Bootstrapping Recommender Systems with the Crowdsourcing

SEMESTER PROJECT REPORT

SUBMITTED BY
Cheng-Hsiang Chiu

SUPERVISED BY
Goran Radanovic

January 2015

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

TABLE OF CONTENTS

1. Introduction

Recommender system is an information filtering technique aiming at predicting meaningful information to users, which is widely implemented for the purpose of suggesting books to consumers in Amazon, movies in Netflix , music in Pandora and so on [1]. The system recommends items to users based on a prediction of users' profiles. Nowadays, recommender system faces a cold start problem which would jeopardize the precision of the prediction. Cold start problem arises from the situation in which there is not enough information to make a good recommendation. In this paper, by taking advantages of crowdsourcing we propose a technique to mitigate the consequences caused by the cold start problem.

The report is structured as follows. In Section 2 we briefly introduce background of recommender system and the approaches behind it. Then related works are also given in this section. After that, the proposed technique is presented in Section 3. Next, Section 4 would cover the experimental results including our proposed method and two comparison methods. Finally, conclusions and reference are noted.

2. Background and Related Work

We first give a brief introduction to recommender system, collaborative filtering techniques, active learning and three performance metrics before describing the proposed idea.

2.1 Recommender System

There are two main approaches to construct the recomender system : content-based filtering and collaborative filtering [2]. Content-based filtering method is based on item profile which is a record or collection of records that describes the characteristics of that item. The item profile of a movie could be the actors, the director, the genre, etc. Items which are featured with the identical profiles as a certain item would be recommended to users who are fond of the specific item.

The other method is collaborative filtering technique, which has two implementations. The necessity of processing users' preferences instead of analyzing item profiles explains how the term "collaborative" comes from. More details about collaborative filtering method would be given in the next section.

In recommender systems, researchers usually transform a data set into a utility matrix, shown in Figure 1 in which columns are denoted as the ratings of one item rated by some users and row are described as the rating of a user given to some items.

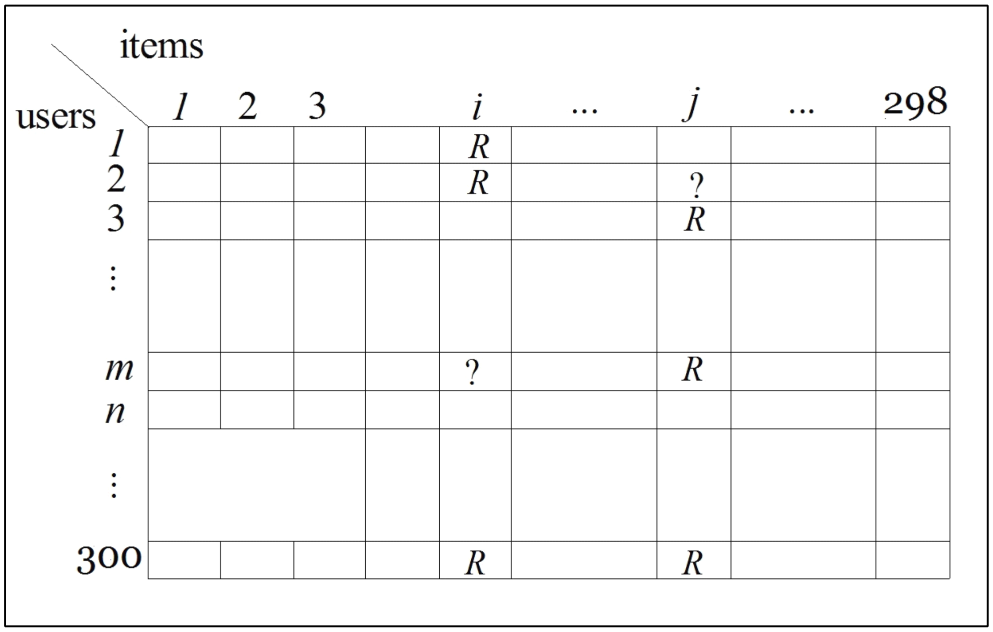


Figure 1 : A utility matrix with columns which are ratings of an item and rows are ratings which are given by a user over some items.

2.2 Collaborative Filtering

Collaborative filtering method recommends items on the basis of similarity distance between users and/or items. While two users have highly similar interests on movies, music, books, and etc., then there is a good reason to recommend some favorite items of one user to the other user, which is the theory of user-based collaborative filtering technique. In order to measure the similarity distance between two users, shown in Figure 2, there are two common approaches, Cosine distance and Pearson correlation, expressed in Equations (1) and (2) respectively. Cosine distance between user *m* and user *n* is defined as

$$sim(m,n) = \frac{\sum_{i \in S_m \cap S_n} r_{m,i} \times r_{n,i}}{\sqrt{\sum_{i \in S_m \cap S_n} r_{m,i}^2} \sqrt{\sum_{i \in S_m \cap S_n} r_{n,i}^2}}, \tag{1}$$

where S_m represents the set of items that user *m* has rated and $r_{m,i}$ denotes the rating of item *i* of user *m* [3][4][5].

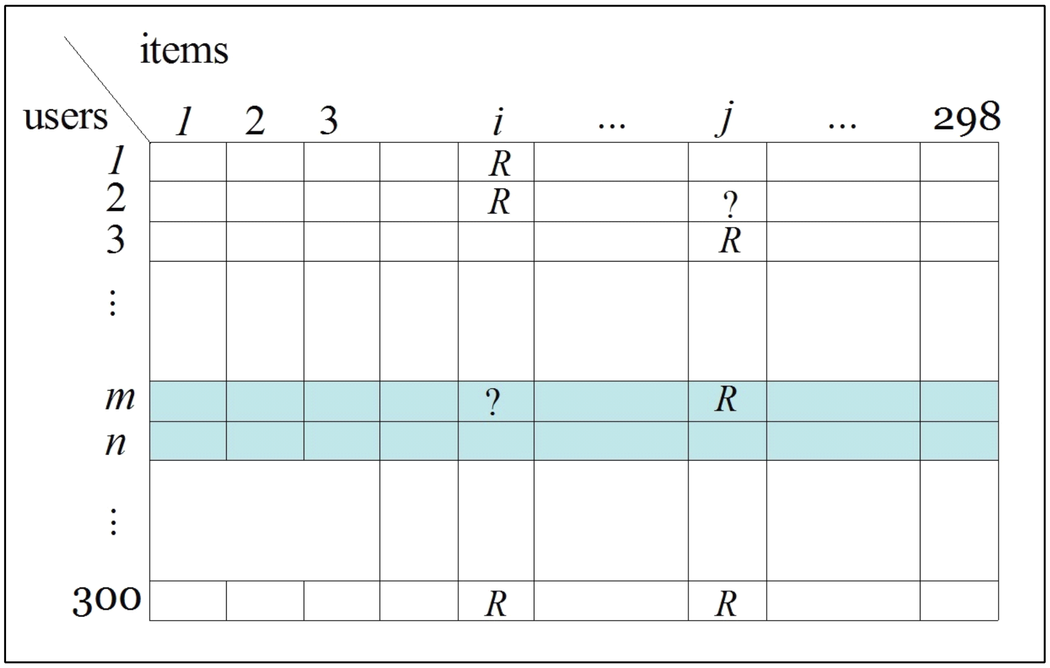


Figure 2 : A utility matrix where rows of user *m* and user *n* are marked in blue.

