Expectation-Maximization Collaborative Filtering with Explicit and Implicit Feedback

Bin Wang, Mohammadreza Rahimi, Dequan Zhou, and Xin Wang

Department of Geomatics Engineering, University of Calgary
2500 University Drive NW, Calgary, AB, Canada, T2N 1N4
{bw.wang,smrahimi,dzho,xcwang}@ucalgary.ca

Abstract. Collaborative Filtering (CF) is a popular strategy for recommender systems, which infers users' preferences typically using either explicit feedback (e.g., ratings) or implicit feedback (e.g., clicks). Explicit feedback is more accurate, but the quantity is not sufficient; whereas implicit feedback has an abundant quantity, but can be fairly inaccurate. In this paper, we propose a novel method, *Expectation-Maximization Collaborative Filtering* (EMCF), based on matrix factorization. The contributions of this paper include: first, we combine explicit and implicit feedback together in EMCF to infer users' preferences by learning latent factor vectors from matrix factorization; second, we observe four different cases of implicit feedback in terms of the distribution of latent factor vectors, and then propose different methods to estimate implicit feedback for different cases in EMCF; third, we develop an algorithm for EMCF to iteratively propagate the estimations of implicit feedback and update the latent factor vectors in order to fully utilize implicit feedback. We designed experiments to compare EMCF with other CF methods. The experimental results show that EMCF outperforms other methods by combining explicit and implicit feedback.

1 Introduction

In the modern digital world, consumers are overwhelmed by the huge amount of product choices offered by electronic retailers and content providers. Recommender systems, which analyze patterns of user interests in products in order to provide personalized recommendatons satisfying users' tastes, have recently attracted a great deal of attention from both academia and industry. *Collaborative filtering* (CF) [9], which analyzes relationships among users and items (i.e., products) in order to identify potential associations between users and items, is a popular strategy for recommender systems. Compared to *content filtering* [8], which is the other recommendation strategy that depends on profiles of users and/or items, CF has the advantage of being free of domain knowledge. Since CF relies only on the history of user behavior, it can address the issues in creating explicit profiles, which are difficult in many recommender system scenarios.

The history of user behavior for CF usually consists of user feedback, which generally refers to any form of user action on items that may convey the information about users' preferences of items. There are two kinds of user feedback:

explicit feedback and *implicit feedback*. Explicit feedback is often in the form of rating actions. For example, Amazon.com asks users to rate their purchased CDs and books on a scale of 1-5 stars. Since explicit feedback directly represents users' judgments on items in a granular way, it has been widely used in many traditional CF recommender systems [2] [9] [10]. However, explicit feedback requires users to perform extra rating actions, which may lead to inconvenience for the user. Given the overwhelming amount of items, it is burdensome for users to rate every item they like or dislike.

Implicit feedback, on the other hand, does not need additional rating actions. Implicit feedback generally refers to any user behavior that indirectly expresses user interests. For example, a news website may cache a user's clicking records when browsing news articles, in order to predict what kind of news the user may prefer. Other forms of implicit feedback include keyword searching, mouse movement, and even eye tracking. Since it is relatively easier to collect such user behaviors, implicit feedback attracts the interest of researchers who attempt to infer user preferences from the much larger amount of implicit feedback. However, implicit feedback is less accurate than explicit feedback. For example, a user may regret buying a product online after receiving the real product. It is difficult to determine whether a user likes a product only based on the purchase behavior, even though the user paid for the product.

Explicit feedback and implicit feedback are naturally complementary to each other. With explicit feedback, the quality is more reliable, but the quantity is limited. However, the quality of implicit feedback is less accurate, but there is an abundant quantity. Most of the existing works have solely considered either explicit feedback [3] or implicit feedback [4] [7] in recommender systems. While there are few works that have tried to unify explicit and implicit feedback, explicit feedback is just treated as a special kind of implicit feedback; and, the implicit feedback is simply normalized to a set of numeric rating values. Without carefully studying how to organically combine explicit and implicit feedback, we will not be able to further improve the performances of recommender systems.

In this paper, we propose a novel recommender method based on matrix factorization [5], called expectation-maximization collaborative filtering (EMCF). The first contribution of this paper is that we combine explicit and implicit feedback together, in which both explicit feedback and implicit feedback are fully utilized. The second contribution is that we observe different cases of implicit feedback and develop the corresponding solutions to estimate implicit feedback for different cases. The third contribution is that we design an expectation-maximization-styled algorithm in EMCF to update the estimations of implicit feedback and latent factor vectors.

