TagiCoFi: Tag Informed Collaborative Filtering

Yi Zhen
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong, China
yzhen@cse.ust.hk

Wu-Jun Li
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong, China
liwujun@cse.ust.hk

Dit-Yan Yeung
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong, China
dyyeung@cse.ust.hk

ABSTRACT

Besides the rating information, an increasing number of modern recommender systems also allow the users to add personalized tags to the items. Such tagging information may provide very useful information for item recommendation, because the users' interests in items can be implicitly reflected by the tags that they often use. Although some *content-based recommender systems* have made preliminary attempts recently to utilize tagging information to improve the recommendation performance, few *recommender systems based on collaborative filtering* (CF) have employed tagging information to help the item recommendation procedure. In this paper, we propose a novel framework, called *tag informed collaborative filtering* (TagiCoFi), to seamlessly integrate tagging information into the CF procedure. Experimental results demonstrate that TagiCoFi outperforms its counterpart which discards the tagging information even when it is available, and achieves state-of-the-art performance.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*; H.2 [**Database Management**]: Database Application—*Data Mining*

General Terms

Algorithms

Keywords

Collaborative filtering, recommender systems, tag

1. INTRODUCTION

Since the amount of information on the Web is increasing at an astonishing rate that is much faster than our ability to process it, recommendation plays a more and more important role for us to make effective use of the information available. Some representative examples include product recommendation in Amazon.com [14], movie recommendation in Netflix [3] and MovieLens[1] [16], reference recommendation in CiteULike[2], and bookmark recommendation in Del.icio.us[3]. Existing recommender systems can be roughly divided into two major categories [1]. *Content-based systems* [2, 12, 15] make use of profiles of the users or products to characterize their nature. On the other hand, systems based on *collaborative filtering* (CF) [4, 9, 16, 17, 19] do not exploit explicit user profiles but only past activities of the users, such as their transaction history or product satisfaction expressed in ratings, to predict the future activities of the users. In recent years, CF-based systems have become more and more popular than content-based systems because it is much easier to collect the past activities of users than their profiles due to privacy considerations.

In recent years, besides the *ratings* on the items given by the users, an increasing number of modern recommender systems also allow the users to add personalized *tags*[4], in the form of words or phrases, to the items. For example, users may add tags to movies in MovieLens, to web sites in Del.icio.us and to references in CiteULike. Such tagging information may provide very useful information for item recommendation, because the users' interests in items can be implicitly reflected by the tags that they often use [21]. For example, if two users often use the tags "Oscar" and "Tom Hanks", both of them may like the movie "Forrest Gump". In fact, the effectiveness of tags in representing users' preference or interests has been validated by Zanardi *et al.* in the CiteULike dataset [27]. Very recently, some *content-based systems*, such as those in [6, 22, 23], have made some preliminary attempts to utilize tagging information to improve the recommendation performance. However, there has been little work on improving CF-based systems with the help of tagging information. Because CF-based systems have become more popular than content-based systems, it would be a very worthwhile endeavor to devise novel CF techniques which can also utilize tagging information for item recommendation.

Existing CF methods can be divided into two main cat-

[1]http://movielens.umn.edu/
[2]http://www.citeulike.org/
[3]http://delicious.com/
[4]It should be emphasized that the setting in this paper is different from those about tag recommendation [7, 24] in which the recommended objects are tags. The recommended objects in this paper are called items, whereas tags are other objects about the items added by users.

egories [10]. Memory-based methods, such as [9, 19], try to predict new ratings by (weighted) averaging the ratings of similar users or items. On the other hand, model-based methods, such as *probabilistic matrix factorization* (PMF) [17], try to learn a model from data using statistical learning techniques. To the best of our knowledge, there exists only one CF method [25] which attempts to utilize tagging information to improve item recommendation. This method is a *memory-based* one. The experimental results in [25] show that little improvement could be achieved on item recommendation by integrating tagging information into the CF procedure under the memory-based framework.

