

Enhanced Prediction Algorithm for Item-Based Collaborative Filtering Recommendation

Heung-Nam Kim¹, Ae-Ttie Ji¹, and Geun-Sik Jo²

¹ Intelligent E-Commerce Systems Laboratory,
Department of Computer Science & Information Engineering, Inha University
{nami, aerry13}@eslab.inha.ac.kr

² School of Computer Science & Engineering, Inha University,
253 Yonghyun-dong, Incheon, Korea 402-751
gsjo@inha.ac.kr

Abstract. As the Internet infrastructure has been developed, a substantial number of diverse effective applications have attempted to achieve the full potential offered by the infrastructure. Collaborative Filtering recommender system, one of the most representative systems for personalized recommendations in E-commerce on the Web, is a system assisting users in easily finding the useful information. But traditional collaborative filtering suffers some weaknesses with quality evaluation: the sparsity of the data, scalability, unreliable users. To address these issues, we have presented a novel approach to provide the enhanced prediction quality supporting the protection against the influence of malicious ratings, or unreliable users. In addition, an item-based approach is employed to overcome the sparsity and scalability problems. The proposed method combines the item confidence and item similarity, collectively called *item trust* using this value for online predictions. The experimental evaluation on *MovieLens* datasets shows that the proposed method brings significant advantages both in terms of improving the prediction quality and in dealing with malicious datasets.

1 Introduction

With the explosive growth of the Internet, recommender systems have been issued as a solution for the problem of information overload. Recommender systems intend to assist users in finding the information most relevant to their preferences [11]. One of the most successful technologies in recommender systems is Collaborative Filtering (CF) and numerous commercial systems apply this technology to serve recommendations to their customers. The traditional the task in CF is to predict the utility of a certain item for the target user (often called active user) from the user's previous preference or the opinion of other similar users, and make appropriate recommendations [2]. However, despite the success and popularity, traditional CF encounters several limitations, namely sparsity, scalability, cold start, and the malicious ratings problem. And a number of researches have been proposed and challenged to address these problems related to collaborative filtering [2, 5, 6, 7, 10, 13].

In this paper, the techniques of CF are exploited, in generating enhanced predictions derived from explicit ratings. The main objective of this research is to develop a robust approach that provides high-quality predictions and recommendations even

when some ratings of users are unreliable. In addition, an item-based approach is employed to overcome the sparsity and scalability problems [2]. The proposed approach first determines the similarities between the items and subsequently identifies the confidence of the items, indicating the accuracy of the past predictions. Furthermore, this paper presents a method of combining the item confidence and item similarity, collectively called *item trust* using this value for online predictions and recommendations. The subsequent sections of this paper are organized as follows: The next section contains a brief overview of some related researches. In section 3, the approach for CF, based on *item trust*, is described. The performance evaluation is presented in section 4. Finally, we remark the conclusions and future works.

2 Background and Related Work

This section briefly explains previous researches related to CF-based recommender systems, which can be divided into two classes: Memory-based CF and Model-based CF [1]. Since the first system to generate automated recommendations, the GroupLens [3], was proposed, the user-based approach has been the most widely used for recommendation systems. User-based CF uses a similarity measurement between neighbors and the target users to learn and predict the preference towards new items or unrated products regarding a target user. Though user-based CF algorithms tend to produce more accurate recommendations, they have some serious problems relating to the complexity of computing each recommendation as the number of users and items grow. In order to improve scalability and real-time performance in large applications, a variety of model-based recommendation techniques were developed [2, 3, 12]. Especially, a new class of Item-based CF, which is one of model-based approaches and this research focuses on, has been proposed. This approach provides item recommendations by first developing a model of user ratings. In comparison to user-based approaches, item-based CF is typically faster in terms of recommendation time, though the method may have an expensive learning or model building process [4]. Instead of computing the similarities between the users, item-based CF reviews a set of items the target user has rated and selects k most similar items, based on the similarities between the items. Sarwar et al. [2] evaluated various methods to compute similarity and approaches to limit the set of item-to-item similarities that must be considered. And Deshpande et al. [5] proposed Item-based top-N recommendation algorithms that are similar to previous item-based schemes. They separated the algorithms two distinct parts for building a model of item-to-item similarities and deriving the top-N recommendations using this pre-computed model. Despite effectiveness of item-base CF algorithms, they still have some weaknesses concerning data sparseness, cold start users and ratings of malicious users. Hence, a number of recent research efforts that focus on the use of trust concepts during the recommendation process [6, 7, 8]. In addition, distributed recommender systems have been proposed to deal with the existing weaknesses [7, 10, 12].