Instead of treating explicit feedback as special implicit feedback, EMCF initializes a latent factor model with explicit feedback and then updates the latent factor vectors based on the explicit feedback ratings and the implicit feedback estimations. The key challenge in utilizing implicit feedback in matrix factorization is that implicit feedback does not have the numeric rating value. Instead of simply normalizing implicit feedback to a set of numeric rating values, EMCF estimates implicit feedback rating values based on the explicit feedback and available latent factor vectors.

We observe that the implicit feedback can be categorized into four cases, in terms of the distribution of latent factor vectors from the currently trained latent factor model. For different cases, EMCF has corresponding solutions, which are not only based on explicit feedback ratings and implicit feedback estimations, but are also based on the graph-based structure of explicit and implicit feedback.

We also observe that only part of implicit feedback ratings can be estimated based on the current situation of the model. Therefore, an expectation-maximization-styled algorithm is designed in EMCF to: 1) propagate current estimations of implicit feedback plus explicit feedback ratings towards the set of implicit feedback that have not yet been estimated, so that more implicit feedback estimations can be added into the model training; 2) Re-train the latent factor model based on all the available implicit feedback estimations and explicit feedback ratings and then update the latent factor vectors of users and items for further estimating. The algorithm not only fully utilizes implicit feedback with explicit feedback, but also prevents noisy implicit feedback from affecting the performance of the EMCF model.

Experiments have been conducted to compare the EMCF model with other popular models. The experimental results show that EMCF outperforms those models, especially when the percentage of explicit feedback is small.

The rest of the paper is organized as follows: in Section 2, a preliminary is given including the formalization, the background of CF, and a related method called co-rating; in Section 3, we present the observations of implicit feedback and propose the solutions to estimate implicit feedback for different cases; in Section 4, we describe the EMCF model and introduce the algorithm to train the model; in Section 5, the experiments make the comparisons between EMCF and other models; the conclusion and some future works are given in Section 6.

2 Preliminary

2.1 Formalization

In this paper, we use $U = \{u_1, u_2, \ldots, u_m\}$ to denote a set of m users and use $I = \{i_1, i_2, \ldots, i_n\}$ to denote a set of n items. The explicit feedback and implicit feedback are defined as the observable actions from U to I that may directly and indirectly reflect users' preferences of the items.

Explicit feedback is usually in the form of rating actions. A user rates items by assigning numeric rating values. The observed rating values are represented by a matrix $R \in \Re^{m \times n}$, in which each entry $r_{ui} \in \Re$ is used to denote the rating on item i given by user u. We use S^E to denote the set of explicit feedback, which consists of user-item-rating triples (u, i, r_{ui}).

Implicit feedback typically consists of various types of actions performed by users on items that can be automatically tracked by systems. In some related works [4] [6], implicit feedback is represented by a binary variable $b_{ui} \in \{0, 1\}$, in which 1 means user u performed some action on item i and 0 means u never touched i. In this paper, we assume that a user has no interest to an item if he/she never touched this item. We use S^I to denote the set of implicit feedback, which consists of user-item pairs (u, i) for which u has implicit feedback on i.

2.2 Collaborative Filtering

There are two primary types of CF methods: *nearest-neighbor methods* and *matrix factorization methods*. A brief introduction of these two types of methods is given in this section.

Nearest-neighbor Methods. Nearest-neighbor methods are prevalent in CF [2]. Generally, the procedures of nearest-neighbor methods follow a similar pattern: first calculate the similarity, which reflects distance, correlation, or weight, between two users or two items; then produce a prediction for the active user by taking the weighted average of all the ratings.

In terms of different focuses in the similarity calculation, there are two types of nearest-neighbor methods: *item-based methods* and *user-based methods*. As the name suggests, item-based methods calculate the similarities between items, try to find nearest-neighbor items for the target item, and then evaluate the active user's rating on the target item based on the ratings of its neighbors. Similarly, user-based methods first identify nearest-neighbor users for the target user by calculating the similarities between users, and then make predictions for the target user to unrated items based on neighbor users' ratings.