Pearson correlation is given in the following equation,

$$sim(m,n)=\frac{\sum_{i\in S_m\cap S_n}\left(r_{m,i}-\overline{r_m}\right)\left(r_{n,i}-\overline{r_n}\right)}{\sqrt{\sum_{i\in S_m\cap S_n}\left(r_{m,i}-\overline{r_m}\right)^2}\sqrt{\sum_{i\in S_m\cap S_n}\left(r_{n,i}-\overline{r_n}\right)^2}}, \tag{2}$$

where $\overline{r_m}$ stands for the average rating of user *m*. After the similarity distance is measured, the subsequent job is to make predictions and suggest the top ranked recommendations to users. The predicted item can be calculated as follows :

$$p_{m,i}=\overline{r_m}+\frac{\sum_{\{u\in U|i\in S_n\}} sim(m,n)\times(r_{n,i}-\overline{r_n})}{\sum_{\{u\in U|i\in S_n\}}\left|sim(m,n)\right|}. \tag{3}$$

Besides the user-based collaborative filtering approach, item-based collaborative filtering is the other technique. The idea behind user-based collaborative filtering is the same as user-based collaborative filtering. Recommender systems suggest some similar items to users according to the similarity matrix between items and items, depicted in Figure 3.

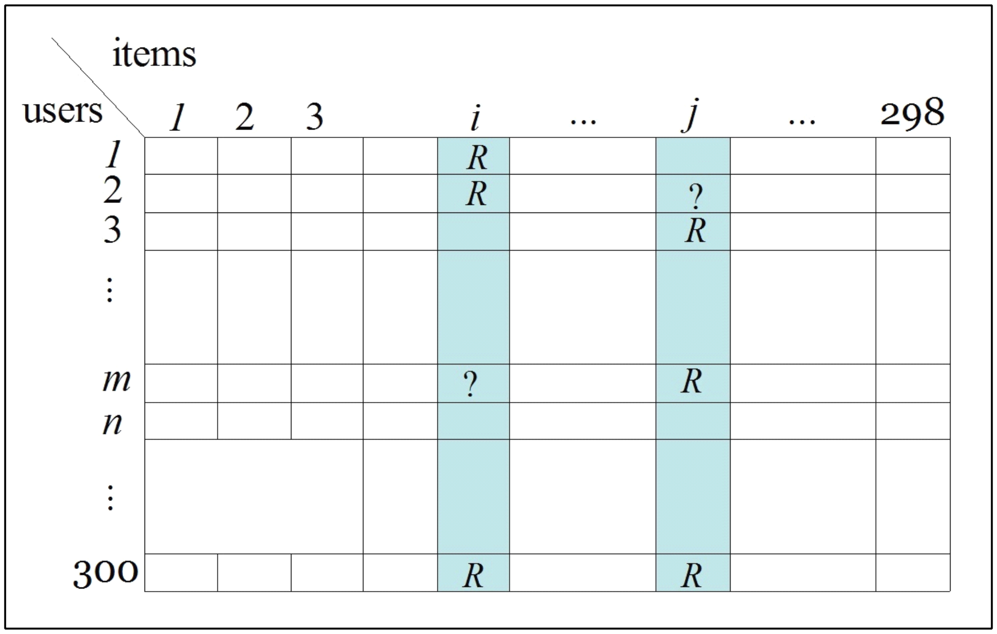


Figure 3 : A utility matrix in which columns of item *i* and item *j* are marked in blue.

The similarity distance of item-based approach between item *i* and item *j* is formulated by using either Cosine distance, shown in equation (4), or Pearson correlation, expressed in (5).

$$sim(i,j) = \frac{\sum_{\{u \in U | i \in S_u \& j \in S_u\}} r_{u,i} \times r_{u,j}}{\sqrt{\sum_{\{u \in U | i \in S_u \& j \in S_u\}} r_{u,i}^2} \sqrt{\sum_{\{u \in U | i \in S_u \& j \in S_u\}} r_{u,j}^2}}, \tag{4}$$

and

$$sim(i,j) = \frac{\sum_{\{u \in U | i \in S_u \& j \in S_u\}} \left(r_{u,i} - \overline{r_u}\right)\left(r_{u,j} - \overline{r_u}\right)}{\sqrt{\sum_{\{u \in U | i \in S_u \& j \in S_u\}} \left(r_{u,i} - \overline{r_u}\right)^2} \sqrt{\sum_{\{u \in U | i \in S_u \& j \in S_u\}} \left(r_{u,j} - \overline{r_u}\right)^2}}, \tag{5}$$

where $r_{u,i}$ is described as the rating of user *u* on item *i* and $\overline{r_u}$ as the average rating that user *u* has rated. And the prediction of item is shown in the following equation,

$$p_{m,i} = \frac{\sum_{\{j \in S_m | j \neq i\}} sim(i,j) \times r_{m,j}}{\sum_{\{j \in S_m | j \neq i\}} \left| sim(i,j) \right|}. \tag{6}$$

2.3 Performance Metrics

After the prediction is finished, we have to evaluate the quality of recommender systems. Generally, there are two common-used performance metrics. They are Mean Absolute Errorr (MAE) and Root Mean Square Error (RMSE). MAE and RMSE are formulated in (7) and (8) respectively :

$$MAE = \frac{1}{|T|} \sum_{(u,i,r) \in T} \left| p_{u,i} - r \right|, \tag{7}$$

and

$$RMSE = \sqrt{\frac{1}{|T|}\sum_{(u,i,r)\in T}\left(p_{u,i} - r\right)^2}\ , \tag{8}$$

where the set of (user, item, rating) triplets is represented as T.