In this paper, we propose a novel framework, called *tag informed collaborative filtering* (TagiCoFi), to seamlessly integrate tagging information into the *model-based* CF procedure. More specifically, we use tagging information to regularize the matrix factorization (MF) procedure of PMF [17] which has been demonstrated to be one of the state-of-the-art CF methods. Some promising properties of TagiCoFi are highlighted here:

- To the best of our knowledge, TagiCoFi is the first work that incorporates tagging information into a model-based CF system for item recommendation.
- TagiCoFi outperforms its counterpart, PMF, which discards the tagging information even when it is available. This shows that the tagging information does contain useful information for item recommendation and TagiCoFi can utilize it very effectively.
- TagiCoFi can overcome, or at least alleviate, the overfitting problem [17] suffered by most MF-based CF methods due to the sparsity of the rating matrix.
- TagiCoFi can solve the *cold-start* problem [8, 11, 20] in that it can give recommendations to novel users who have no preference on any items.

The rest of this paper is organized as follows. In Section 2, we will introduce the notations and some preliminaries. Section 3 describes the details of our model. Experimental results are presented in Section 4 and, finally, we conclude the paper in Section 5.

2. NOTATIONS AND PRELIMINARIES

In this section, we first introduce some notations used in this paper. We then briefly review PMF [17] which is closely related to our work.

2.1 Notations

We use boldface uppercase letters, such as $\mathbf{A}$, to denote matrices, and boldface lowercase letters, such as $\mathbf{b}$, to denote vectors. The ith row and the jth column of a matrix $\mathbf{A}$ are denoted as $\mathbf{A}_{i*}$ and $\mathbf{A}_{*j}$, respectively. The (i,j)th element of $\mathbf{A}$ is denoted as A_{ij} and the ith element of $\mathbf{b}$ as b_i.

Suppose there are N users, M items and K tags. Let $\mathbf{R}$ be the rating matrix in which R_{ij} represents the rating of user i for item j. The matrix $\mathbf{R}$ is sparse because many elements are missing, and each such element R_{ij} is assigned the value of 0 to indicate that item j has not been rated by user i. $\mathbf{Y}$ is the indicator matrix where Y_{ij} is an indicator variable which is equal to 1 if user i rated item j and 0 otherwise. MF-based methods [17] seek to find two low-rank matrices $\mathbf{U} \in \mathbb{R}^{D\times N}$ and $\mathbf{V} \in \mathbb{R}^{D\times M}$, where typically $D \ll N, M$, and use $\hat{\mathbf{R}} = \mathbf{U}^T\mathbf{V}$ to approximate the rating matrix $\mathbf{R}$. The column vectors $\mathbf{U}_{*i}$ and $\mathbf{V}_{*j}$ represent the user-specific and item-specific latent feature vectors, respectively.

Let $\mathbf{Z}$ be the tagging matrix, and each of its elements Z_{ik} is the *tf*idf* value of user i and tag k [18, 23]:

$$Z_{ik} = \mathrm{tf}(i,k) \times \log_2\left(\frac{N}{\mathrm{df}(k)}\right), \qquad (1)$$

where $\mathrm{tf}(i,k)$ is the normalized frequency of tag k appeared in user i's tagging history and $\mathrm{df}(k)$ is the number of users who have used tag k.

2.2 Probabilistic Matrix Factorization

PMF [17] seeks to derive the aforementioned low-rank matrices $\mathbf{U}$ and $\mathbf{V}$ by analyzing the rating matrix $\mathbf{R}$ in a probabilistic framework. The *likelihood* of the observed ratings $\mathbf{R}$ is defined as follows:

$$p(\mathbf{R} \mid \mathbf{U}, \mathbf{V}, \sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}(R_{ij} \mid \mathbf{U}_{*i}^T\mathbf{V}_{*j}, \sigma^2)\right]^{Y_{ij}}, \qquad (2)$$

where $\mathcal{N}(x \mid \mu, \sigma^2)$ denotes the (univariate) Gaussian distribution with mean μ and variance σ^2.