Matrix Factorization Methods. Matrix factorization is the other primary type of approaches for CF. Latent factor models, which try to explain the rating generation by vectors of latent factors inferred from the patterns of ratings, are typically used in matrix factorization methods. In a sense, such latent factors correspond to the dimensions in a latent space in which the profiles of both users and items can be characterized. Koren et al. [5] gave some examples to interpret latent factors: if the items are movies, the latent factors may measure obvious dimensions such as comedy versus drama, less well-defined dimensions such as depth of character development or quirkiness, or completely un-interpretable dimensions; for users, each latent factor may measure how a user scores the corresponding movie factor.

Matrix factorization methods map both users and items to a joint latent factor space, so that each user is modeled by a user latent factor vector and each item is modeled by an item latent factor vector. The rating for a user-item pair is modeled as an inner product of this user's latent factor vector and this item's latent factor vector.

2.3 Co-rating

Liu et al. [6] developed a matrix factorization model called *co-rating*, which tries to unify explicit and implicit feedback. Co-rating treats explicit feedback as a special kind of implicit feedback, so that the entire set of explicit and implicit feedback can be used simultaneously during the model training. For solving the challenge of implicit feedback ratings having only binary values instead of numeric values, co-rating normalizes the rating values of explicit feedback and the binary values of implicit feedback into a range of $[0, 1]$. With the co-rating method, latent factor vectors are learned by solving an objective function in the matrix factorization model trained with explicit and implicit feedback. The co-rating's objective function is different from the one normally used in other matrix factorization methods [1] [5]; an extra weighted term has been added, which aims at controlling the loss when treating explicit feedback as implicit feedback.

3 Explicit and Implicit Feedback

3.1 Matrix Factorization with Explicit Feedback

In this paper, we fully utilize explicit feedback and implicit feedback together to train a latent factor model by the matrix factorization method. In the matrix factorization method, each item i is associated with a latent factor vector $q_i \in \Re^k$, and each user u is associated with a latent factor vector $p_u \in \Re^k$, where k is the number of latent factors. For a given item i, the elements of q_i measure the extent to which the item possesses those factors. For a given user u, the elements of p_u measure the extent of u's interest in items according to the corresponding factors. The rating value r_{ui} is approximated by the dot product $q_i^\intercal p_u$. Therefore, the rating matrix R is approximated by the product of the user latent factor matrix $M_U \in \Re^{k \times m}$ and the item latent factor matrix $M_I \in \Re^{k \times n}$ as

$$R \approx M_U^\intercal \cdot M_I. \tag{1}$$

The matrix factorization method learns the individual latent factor vectors in M_U and M_I by solving

$$argmin_{q^*,p^*} \sum_{(u,i,r_{ui}) \in S^T} (r_{ui} - q_i^\intercal p_u) + \lambda(\|q_i\|^2 + \|p_u\|^2), \tag{2}$$

where S^T is the training set including the known rating values, the term $r_{ui} - q_i^\intercal p_u$ is the estimation of the goodness of the rating approximation, $\|q_i\|^2 + \|p_u\|^2$ is the regularization term to avoid model overfitting, and the constant λ is to control the extent of regularization.

Before utilizing implicit feedback, we first use the set of explicit feedback S^E to initially train the model, so that $S^T \leftarrow S^E$ at this stage. There are two approaches to solve Equation 2: *Stochastic Gradient Descent* (SGD) [5] and *Alternating Least Squares* (ALS) [11].

SGD is a popular approach that is easy to implement and has a relatively fast running time. In SGD, the algorithm loops through all ratings in S^T. For each triple (u, i, r_{ui}), the algorithm computes the associated prediction error:

$$e_{ui} = r_{ui} - q_i^\intercal p_u. \tag{3}$$

The algorithm then modifies the parameters by a magnitude proportional to γ in the opposite direction of the gradient as:

$$q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i) \tag{4}$$

$$p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u) \tag{5}$$

ALS is a different style of algorithm for learning the latent factor vectors. In ALS, one of the unknown latent factor vectors is fixed in order to learn the other vector; and, the latter vector is then fixed to learn the former vector. The procedure is repeated until convergence is reached. With ALS, the optimization of Equation 2 becomes quadratic and can be optimally solved. Although ALS is slower and more complicated than SGD, it is usually favorable when parallelization is needed. Due the space limitation, we are not going to give the details of ALS here.