Besides, Normalized Discounted Cumulative Gain (NDCG) is commonly used as well . It measures the performance of a recommender system on the basis of graded relevance of the recommended items, with 1.0 denoting the best ranking and 0 the worst ranking, and is formulated in (9),

$$NDCG_k = \frac{DCG_k}{IDCG_k}, \tag{9}$$

where DCG_k is defined in (10) or (11), and $IDCG_k$ is the maximal DCG_k,

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)}\ , \tag{10}$$

$$DCG_k = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{\log_2(i)} \tag{11}$$

NDCG is demonstrated in the following example extracted from wikipedia [10]. Suppose that the ordered items are recommended as I1, I2, I3, I4, I5, I6 and the corresponding relevance scores are 3, 2, 3, 0, 1, 2. Then, we can get

$$DCG_6 = rel_1 + \sum_{i=2}^{6} \frac{rel_i}{\log_2(i)} = 3 + (2 + 1.892 + 0 + 0.431 + 0.774) = 8.1,$$

and the maximal DCG_6 is 8.69 by a decreasing order, 3, 3, 2, 2, 1, 0, which results in

$$NDCG_6 = \frac{DCG_6}{IDCG_6} = \frac{8.10}{8.69} = 0.932$$

2.4 Active Learning

In machine learning, there are two famous learning categories, supervised learning and unsupervised learning. In supervised leaning, based on a given bunch of pair of inputs and labels, a model is trained to predict the label of a new input. The contrary case of supervised learning is unsupervised learning. Instead of finding a model which labels the inputs, unsupervised learning is intended to figure out the intrinsic information of inputs.

Sometimes there is a large amount of inputs, and labeling all data is inevitably expensive. An alternative way is to label parts of the inputs. This method is called active learning, which is a case of semi-supervised learning. In the proposed technique, we adopted the idea of active learning to label parts of the items and still achieved a satisfactory result.

2.4 Related Work

In the paper [6] "Alleviating the Sparsity in Collaborative Filtering using Crowdsourcing," a technique is proposed to thwart the cold start problem by the help of crowdsourcing. The approach is to expand the dimension of users, that is, adding the extra workers' ratings to the original utility matrix. Then, the current utility matrix turns out to be the one shown in Figure 4. According to the experimental results revealed in the paper, quality of the recommender system improves. More informations indeed bolsters the precision of recommender systems. But the approach is implicit rather than explicit, because it adds new users hoping that they are similar to the existing user. In crowdsourcing, however, there is a concern regarding the existence of sloppy workers and spammers. Moreover, there is no ground truth of these workers. Therefore, the above reasons lead us to explicit strategy where we fill in the blank elements directly using crowdsourcing.

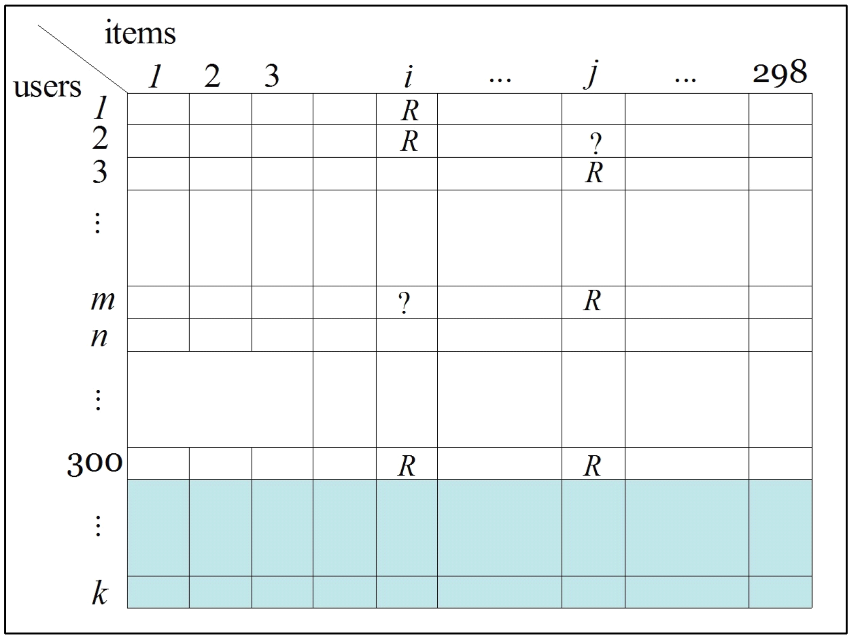


Figure 4 : An extended utility matrix where extra rows marked in blue are obtained from the crowdsourcing.

Another paper was introduced in 2014 [9], "Item Cold-Start Recommendations : Learning Local Collective Embeddings. Instead of focusing on collective filtering technique, the authors adopted both content-based and collective filtering solutions and proposed a collective matrix factorization method that decomposes content and collaborative matrices in low-dimensional space without compromising geometrical structure of data.

We used some experimental results from the prior semester project report [8], and conducted more experiments to support the proposed technique.

3. The Proposed Technique

The quality of the recommender system is highly compromised because of existence of the cold start problem. To mitigate the effect caused by the problem, the utility matrix should be as less sparse as possible. Therefore, we proposed an approach that tries to fill in some blank elements by taking the advantage of crowdsourcing. With the help of crowdsourcing, the recommender system is able to suggest items to users with more precisions.

3.1 System Overview

The following figure is the overall system which has two subparts. The left flow in Figure 5 has the purpose of checking the quality of the recommender system by the use of performance matrix introduced in equations (7), (8) and (9). The right flow focuses on gathering data with crowdsourcing.