Putting zero-mean spherical Gaussian priors on the user-specific and item-specific feature vectors:

$$p(\mathbf{U} \mid \sigma_U^2) = \prod_{i=1}^{N}\mathcal{N}(\mathbf{U}_{*i} \mid \mathbf{0}, \sigma_U^2\mathbf{I})$$
$$p(\mathbf{V} \mid \sigma_V^2) = \prod_{j=1}^{M}\mathcal{N}(\mathbf{V}_{*j} \mid \mathbf{0}, \sigma_V^2\mathbf{I}),$$

we can obtain the *maximum a posteriori* (MAP) estimates of $\mathbf{U}$ and $\mathbf{V}$ by minimizing the following objective function defined based on the sum of squared errors:

$$\begin{aligned} E &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} Y_{ij}(R_{ij} - \mathbf{U}_{*i}^T\mathbf{V}_{*j})^2 \\ &\quad + \frac{\lambda_U}{2}\mathrm{tr}(\mathbf{U}^T\mathbf{U}) + \frac{\lambda_V}{2}\mathrm{tr}(\mathbf{V}^T\mathbf{V}), \end{aligned} \qquad (3)$$

where $\lambda_U = \sigma^2/\sigma_U^2$ and $\lambda_V = \sigma^2/\sigma_V^2$.

3. TAG INFORMED COLLABORATIVE FILTERING

Because PMF [17] has achieved state-of-the-art performance for CF tasks, we use it as the base model to make further enhancement by integrating tagging information in a principled way. The result is our *tag informed collaborative filtering* method, which will be abbreviated as TagiCoFi in the sequel. The key idea of TagiCoFi is to use tagging information to regularize the MF procedure of PMF. More specifically, we seek to make two user-specific latent feature vectors as similar as possible if the two users have similar tagging history.

In the rest of this section, we first introduce some metrics for characterizing the similarity between users based on tagging information. We then propose our TagiCoFi model based on the computed user similarities.

3.1 Tag-based User Similarity Measures

We introduce several possible measures for characterizing user similarities based on the tagging matrix $\mathbf{Z}$. Here, T^{ij} denotes the index set of tags which are used by both user i and user j.

3.1.1 Cosine Similarity

The *cosine similarity* is defined as follows:

$$S_{ij}^{\cos} = \frac{\sum_{k\in T^{ij}} Z_{ik}Z_{jk}}{\sqrt{\sum_{k\in T^{ij}} Z_{ik}^2}\sqrt{\sum_{k\in T^{ij}} Z_{jk}^2}}. \quad (4)$$

3.1.2 Pearson Similarity

The Pearson correlation coefficient between two users is defined as follows:

$$\rho_1(i,j) = \frac{\sum_{k\in T^{ij}}(Z_{ik}-\bar{Z}_i)(Z_{jk}-\bar{Z}_j)}{\sqrt{\sum_{k\in T^{ij}}(Z_{ik}-\bar{Z}_i)^2}\sqrt{\sum_{k\in T^{ij}}(Z_{jk}-\bar{Z}_j)^2}}, \quad (5)$$

where $\bar{Z}_i = \frac{\sum_{k\in T^{ij}} Z_{ik}}{|T^{ij}|}$. The *Pearson similarity* is then defined as:

$$S_{ij}^{\text{pea}} = \frac{1}{1+\exp(-\rho_1(i,j))}. \quad (6)$$

3.1.3 Euclidean-based Similarity

The Euclidean distance between two users is defined as follows:

$$\rho_2(i,j) = \sqrt{\sum_{k\in T^{ij}}(Z_{ik}-Z_{jk})^2}. \quad (7)$$

The *Euclidean-based similarity* is then defined as:

$$S_{ij}^{\text{euc}} = \exp\left(-\frac{[\rho_2(i,j)]^2}{2\sigma^2}\right), \quad (8)$$

where σ is a user-controlled parameter.

3.2 Model Formulation of TagiCoFi

Like in PMF [17], we adopt a similar MF procedure to find $\mathbf{U}$ and $\mathbf{V}$ by minimizing the following criterion function:

$$\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} Y_{ij}(R_{ij}-\mathbf{U}_{*i}^T\mathbf{V}_{*j})^2 + \frac{\alpha}{2}\Big[\text{tr}(\mathbf{U}^T\mathbf{U}) + \text{tr}(\mathbf{V}^T\mathbf{V})\Big], \quad (9)$$

where α is a regularization parameter for complexity control.