3.2 Enhance Explicit Feedback with Implicit Feedback

Matrix factorization methods require numeric rating values to learn the latent factor vectors. Although explicit feedback satisfies this requirement, the amount of explicit feedback ratings is usually not sufficient to train an accurate model. On the other hand, the amount of implicit feedback is much larger than the amount of explicit feedback due to the lack of additional rating actions. If there is a way to assign a meaningful numeric estimation to implicit feedback, it can be used to enhance the model trained only based on explicit feedback ratings.

After initializing the model based on explicit feedback ratings, latent factor vectors can be attained for users and items that are included in S^E. For each user-item pair (u, i) in S^I, there are four possible cases:

- **Case 1**: Both u and i have been assigned latent factor vectors p_u and q_i, respectively, because u and i are also included in S^E.
- **Case 2**: u has been assigned latent factor vector p_u because u is also included in S^E, but i has no latent factor vector since i is not included in S^E.
- **Case 3**: i has been assigned latent factor vector q_i because i is also included in S^E, but u has no latent factor vector since u is not included in S^E.
- **Case 4**: Both u and i have no latent factor vectors, because u and i are not included in S^E.

These four cases are demonstrated in Figure 1.

For Case 1, the target implicit feedback can be straightforwardly estimated using the latent factor vectors of u and i. The estimation $\hat{r}^I_{ui}$ can be computed as:

$$\hat{r}^I_{ui} = q_i^\intercal p_u. \tag{6}$$

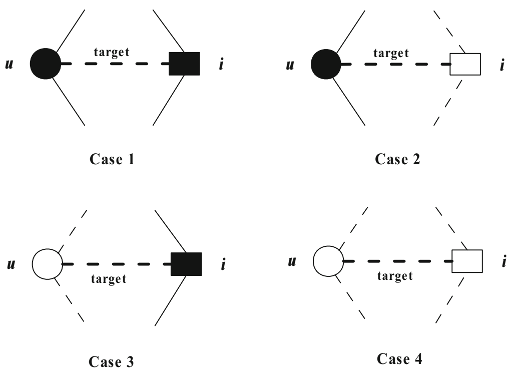

Fig. 1. Four cases of implicit feedback. The black circles and black squares are used to represent users and items, respectively, on which the latent factor vectors have been assigned, and the white circles and white squares are used to represent the users and items, on which there is no latent factor vector assigned yet. The solid lines represent the explicit feedback, and the dash lines represent the implicit feedback. The thick dash lines are the targets of implicit feedback that we will estimate based on the current situation.

For Case 2, the target implicit feedback cannot be directly estimated using latent factor vectors due to the lack of q_i. We use an item-based CF method to estimate it. First, the similarity $sim(i,j)$ between item i and item j is calculated using Jaccard Similarity Coefficient as:

$$sim(i,j) = \frac{|A_i \bigcap A_j|}{|A_i \bigcup A_j|}, \tag{7}$$

where A_i and A_j are the set of users who have either explicit or implicit feedback actions on i and j respectively. Next, we look for the set of neighbor items N_i of item i. In N_i, each neighbor item j has to satisfy the conditions as: 1) the similarity $sim(i,j)$ is larger than a pre-defined threshold; 2) a latent factor vector has already been assigned on j. Then, the estimation $\hat{r}^I_{ui}$ for the target implicit feedback can be computed as:

$$\hat{r}^I_{ui} = \frac{\sum_{j \in N_i} sim(i,j) q_j^\intercal p_u}{\sum_{j \in N_i} sim(i,j)}. \tag{8}$$

Similarly for Case 3, the similarity $sim(u,v)$ between user u and user v is also calculated by Jaccard Similarity Coefficient as:

$$sim(u,v) = \frac{|A_u \bigcap A_v|}{|A_u \bigcup A_v|}, \tag{9}$$

where A_u and A_v are the set of items on which u and v have either explicit or implicit feedback actions respectively. The set of neighbor users N_u for user

u is found, in which each neighbor user v has a value $sim(u,v)$ larger than a threshold and has an assigned latent factor vector. The estimation $\hat{r}^I_{ui}$ for the target implicit feedback can be computed using a user-based CF method as

$$\hat{r}^I_{ui} = \frac{\sum_{v \in N_u} sim(u,v) q_i^\intercal p_v}{\sum_{v \in N_u} sim(u,v)}. \tag{10}$$

For Case 4, there is no sufficient information to estimate the target implicit feedback based on the current situation.