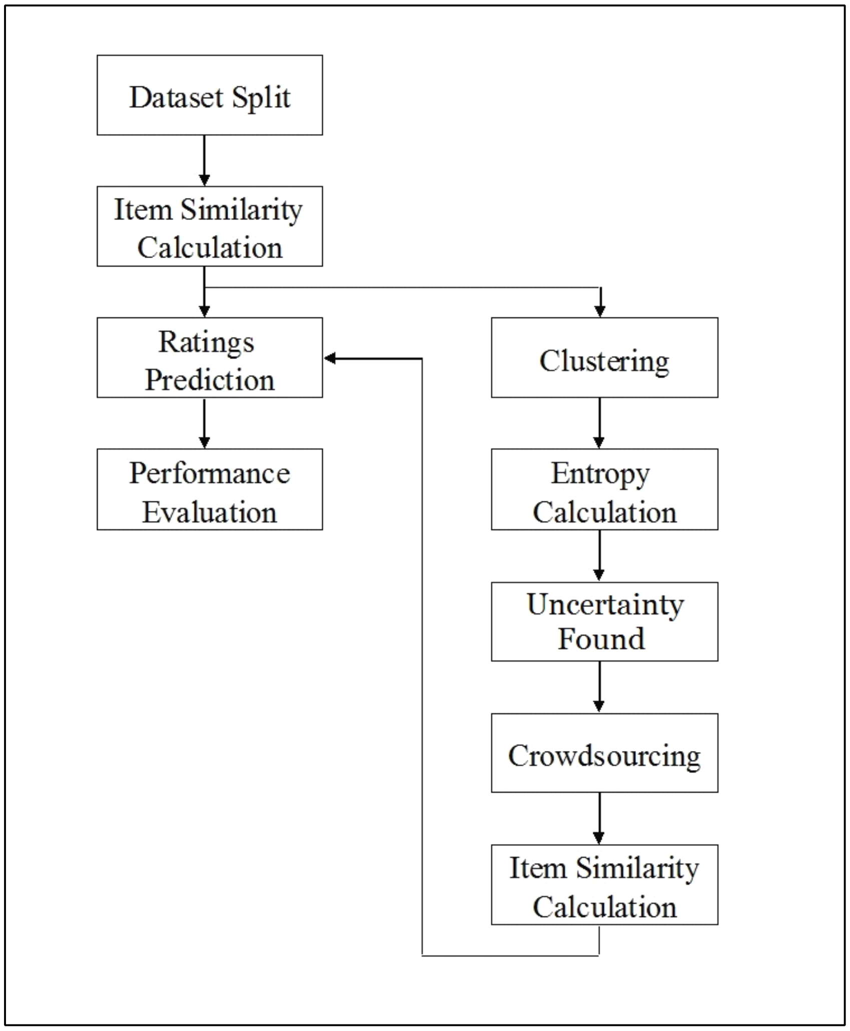


Figure 5 : System Flowchart

3.2 The Left Flow - Performance Evaluation

The left flow is basically the job of evaluating performance of all the recommender system and works in the following ways :

1. Measure the item-item similarity distances by the help of equation (5).
2. Predict the ratings which appears in the testing set only by using formula (6) and the similarity distances obtained in the first step.
3. Evaluate the performance matrics, RMSE, MAE and NDCG, over the data set and the predicted ratings with equations (7), (8) and (9).

After finishing the above three steps, we can tell whether the recommender system is performed well or bad by comparing with the results of different approaches.

3.3 The Right Flow - Crowdsourcing

The proposed technique is implemented in the right flow and is consisted of seven steps :

1. Measure the user-user similarity matrix with equation (5).
2. Implement K-means clustering algorithm on the training set to put similar users together.
3. Calculate entropy of each user with respect to all the centroids. We wish to attain good learning performance without demanding too many users. Hence, entropy, one of the common-used approaches in active learning, is utilized to label the most uncertain users [7]. Entropy of users j is formulated in the following :

$$Entropy(j) = -\sum_{\{x \in centroids\}} \left(p_{j,x} \times \log p_{j,x}\right), \tag{12}$$

where $$p_{j,x} = \frac{\left|\vec{j} - \vec{x}\right|}{\sum_{\{y \in centroids\}} \left|\vec{j} - \vec{y}\right|} \tag{13}$$

and $\vec{j} = \{sim(j,k), k = 1,2,\cdots,1682\}$ denotes a vector of user j with respect to all the users.

4. Pick some users which are top-ranked in descending entropy. Since the higher entropy value an user has, the higher uncertainty it possesses. The top uncertain users are the red dots demonstrating in Figure 6.

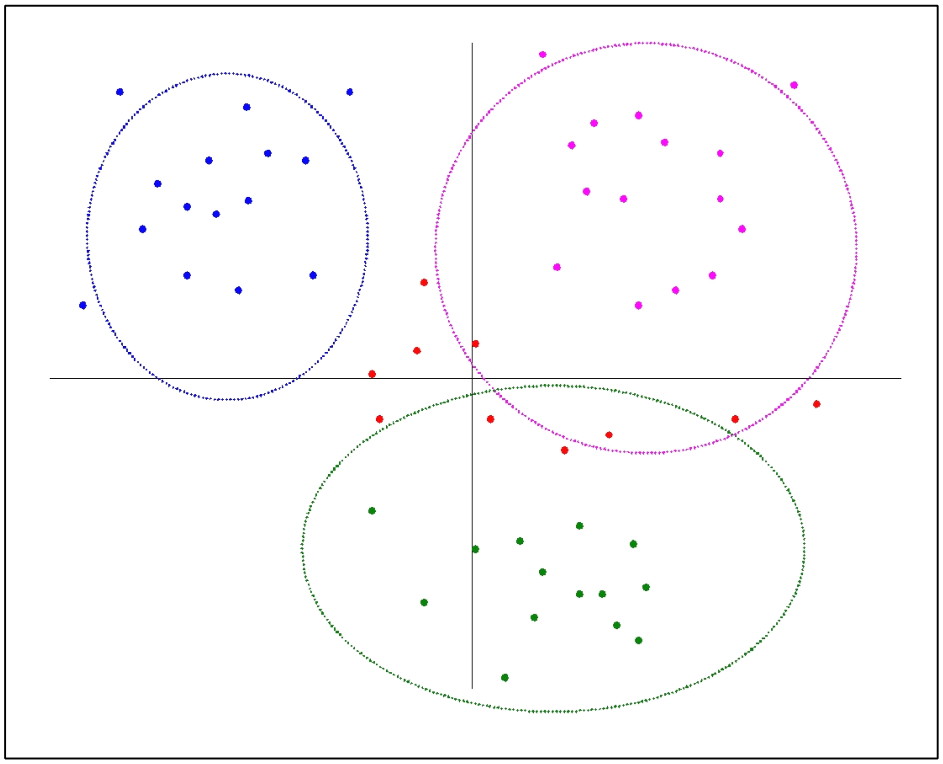


Figure 6 : An demonstration in which users are clustered. The most uncertain users are those dots marked in red.

5. Post the built web page on the Amazon Mechanical Turk for workers to rate the uncertain users. What we want to do is to add more ratings in the original utility matrix. Hence, the workers who are hired to help us rate items should pretend to a

specific user. For example, in Figure 3 the rating of user *m* on item *i* is blank and item *i* is supposed to be a uncertain item. After referring to some known ratings of user *m* on items other than item *i*, workers should predict the rating of item *i* on behalf of user *m*; that is to say, workers have to figure out the preference profile of user *m* and give the most probable rating for item *i*.