Furthermore, TagiCoFi employs the user similarities defined based on the tagging information to regularize the MF procedure, with the goal to make the user-specific latent feature vectors as similar as possible if the corresponding users have similar tagging history. We can achieve this goal by minimizing the following criterion function:

$$\begin{aligned} f_1 &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} S_{ij}\|\mathbf{U}_{*i}-\mathbf{U}_{*j}\|^2 \\ &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\Big[S_{ij}\sum_{d=1}^{D}(U_{di}-U_{dj})^2\Big] \\ &= \sum_{d=1}^{D}\mathbf{U}_{d*}\boldsymbol{\mathcal{L}}\mathbf{U}_{d*}^T \\ &= \text{tr}(\mathbf{U}\boldsymbol{\mathcal{L}}\mathbf{U}^T), \end{aligned} \quad (10)$$

where S_{ij} is the tag-based similarity between user i and user j computed based on one of the measures defined in Section 3.1, $\boldsymbol{\mathcal{L}} = \mathbf{D}-\mathbf{S}$ is known as the Laplacian matrix [5] with $\mathbf{D}$ being a diagonal matrix whose diagonal elements $D_{ii} = \sum_j S_{ij}$, and $\text{tr}(\cdot)$ denotes the trace of a matrix.

To integrate tagging information into the CF procedure, TagiCoFi combines the criteria (9) and (10) to give the following objective function for minimization:

$$\begin{aligned} f = &\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} Y_{ij}(R_{ij}-\mathbf{U}_{*i}^T\mathbf{V}_{*j})^2 \\ &+ \frac{\alpha}{2}\Big[\text{tr}(\mathbf{U}^T\mathbf{U}) + \text{tr}(\mathbf{V}^T\mathbf{V})\Big] \\ &+ \frac{\beta}{2}\Big[\text{tr}(\mathbf{U}\boldsymbol{\mathcal{L}}\mathbf{U}^T)\Big], \end{aligned} \quad (11)$$

where β is an additional regularization parameter to control the contribution from the tagging information.

The formulation in (11) can be seen as an adaptation of *relation regularized matrix factorization* (RRMF) [13] which models relational data containing both relation information and content information. The main difference between TagiCoFi and RRMF is that TagiCoFi can handle missing data, which is one of the key characteristics of CF.

3.3 Learning

The objective function in (11) can be rewritten as follows:

$$\begin{aligned} f = &\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} Y_{ij}(R_{ij}-\mathbf{U}_{*i}^T\mathbf{V}_{*j})^2 \\ &+ \frac{1}{2}\text{tr}\Big[\mathbf{U}(\alpha\mathbf{I}+\beta\boldsymbol{\mathcal{L}})\mathbf{U}^T\Big] \\ &+ \frac{\alpha}{2}\Big[\text{tr}(\mathbf{V}^T\mathbf{V})\Big], \end{aligned} \quad (12)$$

where $\mathbf{I}$ is the identity matrix. We use an alternating gradient descent procedure to optimize (12). More specifically, each time we fix one variable ($\mathbf{U}$ or $\mathbf{V}$) and minimize the objective function with respect to the other one ($\mathbf{V}$ or $\mathbf{U}$). This procedure is repeated for several iterations until some termination condition is satisfied.

To learn $\mathbf{U}$, we first rewrite (12) as follows:

$$f = g + h + C, \quad (13)$$

where C is a constant independent of $\mathbf{U}$, and

$$\begin{aligned} g &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} Y_{ij}(R_{ij}-\mathbf{U}_{*i}^T\mathbf{V}_{*j})^2 \\ h &= \frac{1}{2}\sum_{d=1}^{D}\mathbf{U}_{d*}(\alpha\mathbf{I}+\beta\boldsymbol{\mathcal{L}})\mathbf{U}_{d*}^T. \end{aligned} \quad (14)$$

From (14), we can see that the rows of $\mathbf{U}$ in h are decoupled. Hence, we apply gradient descent to optimize one row of $\mathbf{U}$ at a time with the other rows fixed.