4 Expectation-Maximization Collaborative Filtering

The user neighbor set N_u and the item neighbor set N_i may have no eligible members. Although some user or item is similar enough with the target user or item, they still can not become eligible neighbors due to lack of latent factor vectors. On the other hand, estimations from Equation 8 and 10 are based on currently learned latent factor vectors, and latent factor vectors need to be updated based on the updated training set, in which new estimations will be added in. To address these issues, we design the *Expectation-maximization Collaborative Filtering* (EMCF) algorithm. The basic idea is to iteratively propagate the available implicit feedback estimations, plus the explicit feedback, towards the unavailable implicit feedback, in order to make possible the estimations on such implicit feedback.

If we treat CF model as the objective and treat latent factor matrices as estimated parameters, we can map the classic EM into our problem scenario. Our goal is to build CF model that uses the matrix factorization method. The model depends on both user latent factor vectors and item latent factor vectors. The parameters that we use to estimate latent factor vectors are explicit feedback ratings and implicit feedback estimations. The two steps are defined as the following:

– **E Step:** Train the collaborative filtering model using all the explicit feedback ratings and currently available estimations of implicit feedback.
– **M Step:** Estimate the implicit feedback based on latent factor vectors that are output from the CF model trained in the previous E Step.

The EMCF algorithm is an iterative procedure. We begin by using explicit feedback ratings to initialize the CF model with the matrix factorization method, which is introduced in Section 3.2. The set of implicit feedback is then categorized based on four cases (Cases 1-4 above), in terms of whether the involved users and item have latent factor vectors or not. Following the estimation methods introduced in Section 3.3, the implicit feedback ratings are estimated if they are eligible. The estimated implicit feedback ratings are put together with explicit feedback ratings to train the EMCF model again. Thus, for the user and/or the item involved in the target implicit feedback, new latent factor vectors are assigned if the user and/or the item do not yet have them. For other

```
Algorithm EMCF
Input: Explicit feedback set $S^E$, implicit feedback set $S^I$, user set $U$, and item set $I$.

Output: User latent factor matrix $M^U$ and item latent factor matrix $M^I$.

Initialization:

  – Initialize training set $S^T$ with $S^E$, train the latent factor vectors for users and items in $S^T$, and assign them
    back to $M^U$ and $M^I$.
  – Initialize an empty set $\hat{S}^E$, in which the implicit feedback estimation triples $(u, i, \hat{r}_{ui}$ will be included.

BEGIN
Repeat:
    For each user-item pair $(u, i)$ in $S^I$:
        If both $u$ and $i$ have latent factor vectors:
            Estimate rating $\hat{r}_{ui}$ for $(u, i)$;
            Put $(u, i, \hat{r}_{ui})$ in $\hat{S}^E$ by Equation 6;
            Remove $(u, i)$ from $S^I$;
        Else If $u$ has latent factor vector but $i$ not:
            Estimate rating $\hat{r}_{ui}$ for $(u, i)$ by Equation 8 when item neighbors of $i$ can be found;
            Put $(u, i, \hat{r}_{ui})$ in $\hat{S}^E$;
            Remove $(u, i)$ from $S^I$;
        Else If $i$ has latent factor vector but $u$ not:
            Estimate rating $\hat{r}_{ui}$ for $(u, i)$ by Equation 10 when user neighbors of $u$ can be found;
            Put $(u, i, \hat{r}_{ui})$ in $\hat{S}^E$;
            Remove $(u, i)$ from $S^I$;
    Train model using $S^T \leftarrow S^T \bigcup \hat{S}^E$;
    Update the corresponding columns of $M^U$ and $M^I$ by updated latent factor vectors;
    Evaluate the difference between the rating estimations produced by previous latent factor vectors and the
    rating estimations produced by current latent factor vectors;
Until there is no new entry added in $\hat{S}^E$ and the estimation difference is lower than the threshold.
END
```

Fig. 2. The formal description of EMCF algorithm

users and items that already have latent factor vectors, their latent factor vectors are updated, since the EMCF model is re-trained using the updated rating set. Therefore, the estimations of some non-eligible implicit feedback in the previous round become possible. Then, EMCF algorithm is back to the step of estimating implicit feedback, and the above steps are repeated. The algorithm is terminated when there is no longer eligible implicit feedback to estimate and the rating estimation difference between the previous round and current round is lower than a pre-defined threshold. The formal algorithm procedure description is shown in Figure 2.