6. Deploy a quality control mechanism in the web page to filter out sloppy workers or spammers. Since they rate items casually, the variance between the ratings they give and the real ratings would be undoubtedly huge . Hence, the mechanism is to calculate a score based on the known ratings and predicted ratings and the score is given in the follwoing :

$$score = 200 \times \left(1 - \left|\frac{known\_rating - predicted\_rating}{4}\right|\right). \tag{14}$$

The score decreases as the variance increases. And the higher the score is, the more reliable the worker's predictions should be. Therefore, the predicted ratings are aggregated in the end with the following equation :

$$r_\alpha = \frac{\sum_{\beta \in W} R_\beta \times \frac{score(\beta)}{1000}}{\sum_{\beta \in W} \frac{score(\beta)}{1000}}, \tag{15}$$

where r_α is represented as the rating of uncertain item α , W={all workers who rated item α } and β is a worker. The idea of equation (15) is that the workers of the higher scores gain more dominant portion in the aggregation to the final predicted ratings.

7. Add the extra ratings into the original utility matrix and repeat the left flow depicted in Figure 5.

After these steps, the utilty matrix becomes more dense than the original matrix. And the predicted ratings will be more precise since more information are presented.

4. Experimental Results

Before presenting the experimental results, we explain the data set and how we processed it. From now on, our proposed technique is referred to as explicit method.

4.1 Data Set

The data set is from MovieLens which provides data sets of three different sizes, 100K, 1M and 10M. Not only ratings but users' profiles and items' profiles are all included in the data set. We used the one of size 100K in which 300 users and 298 movies were selected in the experiment. The reason why downsizing the data set to 300 users and 298 movies is that one of the comparison method we used requires the movies descriptions. And among the original data set, in which there are 943 users and 1682 movies, some descriptions have already been stored in the database we built last time. Hence, this time the smaller data set is used.

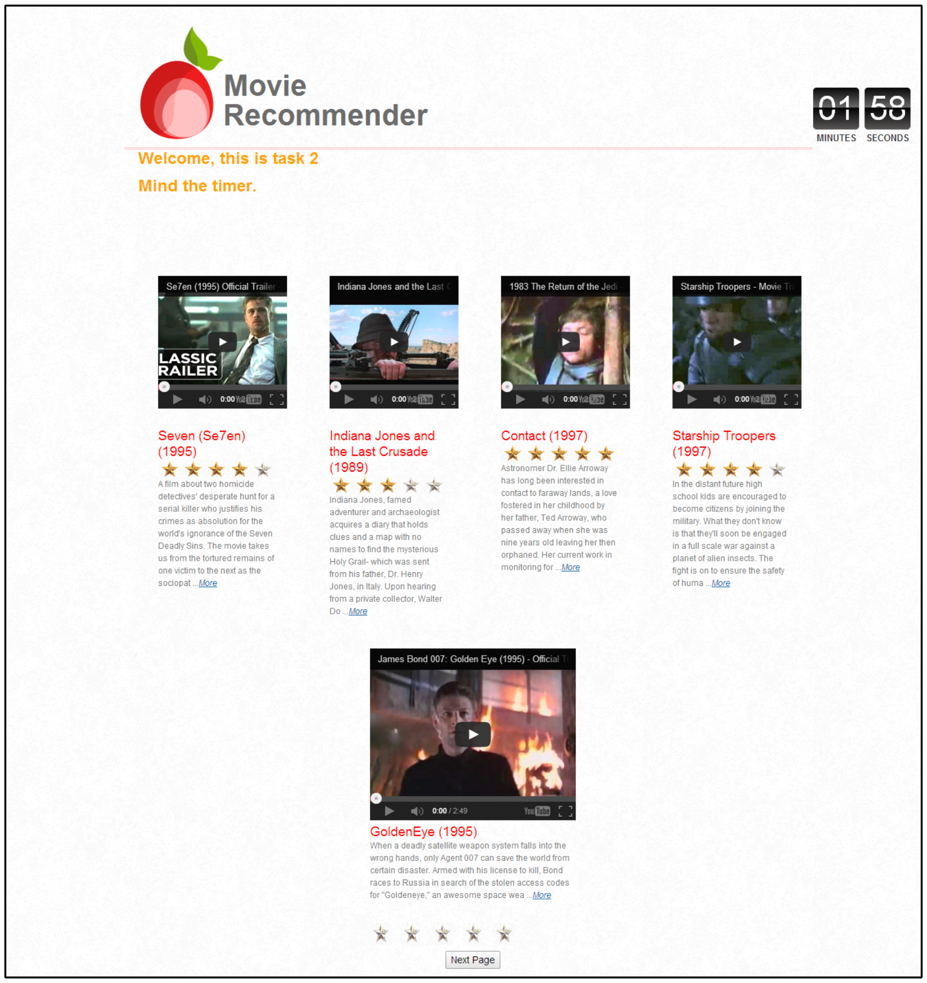
4.2 Preprocessing of Data Set

We used the method introduced in [6], which is called implicit method in the report, and [9] as the baselines for comparisons. Since in this paper [9], called LCE here, contend-based is needed, we applied information retrieval technique to the data set. In the beginning, we constructed a word dictionary based on the descriptions of all 298 movies which were extracted from IMDB website. Then, each movie is represented as a vector with respect to the word frequency appearing in the dictionary.

Moreover, for re-conducting the experiment of implicit method, 298 movies were divided into two parts, 250 movies of them are training data set and the remaining 48 movies are supposed to be the new data set.

4.3 Web Page for Amazon Mechanical Turk

We built one web page for our proposed method, the explicit method, and another for implicit method as a comparison. The following figure is a snapshot of the first web page. In Figure 7, the movie located on the bottom is not rated and is one element in the uncertainty set. Workers predict the ratings of the movies based on users' preference profiles, which are consisted of movie trailers, movie descriptions and ratings of other four movies. And each worker rates 30 movies.


Movie Recommender
01 58
MINUTES SECONDS
Se7en (1995) Official Trailer
Indiana Jones and the Last
1983 The Return of the Jedi
Starship Troopers - Movie Tr
James Bond 007: Golden Eye (1995) - Official T
Next Page

Welcome, this is task 2

Mind the timer.