Because

$$\begin{aligned} \frac{\partial g}{\partial U_{di}} = &\Big(\sum_{j=1}^{M} Y_{ij}V_{dj}^2\Big)U_{di} - \\ &\sum_{j=1}^{M} Y_{ij}V_{dj}(R_{ij}-\mathbf{U}_{*i}^T\mathbf{V}_{*j}+U_{di}V_{dj}), \end{aligned} \quad (15)$$

we have

$$\frac{\partial g}{\partial U_{d*}} = \mathbf{W} U_{d*}^T - \mathbf{x}, \tag{16}$$

where $\mathbf{W}$ is an $N \times N$ diagonal matrix with $W_{ii} = \sum_{j=1}^{M} Y_{ij} V_{dj}^2$, and $\mathbf{x}$ is an $N \times 1$ vector with $x_i = \sum_{j=1}^{M} Y_{ij} V_{dj} (R_{ij} - \mathbf{U}_{*i}^T \mathbf{V}_{*j} + U_{di} V_{dj})$.

Then, we can get

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{U}_{d*}} &= \frac{\partial g}{\partial \mathbf{U}_{d*}} + \frac{\partial h}{\partial \mathbf{U}_{d*}} \\ &= (\mathbf{W} + \alpha \mathbf{I} + \beta \boldsymbol{\mathcal{L}}) \mathbf{U}_{d*}^T - \mathbf{x}. \end{aligned} \tag{17}$$

The learning process of $\mathbf{V}$ is different from that of $\mathbf{U}$, because the columns (not rows) of $\mathbf{V}$ are decoupled. Hence, we apply gradient descent to optimize one column of $\mathbf{V}$ at a time with the other columns fixed. The gradient can be computed as follows:

$$\frac{\partial f}{\partial \mathbf{V}_{*j}} = \Big(\alpha \mathbf{I} + \sum_{i=1}^{N} Y_{ij} \mathbf{U}_{*i} \mathbf{U}_{*i}^T\Big) \mathbf{V}_{*j} - \sum_{i=1}^{N} Y_{ij} R_{ij} \mathbf{U}_{*i}. \tag{18}$$

The overall learning procedure of TagiCoFi is summarized in Algorithm 1 below.

Algorithm 1 Learning procedure of TagiCoFi

```
1: INPUT:
     R – rating matrix
     Z – tagging matrix
     D – number of latent features
     W – number of iterations
     δ – step size for gradient descent
2: Compute user similarity matrix S based on Z
3: Compute Laplacian matrix 𝓛 based on S
4: Initialize U^0, V^0
5: for w = 1 to W do
6:    for d = 1 to D do
7:       U^w_{d*} ← U^{w−1}_{d*} − δ ∂f/∂U_{d*}
8:    end for
9:    for j = 1 to M do
10:      V^w_{*j} ← V^{w−1}_{*j} − δ ∂f/∂V_{*j}
11:   end for
12: end for
13: return U^W, V^W
```

3.4 Complexity Analysis

The main computation of TagiCoFi is to evaluate the gradients of the objective function with respect to the latent variables and to compute the user similarities. The time complexity of computing the gradient $\frac{\partial f}{\partial \mathbf{U}_{d*}}$ is $O(N^2D)$) and that of $\frac{\partial f}{\partial \mathbf{V}_{*j}}$ is $O(NMD)$. The time complexity of computing the user similarities and $\boldsymbol{\mathcal{L}}$ is $O(N^2K)$. Hence, the time complexity of the entire alternating gradient descent procedure is $O(W(N^2D + NMD) + N^2K)$.

4. EXPERIMENTAL EVALUATION

We have conducted several experiments to compare the performance of our method with that of other methods. Through the experiments, we have tried to answer the following questions:

1. How does TagiCoFi perform in real applications when compared with state-of-the-art methods?
2. How effective are the different user similarity measures?
3. How does tagging information improve collaborative filtering?
4. How does the number of latent features used affect the performance of TagiCoFi?
5. Does TagiCoFi work for users without any training ratings?