3 Collaborative Filtering Based on Item Trust

The proposed method is divided into two phases, an offline phase and an online phase. The offline phase is a building model phase, and the online phase is either a prediction or recommendation phase. Fig. 1 illustrates a brief overview of the system.

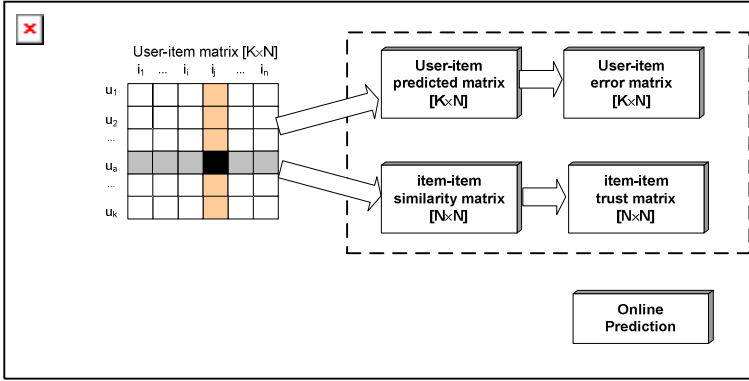


Fig. 1. Collaborative filtering recommendation based on item trust: item-based approach

3.1 Cosine-Based Similarity with Inverse Item Frequency

The most important task in CF recommendation is the similarity measurement because different measurements lead to different neighboring users or items, in turn, leading to different recommendations. From the item-based similarity viewpoint, there are several different methods of computing the similarity between items, such as correlation-based similarity, cosine similarity, and adjusted cosine similarity [2]. Initially, the methods that would be more accurate in the proposed system were examined. As a result, the cosine measures greater accuracy than the other measures (see Table 1).

In cosine similarity between items, two items are treated as two vectors in the space of users. In addition, we also consider the number of users' ratings for items as mentioned in [5]. Consider two users A and B, both of whom have co-rated item i and j , however user A rated just 5 items whereas user B rated 100 items. In this situation, user A, rating fewer items, is a relatively more reliable for the similarity of items i and j than user B rating lots of items. Therefore, the inverse user frequency as described in [1] for the proposed system is modified, namely *the inverse item frequency*.

In a system which users have co-rated items, *the inverse item frequency* can be applied to the cosine similarity technique. The similarity between two items, i and j is measured by equation (1).

$$Sim(i, j) = \frac{\sum_{u \in User} r_{u,i} \cdot r_{u,j} \cdot f_u^2}{\sqrt{\sum_{u \in User} r_{u,i}^2 \cdot f_u^2} \sqrt{\sum_{u \in User} r_{u,j}^2 \cdot f_u^2}} \tag{1}$$

where $User$ is a set of users who both rated i and j , $r_{u,i}$ is the rating of user u on item i , and $r_{u,j}$ is the rating of user u on item j . The inverse item frequency of user u , f_u is defined as $\log(n/n_u)$ where n_u is the number of items rated by user u and n is the total number of items in the database. If user u rated all items, then the value of f_u is 0. Likewise the inverse user frequency, the main idea of the inverse item frequency is that users rating lots of items present less contribution with regard to prediction, than users rating a smaller number of items.

3.2 Item Confidence for Computing Item Trust

Before describing the algorithms, some definitions of the matrices are introduced.

–**User-Item actual rating matrix.** If there is a list of k users $U=\{u_1, u_2, \dots, u_k\}$, a list n items $I=\{i_1, i_2, \dots, i_n\}$, and a mapping between user-item pairs, and the explicit ratings, $k \times n$ user-item data can be represented as a rating matrix. This matrix is called a *User-Item actual rating matrix*, A . The matrix rows represent users, the columns represent items, and $A_{a,j}$ represents the rating of a user a on an item j . Some of the entries are not filled, as there are items not rated by some users.