Seven (Se7en) (1995)
★★★★☆
A film about two homicide detectives' desperate hunt for a serial killer who justifies his crimes as absolution for the world's ignorance of the Seven Deadly Sins. The movie takes us from the tortured remains of one victim to the next as the sociopat ...More

Indiana Jones and the Last Crusade (1989)
★★★☆☆
Indiana Jones, famed adventurer and archaeologist acquires a diary that holds clues and a map with no names to find the mysterious Holy Grail- which was sent from his father, Dr. Henry Jones, in Italy. Upon hearing from a private collector, Walter Do ...More

Contact (1997)
★★★★★
Astronomer Dr. Ellie Arroway has long been interested in contact to faraway lands, a love fostered in her childhood by her father, Ted Arroway, who passed away when she was nine years old leaving her then orphaned. Her current work in monitoring for ...More

Starship Troopers (1997)
★★★★☆
In the distant future high school kids are encouraged to become citizens by joining the military. What they don't know is that they'll soon be engaged in a full scale war against a planet of alien insects. The fight is on to ensure the safety of huma ...More

GoldenEye (1995)
When a deadly satellite weapon system falls into the wrong hands, only Agent 007 can save the world from certain disaster. Armed with his license to kill, Bond races to Russia in search of the stolen access codes for "Goldeneye," an awesome space wea ...More
☆☆☆☆☆

Figure 7 : A snapshot of the web page of explicit method

The uncertain movie set is demonstrated in Figure 8. We chose top-50 uncertain users, each of them is randomly assigned 5 movies out of 48 new movies. That is, the uncertain set contains 250 elements which require workers to rate. To evaluate each worker's credibility, we implemented a quality control mechanism, which distributed five movies with known ratings into the 30 movies equally. After a worker rates six movies, his/her score is calculated with equation (14). The perfect score of 30 movies is 1,000 and a bonus is given to the worker if the score exceeds 750.

Subsequently, these gathered ratings were firstly aggregated according to workers' scores by using equation (15). Then with these aggregated ratings and the 300 * 300 user-similarity matrix, we can predict the remaing ratings in the 300 * 48 utility matrix.

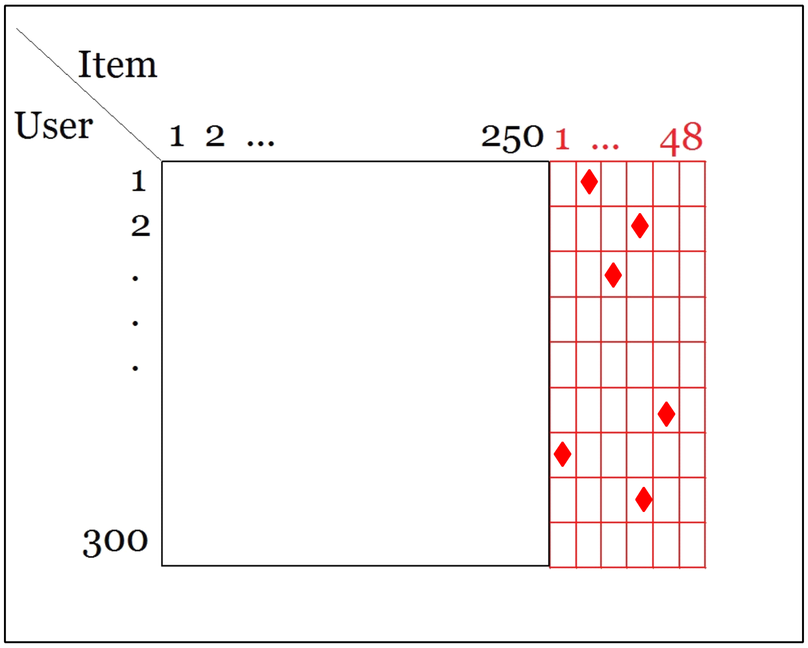
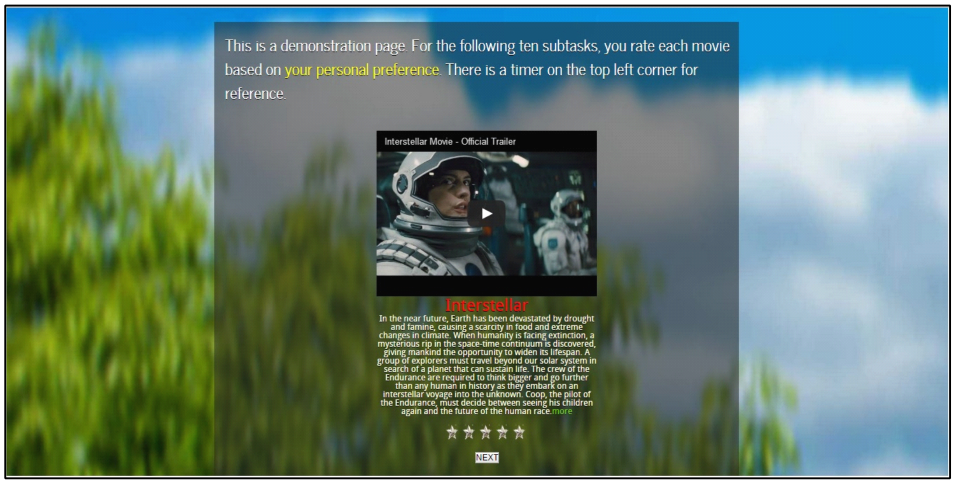


Figure 8. The red part is the new data set and the red diamond denotes the rating we want workers to rate.

In addition to the web page introduced above, a second web page for the implicit method was built as a comparison. The snapshot of the second web page is demonstrated in Figure 9, where each worker is going to rate the movie according to his/her personal preference. There are ten sub-tasks for each worker. Each page contains one movie which is selected randomly from 98 movies. The 98 movies are comprised of the new 48 movies and 50 movies which are the top-50 uncertain movies in the 250 movies. The top-50 uncertain movies were chosen by using the same idea depicted in section 3.3, that is, 250 movies were clustered and ordered in a decreasing entropy, then the heading 50 movies were the most uncertain movies. The uncertain data set is depicted in Figure 10.



This is a demonstration page. For the following ten subtasks, you rate each movie based on your personal preference. There is a timer on the top left corner for reference.

Interstellar

In the near future, Earth has been devastated by drought and famine, causing a scarcity in food and extreme changes in climate. When humanity is facing extinction, a mysterious rip in the space-time continuum is discovered, giving mankind the opportunity to widen its lifespan. A group of explorers must travel beyond our solar system in search of a planet that can sustain life. The crew of the Endurance are required to think bigger and go further than any human in history as they embark on an interstellar voyage into the unknown. Coop, the pilot of the Endurance, must decide between seeing his children again and the future of the human race.more

Figure 9. The snapshot of the second web page.