The EMCF algorithm has advantages by combining explicit feedback with implicit feedback. First, the implicit feedback is categorized into the four disjoint sets, in terms of the current situations of user and item latent factor vectors. Therefore, we have a chance to deal with implicit feedback differentially. Second, the estimation methods for different cases not only depend on the rating calculation from the matrix factorization, but also consider the neighbor structure built by both explicit feedback and implicit feedback. Third, the iterative procedure of EMCF fully utilizes implicit feedback by providing the opportunity to include more estimations of implicit feedback, which are not eligible in the previous operational round of the algorithm. Finally, the EMCF algorithm prevents noisy implicit feedback from the model training procedure, so that the performance of output model can be improved. Some implicit feedback is not used, since there is no sufficent information for estimation. Usually, such implicit feedback is suspected of being noise.

5 Experiments

We design experiments to demostrate the performance of EMCF. MovieLens[1] is used in the experiments. The dataset consists of one million ratings from 6,000 users and 4,000 movies. In the dataset, 20% ratings are randomly selected and held as the testing set, and the other 80% ratings are used as the training set. In the training set, we follow the idea proposed in [1] to create implicit feedback from explicit ratings data by considering whether a movie is rated by a user. In each experiment, the percentage of implicit feedback is pre-defined, and the rest of ratings in the training set are used as explicit feedback. The experiments are conducted on Apache Mahout[2], which is a recently popular machine learning platform. On Mahout, we implement matrix factorization and co-rating using ALS, and implement EMCF using SGD. All the methods are trained based on the same sets of explicit and implicit feedback, and are tested on the same sets of ratings. The root mean square error (RMSE) has been used as the evaluation measure to compare the methods.

The first set of experiments aims at comparing the performances of EMCF when only considering individual cases of implicit feedback. Given 20% of training set as explicit feedback, the baseline is the matrix factorization only using explicit feedback, which is used to compare to EMCF with different cases of implicit feedback. The results are shown in Table 1.

Table 1. Given 20% explicit feedback, experimental results of matrix factorization only with explicit feedback (MF+Explicit), EMCF with implicit feedback of Case 1, EMCF with implicit feedback with Case 2, EMCF with implicit feedback of Case 3, EMCF with implicit feedback with Case 2 and Case 3, and EMCF with all the implicit feedback

	MF + Explicit	EMCF + Case1	EMCF + Case2	EMCF + Case3	EMCF + Cases2+3	EMCF + All
RMSE	1.039	1.048	0.985	0.990	0.968	0.945

From the results, we can see that the performance of EMCF with implicit feedback of Case 1 is worse than the baseline. It is because the implicit feedback of Case 1 is estimated by the latent factor vectors learned from the model only based on explicit feedback. Without estimations of other cases of implicit feedback, EMCF with Case 1 overfits the model. EMCF with implicit feedback of Case 2 or Case 3 outperforms the baseline. But the improvements of performance are not obvious. EMCF with the implicit feedback combination of Case 2 and Case 3 has a greater improvements compared to the baseline. However, the implicit feedback is not fully utilized due to the lack of Case 1. EMCF with all the implicit feedback has the best performance since the EM-style algorithm of EMCF fully utilizes all the implicit feedback.

[1] http://www.grouplens.org/node/73

[2] http://mahout.apache.org/

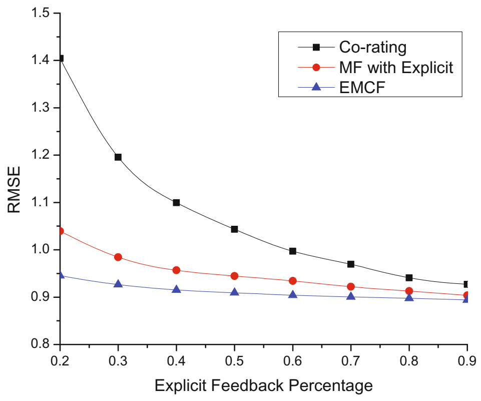

Fig. 3. The experimental results of matrix factorization (MF) only with explicit feedback, co-rating, and EMCF with different percentage of explicit feedback