–**User-Item predicted rating matrix.** This is a matrix of users and items that have the predicted values for users on items. From a matrix A , the system can predict $P_{a,i}$ for a given target item i which has already been rated by target user a . This matrix is called a *User-Item predicted rating matrix*, P . Likewise a matrix A , The matrix rows represent users, the columns represent items and the elements of the matrix P is a subset of the elements of a matrix A , $P \subseteq A$.

–**User-Item error matrix.** From the given set of actual and predicted rating pairs $\langle A_{a,j}, P_{a,j} \rangle$ for all the data in a matrix A and P , a *User-Item error matrix*, E , can be represented as absolute error, which can be computed by subtracting the predicted rating for users on items from the actual rating for users on items. The elements of A matrix E is also a subset of the elements of a matrix A , $E \subseteq A$.

For constructing a matrix E , firstly a user's rating should be predicted for an item which has already been rated. For the purpose of this, a user-based Resnick prediction measure can be modified, which was introduced by [3], to an item-based prediction measure, as presented in equation (2). The prediction for the target user a on item i , $P_{a,i}$, is obtained as the following:

$$P_{a,i} = \bar{A}_i + \frac{\sum_{j \in N(a)} (A_{a,j} - \bar{A}_j) \cdot \text{sim}(i, j)}{\sum_{j \in N(a)} |\text{sim}(i, j)|} \quad (2)$$

where $N(a)$ is the set of k most similar items which the user a rated and $A_{a,j}$ is the rating of the user a on item j . In addition, \bar{A}_i and \bar{A}_j refer to the average rating of the item i and j . $\text{sim}(i, j)$ represents the similarity between the items i and j , which is calculated as mentioned in equation (1).

Once the predictions for users on items are represented on a user-item predicted rating matrix, absolute error of each prediction can be computed for constructing a user-item error matrix. Given the set of actual and predicted rating pairs $\langle A_{u,j}, P_{u,j} \rangle$ for all data in the user-item matrices, an absolute error, $E_{u,j}$, is calculated as:

$$e_{u,j} = |A_{u,j} - P_{u,j}|$$

As a result of the error matrix, the confidence of an item, indicating the percentage of accurate predictions for an item, is computed from each column in the user-item error matrix and is defined as the following equation (3).

$$confidence(j) = \sum_{u \in U} \frac{|E_{u,j} \cap E_{u,j}^r|}{|E_{u,j}|} \tag{3}$$

where $E_{u,j}$ is a set of errors predicted for user u on item j and $E_{u,j}^r$ is the set of errors for which an absolute error of $e_{u,j}$ is within a predefined ϵ ($e_{u,j} < \epsilon$). U is the set of users rating item j . For example, given item j , if a hundred errors have been computed for an item j and eighty of these predictions are accurate, the confidence of item j , $confidence(j)$, is 0.8.

3.3 Prediction Based on Item Trust

As mentioned previously, the item-based CF approach builds a model of item similarity, which can be achieved offline, prior to online prediction or recommendation. Since most of tasks can be conducted in the offline phase, this approach can result in fast online performance. In addition, this assists in solving the sparsity and scalability problems [2, 5]. The proposed method also provides another advantage, the ability to protect the influence of malicious ratings.

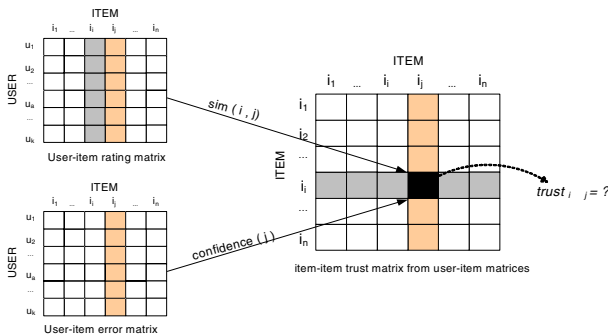


Fig. 2. The item-item matrix for a pair of items trust from the user-item matrices

In order to support fast online predictions, the trust value between two items is calculated in offline, namely *item-item trust matrix*. Fig. 2 illustrates the process of the item-item trust matrix construction from the user-item matrices.