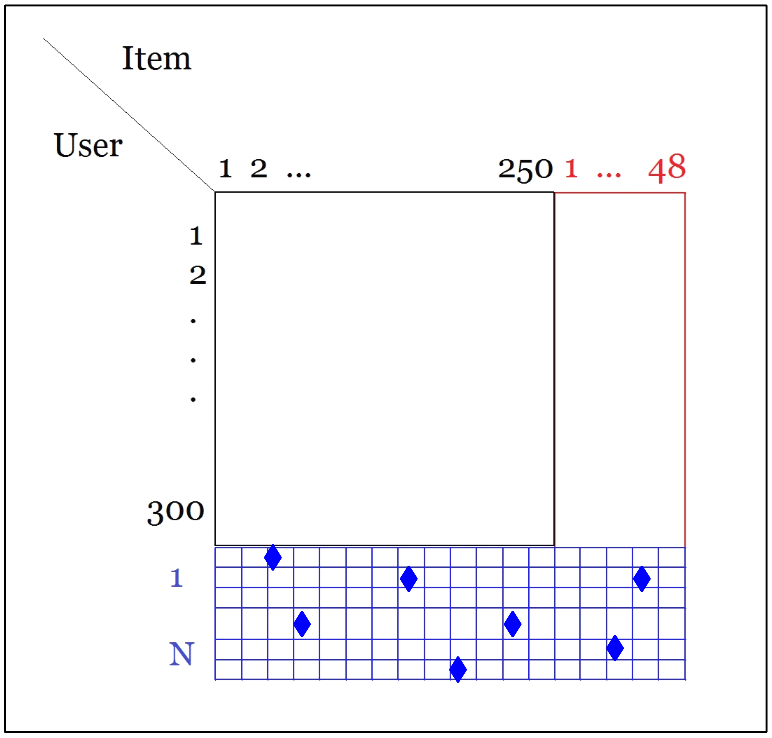


Figure 10. The red part is the new movie. The blue part is the added ratings from workers. The blue diamonds denote the ratings which are given by workers.

As in the explicit method, we implemented a quality control method in the implicit method as well. The quality control mechanism is a combination of total eclipsed time and a score found on the similarity among workers and the original data set. The workers who spent less time on rating the movies will be given less weighting while aggregating ratings. And the higher score represents a higher weighting in the aggregating process. The scoring mechanism is expressed in the following equation,

$$score = 100 + 100 * (\bar{x}_i - x_i)\,, \tag{16}$$

where
$$\bar{x}_i = \frac{\#(rating\_i\_movie\_l)}{\#(rating\_movie\_l)}, \tag{17}$$

and
$$x_i = \frac{\#(rating\_i\_dataset)}{\#(rating\_dataset)} \tag{18}$$

That is, $\bar{x}_i$ is the fraction of the number of rating i of movie l over the number of ratings of movie l. x_i is the fraction of the number of rating i in the dataset over the number of ratings in the dataset. The aggregated ratings were calculated by using equation (15), and were used to predict ratings with the assistance of (300+N) * (300 + N) user-similarity matrix, where N denoting the number of effective workers involved in the experiment.

4.4. LCE Method

All of the works demanded in the second comparison method, LEC, was the formation of the word frequency vectors. By adopting the idea of information retrieval, we firstly extracted the key words from the descriptions of the 248 movies and formed a dictionary. Then, each movie was expressed as a vector representing the word frequency in the dictionary. In Figure (11), for example, movie 248 has word indexed1 twice and word indexed 3610 once. Once the vectors were completed, the performance measurement of LCE on the vectors could be done by the implementation code releasing by the author.

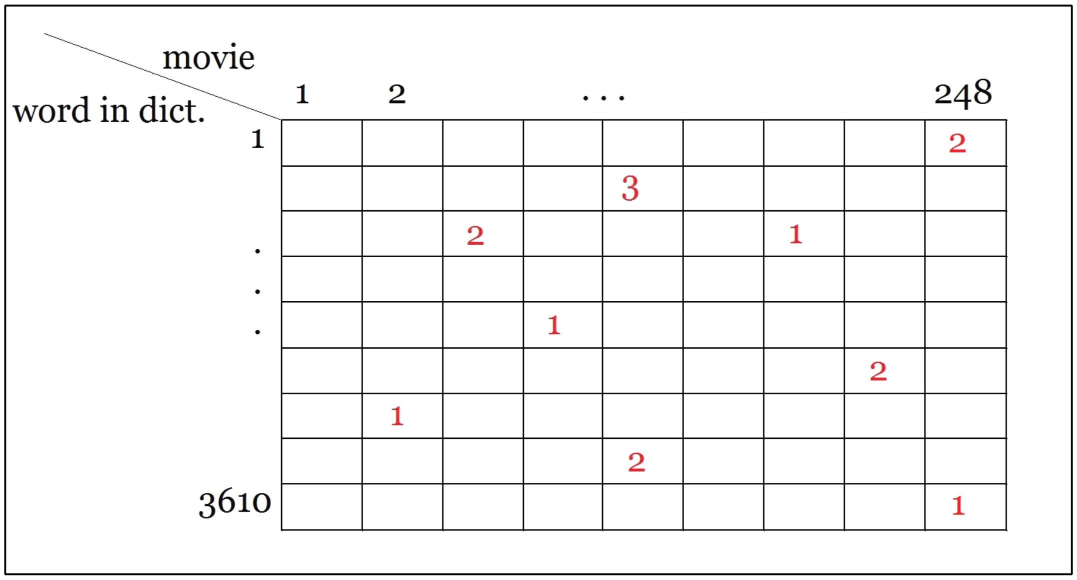


Figure 11. Each column is a vector representing the word frequency of the movie in the dictionary.

4.5. Results

The results of RMSE are shown in Figure 12 and MSE in Figure 13. And Figure 14 demonstrates the NDCG of three methods.