The second set of experiments aims at comparing the performances of EMCF with different percentages of explicit feedback. There are two baseline methods: matrix factorization only with explicit feedback (MF with Explicit) and co-rating [6]. There are several differences between the co-rating method and our EMCF method. First, EMCF does not treat explicit feedback as a form of special implicit feedback. Instead, EMCF initializes a latent factor model with explicit feedback and then updates the latent factor vectors based on explicit feedback ratings and implicit feedback estimations. Second, EMCF does not normalize explicit feedback and implicit feedback into the same scale. On the contrary, EMCF estimates the ratings of implicit feedback in terms of the scale of explicit feedback. Third, EMCF does not add any new term in the objective function of the matrix factorization. The proposed Expectation-maximization-styled algorithm in EMCF ensures that the estimations of implicit feedback can be adjusted in order to improve the performance of EMCF. The experimental results are shown in Figure 3.

From the results, we can see that EMCF outperforms the other two baseline methods, especially when the percentage of implicit feedback is small. On one hand, the results of the comparison between EMCF and matrix factorization show that utilizing implicit feedback with explicit feedback truly outperforms the method based only on explicit feedback. On the other hand, the results of the comparison between EMCF and co-rating show that simply treating explicit feedback as implicit feedback hurts the performance of the method. The reason may be due to too much noisy implicit feedback added into the model training. The EMCF not only differentially estimates implicit feedback, but also iteratively updates the estimations based on both explicit and implicit feedback, by which noisy implicit feedback can be prevented from affecting the performance.

6 Conclusion and Future Works

In this paper, we present a novel method, Expectation-Maximization Collaborative Filtering (EMCF), which combines explicit and implicit feedback using matrix factorization for recommender systems. EMCF is based on the fact that explicit feedback and implicit feedback are naturally complementary to each other, since explicit feedback has good accuracy, but the quantity is insufficient; whereas, implicit feedback has abundant quantity, but does not have good accuracy. After initializing the EMCF model with explicit feedback, we observe that the implicit feedback can be categorized into four cases, in terms of the distribution of latent factor vectors. We propose three methods to differentially estimate the implicit feedback for the different cases. An EM-styled algorithm is then designed to iteratively propagate the implicit feedback estimations and update the latent factor vectors based on all the available explicit feedback ratings and implicit feedback estimations. We conduct experiments to compare EMCF with two other baseline methods. The experimental results show that EMCF outperforms the other two baselines, especially when the explicit feedback percentage is small.

In the future, we will continue to study how explicit feedback should be combined with implicit feedback in recommender systems. We will adapt different recommender methods for explicit and implicit feedback. We will also consider more complicated types of implicit feedback, such as mouse movements, and more features of implicit feedback, such as the durations of implicit feedback actions.

References

1. Bell, R.M., Koren, Y.: Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights, Los Alamitos, CA, USA, pp. 43–52 (2007)
2. Desrosiers, C., Karypis, G.: A Comprehensive Survey of Neighborhood-based Recommendation Methods. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 107–144. Springer, Boston (2011)
3. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Transactions on Information Systems (TOIS) 22, 89–115 (2004)
4. Hu, Y., Koren, Y., Volinsky, C.: Collaborative Filtering for Implicit Feedback Datasets. In: IEEE International Conference on, Los Alamitos, CA, USA, pp. 263–272 (2008)
5. Koren, Y., Bell, R., Volinsky, C.: Matrix Factorization Techniques for Recommender Systems. Computer 42(8), 30–37 (2009)
6. Liu, N.N., Xiang, E.W., Zhao, M., Yang, Q.: Unifying explicit and implicit feedback for collaborative filtering, New York, NY, USA, pp. 1445–1448 (2010)
7. Pan, R., Scholz, M.: Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD, Paris, France, p. 667 (2009)

8. Pazzani, M.J., Billsus, D.: Content-Based Recommendation Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)
9. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. In: Advances in Artificial Intelligence (2009)
10. Wang, J., Robertson, S., Vries, A.P., Reinders, M.J.T.: Probabilistic relevance ranking for collaborative filtering. Information Retrieval 11(6), 477–497 (2008)
11. Zhou, Y., Wilkinson, D., Schreiber, R., Pan, R.: Large-Scale Parallel Collaborative Filtering for the Netflix Prize. In: Fleischer, R., Xu, J. (eds.) AAIM 2008. LNCS, vol. 5034, pp. 337–348. Springer, Heidelberg (2008)