–Item-item trust matrix. The item trust model can be represented as a matrix, T , in which rows and columns are both items. An entire $n \times n$ item-item trust matrix can be filled in, given by the $k \times n$ user-item matrices, A and E using equation (4)

$$trust^{\beta}_{i \rightarrow j} = \frac{(\beta^2 + 1) \cdot sim(i, j) \cdot confidence(j)}{\beta^2 \cdot sim(i, j) + confidence(j)} \tag{4}$$

where parameter β is specified for adjusting the relative weighting between the similarity of items and the confidence of an item. If $\beta=0$ then $trust^{\beta}_{i \rightarrow j}$ just takes $sim(i, j)$ into account whereas if $\beta=+\infty$ then $trust^{\beta}_{i \rightarrow j}$ just coincides with $confidence(j)$. When

a value of $\beta = 1$ is used, the equal importance to $sim(i,j)$ and $confidence(j)$ is considered. The trust value between a pair of items is in the range of $[0, 1]$ and is not symmetric ($trust^{\beta}_{i \rightarrow j} \neq trust^{\beta}_{j \rightarrow i}$). The appropriate value for β is selected by performing experimental analysis.

The most important task in a CF is to generate the prediction, attempting to guess the rating that a user would provide for an item [2]. In order to compute the predicted rating of target user a for the target item i , the item-based Resnick prediction measure discussed in section 3.2 is used. However, instead of using item similarity, $sim(i,j)$, the prediction algorithm in the online phase, uses the item trust value, $trust^{\beta}_{i \rightarrow j}$ as defined in equation (5).

$$P_{a,i} = \bar{A}_i + \frac{\sum_{j \in N(a)} (A_{a,j} - \bar{A}_j) \cdot trust^{\beta}_{i \rightarrow j}}{\sum_{j \in N(a)} trust^{\beta}_{i \rightarrow j}} \quad (5)$$

where $N(a)$ is the set of k most similar items which the user a rated and $A_{a,j}$ is the rating of the user a on item j . In addition, \bar{A}_i and \bar{A}_j refers to the average rating of the items i and j .

4 Experimental Evaluation

In this section, experimental results of the proposed method are presented. In order to compare the performance of the proposed method, user-based and item-based CF recommendation systems were implemented. All experiments were carried out on a Pentium IV 3.0GHz with 1GB RAM, running MS-window 2003 server. In addition, the recommendation system for the web was implemented using MySQL 4.0 and PHP 4.4 on an Apache 1.3 environment.

4.1 Data Set and Evaluation Metric

The experimental data comes from *MovieLens* which is a web-based research recommendation system (www.movielens.org). The data set contains 100,000 ratings of 1682 movies rated by 943 users (943 rows and 1682 columns of a *user-item matrix* A). These ratings were divided into two groups: 80% of the data (80,000 ratings) was used as a training set and 20% of the data (20,000 ratings) was used as a test set. Prior to evaluating the accuracy of the proposed method, a *user-item error matrix* E should first be constructed. Therefore, the training data set was further subdivided into training and testing portions, a matrix E was generated using a 5-fold cross validation scheme. After this process, a model (an *item-item trust matrix* T) for evaluating the method was created.

In order to measure the accuracy of the predictions, *mean absolute error* (MAE), which was widely used for the statistical accuracy measurements in the diverse algorithms [1, 2, 7] was adopted. The mean absolute error for user u is defined as:

$$MAUE(u) = \frac{\sum_{i \in I_u} |A_{u,i} - P_{u,i}|}{|I_u|}$$

where I_u is a item list of user u and $\langle A_{u,i}, P_{u,i} \rangle$ is the actual/predicted rating pairs of user u in the test data. Finally, the MAE of all users in the test set is computed as:

$$MAE = \frac{\sum_{u=1}^k MAUE(u)}{K}$$

4.2 Parameter Tuning Experiments

Prior to running the main experiment, the sensitivity of the two parameters: *item-item similarity* and β value, were first determined. In determining the sensitivity of these parameters, the training data set was focused on, which was further divided into two portions, 80% training and 20% testing. For parameter evaluation experiments, the full model size was used for model building, and $k=30$ was selected meaning the number of most similar items, for prediction generation.