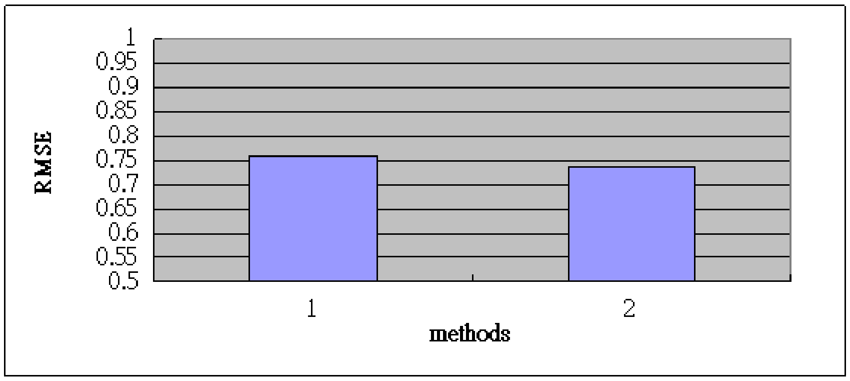


Figure 12. The left column is RMSE of explicit method and the right is of implicit method.

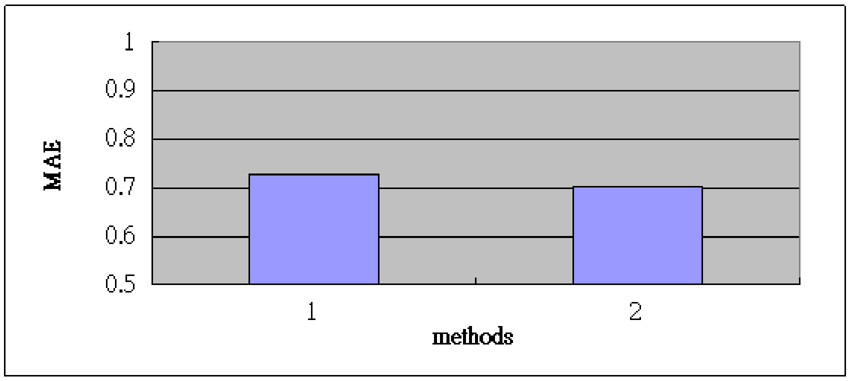


Figure 13. The left column is the MAE of explicit method and the right is of implicit

method.

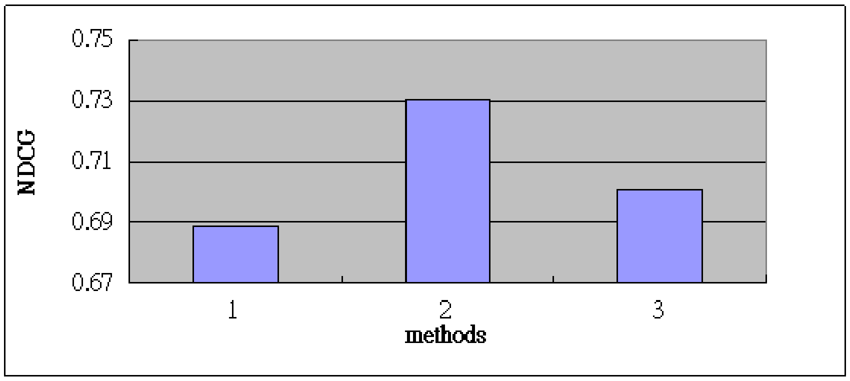


Figure 14. The left column is the NDCG of explicit method , the middle is of implicit method

and the right is of LCE

5. Conclusions and Future Works

In this report, the technique is proposed to alleviate the cold start problem. We adopted the idea of active learning, chose most uncertain elements, gathered ratings via a built web page, filtered out sloppy workers or spammers with a quality control mechanism. In the prior report, the proposed method was demonstrated to be an effective way and by the help of crowdsoucring could indeed relief the issue caused by the cold start problem.

From the experimental results, however, the degree of improving performance is not as good as the other two methods even though the gap is not very big. The problems might arise from the situations that workers lose patience more easily and are eager to finish the whole task while rating the 30 subtasks in the explicit method than rating only 10 subtasks in the implicit method. The average eclipsed time workers spent in finishing the tasks in explicit method is 10 to 20 minutes while it is 10 minutes in implicit methods. That is, workers spent less time for the last 20 subtasks. So the precision of the last 20 subtasks is not as high as the first 10 subtasks.

Besides, in implicit method, workers are required to rate movies based on their personal preferences, which in some way is easier than to guess other people's inclinations. Even there is a quality control technique in the explicit method, the hardness of predicting human remains a trouble.

Most of the feedbacks from the worker are positive. There are some complaints and suggestions. For example, 10 subtasks for some workers are excessive, let alone 30 subtaks. Hence, in the future the number of subtasks should be restricted to a reasonable number. Besides, some workers suggest that it is update instead of antique movie data set we should use. The only problem, however, is the difficulty to find a data set of latest movies.

In addition to a movie data set, other kinds of data set, such as music, books and news, could also be tested in the experiments since algorithm may have good performance on some data sets and bad on others owing to different intrinsic characteristics data set might possess.

6. References

[1] Wikipedia, *Recommender system* [Online]. Available :

http://en.wikipedia.org/wiki/Recommender_system

[2] J. D. Ullman, *Recommendation Systems* [Online]. Available :

http://infolab.stanford.edu/~ullman/mmds/ch9.pdf

[3] L.Candillier, F.Meyer, and M. Boulle, "Comparing State-of-the-Art Collaborative Filtering Systems" in *5th International Conference, MLDM 2007*, Vol.4571, Leipzig, Germany, July 18-20, 2007, pp 548-562

[4] X. Su and T.M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques" in *Advances in Artificial Intelligence*, Vol. 2009, New York, USA, January 2009

[5] J. Lee, M. Sun, and G. Lebanon, *A Comparative Study of Collaborative Filtering Algorithms* [Online]. Available : http://arxiv.org/pdf/1205.3193.pdf

[6] J. Lee et al., "Alleviating the Sparsity in Collaborative Filtering using Crowdsourcing" in CrowdRec 2013: Workshop on Crowdsourcing and Human Computation for Recommender Systems.

[7] J. Zhu et al., "Active Learning With Sampling by Uncertainty and Density for Data Annotations" in IEEE Transactions on Audio, Speech, and Language Processing, Vol. 18, NO. 6, August 2010, pp1323-1331

[8] C.H. Chiu,"Bootstrapping Recommender Systems with the Crowdsourcing" Semester Project Report, June 2014, EPFL

[9] M. Saveski and A. Mantrach, "Item Cold-Start Recommendations : Learning Local Collective Embeddings" in Proceedings of the 8th ACM Conference on Recommender systems, October 2014, pp89-96

[10] Wikipedia, *Normalized Discounted Cumulative Gain* [Online]. Available :

http://en.wikipedia.org/wiki/Discounted_cumulative_gain#Normalized_DCG