in Banff, Alberta, Canada from July 4-8, 2004.* 281-288.

[6] Hasan, M., Chaoji, V., Salem, S. and Zaki, M. J. Link Prediction using Supervised Learning, In *Proceedings of the Workshop on Link Analysis, Counter-terrorism and Security (with SIAM Data Mining Conference), Bethesda, MD, 2006*.

[7] Dhillon, I. S., Mallela, S., and Modha, D. S. Information-theoretic co-clustering. In *Proceedings of The Ninth ACM SIGKDD International Conference*, 89-98, 2003.

[8] Madadhain, O. J., Hutchins, J. and Smyth, P. Prediction and ranking algorithms for event-based network data. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2005) SIGKDD Explorations* 7(2): 23-30 (2005).

[9] Airoldi, E. and Carley, K. M. Sampling algorithms for pure network topologies: a study on the stability and the separability of metric embeddings. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2005). SIGKDD Explorations* 7(2): 13-22 (2005)

[10] Rege, M., Dong, M. and Fotouhi, F. Co-clustering Documents and Words Using Bipartite Isoperimetric Graph Partitioning. In *Proceedings of the IEEE International Conference on Data Mining* ICDM 2006: 532-541

[11] Getoor,L., Friedman,N., Koller,D. and Taskar. B. Learning Probabilistic Models of Link Structure. *Journal of Machine Learning Research* 3: 679-707 (2002).

[12] Popescul, A. and Ungar, L. H. Statistical Relational Learning for Link Prediction. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)-2003*.

[13] Huang, Z., Li, X. and Chen, H. Link prediction approach to collaborative filtering. In *Proceedings of the Joint Conference on Digital Libraries (JCDL 2005*), *Denver, Colorado, USA in June 7-11, 2005*.: 141-142

[14] Taskar, B., M., Wong, F. P.,Abbeel and Koller, D. Link Prediction in Relational Data. In *Proceedings of Neural Information Processing Systems (NIPS2003)*.

[15] Hofmann, T. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst. 22(1)*: 89-115 (2004)

[16] Hoff, P., D., Raftery, A. E. and Handcock, M. S. Latent space approaches to social network analysis. *Journal of the American Statistical Association; Dec 2002; 97, 460; ABI/INFORM Global* pg. 1090.

[17] Shortreed, S., S. Handcock M. and Hoff, P. Positional Estimation within a Latent Space Model for Networks. *Methodology*, Vol. 2, No. 1, 2006

[18] Madadhain, J. O. and Smyth, P. EventRank: a framework for ranking in time-varying networks. In *Proceedings of the ACM SIGKDD Workshop on Link Discovery*, August 2005.

[19] Sarkar, P. and Moore, A. W. Dynamic social network analysis using latent space models. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2005). SIGKDD Explorations* 7(2): 31-40 (2005).