Comparison of Similarity Algorithms. Prior to evaluating the item trust-based prediction method, a *user-item predicted matrix*, P , for calculating item confidence which is closely connected with the similarity algorithm, should first be built up. Thereby, we implemented diverse similarity algorithms such as correlation-based similarity, cosine-based similarity, adjusted cosine similarity as described in [2] and correlation-based similarity with inverse item frequency (*correlation+iif*) as described in [1]. And we compared them with cosine-based similarity with inverse item frequency (*cosine+iif*) as described in Section 3.1. For each similarity algorithms, the item-based Resnick measurement was used to generate the prediction. As seen from the results of Table 1, the prediction with the *cosine+iif* algorithm was generated, the prediction quality is improved, when compared to the other algorithms. Therefore, the cosine similarity with inverse item frequency is taken up in subsequent of experiments.

Table 1. Comparison of the prediction quality achieved by five different similarity measures

	cosine	cosine + iif	correlation	correlation + iif	adjusted cosine
MAE	0.74919	0.74248	0.75496	0.75242	0.76408

Sensitivity of β Value for Item Trust. As stated in Section 3.3, β is the parameter used for adjusting the relative weighting, where the similarity of items and the confidence of an item are important in the generation of an item trust. From the previous experiment, the error threshold ϵ for calculating the item confidence was set to be MAE of 0.742. Fig. 3(1) presents a variation in average MAE, by changing the β value. As a result, it can be observed that the quality of prediction improves as the β value is increased from 0 to 2, after 2, the curve tends to become flat. When β is set to infinity, the curve of the graph tends to rise. Hence, $\beta=2.5$ is selected as an optimal value for computing the item trust.

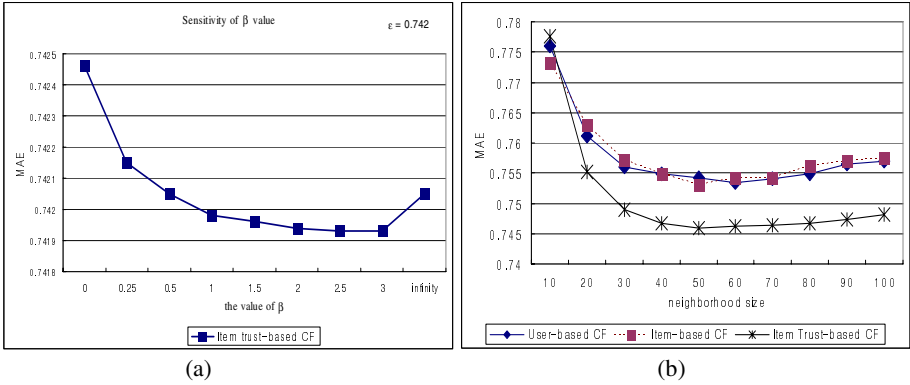


Fig. 3. Sensitivity of parameter β for the item trust (a) and Comparison of prediction quality of user-based CF, item-based CF and Item trust-based CF (b)

4.3 Performance Evaluation

The performance evaluation is divided into two dimensions. The quality of the prediction based on item trust is first evaluated, and then the robustness of the prediction to the malicious ratings problem is evaluated. Once the optimal values of the parameters are obtained, the prediction quality of the proposed method is evaluated in comparison with the traditional user-based and item-based schemes.

Quality of the Prediction. The model size has significant impact on the prediction quality in a model-based approach [5]. However, the experimental result of the previous research in [2] demonstrates that a full model size obtains superior prediction quality than a small model size, although the time cost for building the model is greater. Therefore, in the prediction quality experiment, the full model size was used, and the number of item neighbors to be used for the online prediction generation was changed. The experimental results are depicted in Fig 3(b). It can be observed from the graph that the size of the neighborhood affects the prediction quality and the three methods demonstrate similar types of charts. The model-based approaches (item-based CF and item trust-based CF) elevate the prediction quality as the neighborhood size increases from 10 to 50, after this value, the quality decreased slightly. Likewise, a user-based CF improved until a neighborhood size of 60. The result demonstrates that, at all neighborhood size levels, except for a neighborhood size of 10, the proposed algorithm provides more accurate predictions than the traditional user-based and item-based algorithm. For example, when neighborhood size is 50, item trust-based CF obtains an MAE of 0.745, which is the best prediction quality, whereas item-based and user-based methods demonstrate an MAE of 0.753 and 0.754 respectively. However, the classic item-based scheme provides better quality in the event of a high sparsity level (neighborhood size =10).

Robustness on Malicious Ratings. For evaluating the robustness on fraud ratings, 10%, 20%, 30%, and 40% of malicious ratings were included in the training set, and the experiments were ran again using the full model size and a neighborhood size of 30. Table 2 summarizes the result of the experiment. In general, with the growth of

malicious ratings, the prediction quality decreases, as can be seen from Table 1. However, the item trust-based CF shows the improved performance on all occasions, compared to traditional user-based and item-based CF. As the percentage of fraud ratings in a training data set increases, efficient improvement in performance can be obtained. Although the prediction quality is improved slightly in the case of 10% ratings being malicious, in the case of 40% ratings being malicious the proposed method achieves 5% improvement, compared to the other methods, respectively. As a result, the item trust-based CF brings 12% degradation in terms of the four cases in average, compared to an original rating set (0% malicious ratings set) whereas the average degradation of robustness is 15% for the user-based CF and 14% for the item-based CF.

Table 2. Robustness of user-based CF, item-based CF and item trust-based CF on fraud rating

malicious ratings	User-based CF	Item-based CF	Item Trust-based CF
0 %	0.756	0.7572	0.7489
10 %	0.8042	0.8051	0.7954
20 %	0.8649	0.8601	0.8407
30 %	0.9542	0.9413	0.9184
40 %	1.0119	1.0059	0.9551

5 Conclusion and Future Work

Collaborative Filtering for Recommendations is a powerful technology for users to find information relevant to their needs. We have presented, in this paper, a novel approach to provide the enhanced prediction quality and to solve some of the limitation in traditional CF systems. And we propose a new method of building a model, namely *item-item trust matrix*, for CF-based recommender systems. The major advantage of the proposed approach is that it supports the protection against the influence of malicious ratings, or unreliable users. The experimental results demonstrate that the proposed method obtains significant advantages both in terms of improving the prediction quality and in dealing with malicious data sets as compared to traditional CF algorithms. However, there still remains a defect that the proposed method performs worse at a high sparsity level.

An ongoing area of current is a distributed recommender system [10, 12]. We are currently extending our algorithm to a personalized recommendation in a peer-to-peer environment or a social network. Therefore, we will further study the impact of using trust values, such as *web of trust* [6], and the technique of trust propagations.

References

1. Breese, J.S., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence (1998) 43–52
2. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based Collaborative Filtering Recommendation Algorithms. In Proc. of the 10th Int. Conf. on World Wide Web (2001)

3. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews. In Proc. of the ACM Conf. on Computer supported Cooperative Work (1994) 175–186
4. Lemire, D., Maclachlan, A.: Slope One Predictors for Online Rating-Based Collaborative Filtering. In Proc. of SIAM Data Mining (2005)
5. Deshpande, M., Karypis, G.: Item-based top-N recommendation algorithms. ACM Transactions on Information Systems, Vol. 22 (2004) 143–177
6. Massa, P., Avesani, P.: Trust-aware Collaborative Filtering for Recommender Systems. In Proc. of Int. Conf. on Cooperative Information Systems (2004)
7. Papagelis, M., Plexousakis, D., Kutsuras, T.: Alleviation of the Sparsity Problem of Collaborative Filtering Using Trust Inferences. In Proc. of the 3rd Int. Conf. on Trust Management (2005) 224–239
8. O'Donovan, J., Smyth, B.: Trust in recommender systems. In Proc. of the 10th Int. Conf. on Intelligent user interfaces (2005) 167–174
9. Mobasher, B., Jin, X., Zhou, Y.: Semantically Enhanced Collaborative Filtering On the Web. Lecture Notes in Computer Science, Vol. 3209. Springer-Verlag, Berlin Heidelberg (2004) 57–76
10. Kim, H. J., Jung, J. J., Jo, G. S.: Conceptual Framework for Recommendation System based on Distributed User Ratings. In Proc. of the 2nd Int. Workshop on Grid and Cooperative Computing (2003)
11. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of recommendation algorithms for E-commerce. In Proc. of ACM'00 Conf. on Electronic Commerce (2000) 158–167
12. Miller, B. N., Konstan, J. A., Riedl, J.: PocketLens: Toward a personal recommender system. ACM Transactions on Information Systems, Vol. 22 (2004) 437–476
13. Schein, A. I., Popescul, A., Ungar, L. H.: Methods and Metrics for Cold-Start Recommendations. In Proc. of the 25th Int. ACM Conf. on Research and Development in Information Retrieval (2002)