These questions are answered separately: question 1 in Section 4.3, questions 2–4 in Section 4.4 as three different subsubsections, and question 5 in Section 4.5.

4.1 Data Set

We evaluate our algorithm on the MovieLens dataset[5], which, as far as we know, is the only publicly available dataset containing both tagging and rating information.

We first prune the dataset for our analysis. For the tagging information, we only keep those tags which are added on at least three distinct movies. As for the users, we only keep those users who used at least 3 distinct tags in their tagging history. For movies, we only keep those movies that are annotated by at least 3 distinct tags. It should be emphasized that our model still works under situations where there are users or movies with rating information only but no tagging information. For those users without any tagging information, the tag-based similarities between them and the other users are 0, which means that the last term in (11) will have no effect on those users. Subsequently, the recommendation result for those users without tagging information only depends on the MF procedure of the rating matrix, which is similar to the result of PMF. As the focus of this paper is on evaluating the effectiveness of tagging information in addition to rating information, we only keep the users who have both rating history and tagging history in the original rating records.

We obtain two kinds of records after pruning, the tagging records and the rating records. The tagging records include 13,431 *tagging applications*[6] contributed by 757 users with 2,271 distinct tags. Based on the tagging records, we construct the tagging matrix $\mathbf{Z}$, whose elements are defined by Equation (1) in Section 2.1. The rating records include 167,474 ratings rated by 757 users (the same as those in the tagging records) on 9,485 movies, and based on these rating records we construct the rating matrix $\mathbf{R}$. More statistics about the rating matrix $\mathbf{R}$ are shown in Table 1, where the numbers behind $\pm$ denote the standard deviations.

Table 1: Description of rating data

Statistics	Users	Movies
Min. # of ratings	20	1
Max. # of ratings	2,634	625
Mean # of ratings	441.95 ± 420.88	35.27 ± 67.30

[5] `http://www.grouplens.org/node/73`

[6] If user i adds tag k on item j, we say this is a tagging application.

4.2 Evaluation Metric

For consistency with experiments reported in the literature, we use the Mean Absolute Error (MAE) as evaluation metric. MAE gives the average absolute deviation of prediction from the ground truth:

$$\mathrm{MAE} = \frac{\sum_i \sum_j Y_{ij} |R_{ij} - \hat{R}_{ij}|}{\sum_i \sum_j Y_{ij}},$$

where R_{ij} and $\hat{R}_{ij}$ are the true and predicted rating values, respectively. A smaller value of MAE indicates a better performance.

In our experiments, we randomly split the rating records into two parts, each of which contains 50% of the observations in the rating matrix. One part is used as the test set, which is kept the same for all experiments. The other part is used as a pool from which training sets are generated. For example, a training set size of 20% means that 20% of the records are randomly selected from the pool to form a training set. For each training set size, we randomly generate 10 different training sets based on which 10 experiments are performed and the average result is reported.

4.3 Performance

In this section, we compare our method with PMF which has been demonstrated to be one of the state-of-the-art CF methods [17]. For fairness, we perform parameter tuning in advance for each method and then use the best settings found in all the experiments. For both methods, we initialize the latent features to random numbers in $[0, 1]$ and set the step size for gradient descent to 0.001. The parameters specific to our method are set as $\alpha = 1$ and $\beta = 50$. Actually, we find that the performance will be stable after about 1000 rounds of gradient decent (see Figure 3). Hence, we set $W = 1000$ for all the following results. Furthermore, we adopt the Pearson similarity for all the experiments. The performance of other measures will be discussed in Section 4.4.1.

The results reported in Table 2 are the average MAE values of PMF and TagiCoFi and their corresponding standard deviations. The better results are shown in bold. It iss clear that TagiCoFi achieves better performance than PMF.

To evaluate how significant TagiCoFi outperforms PMF, we have conducted paired t-tests [26] on the results of PMF and TagiCoFi. Given two approaches, say A and B, and a set of n experiments, the MAE values are obtained for both approaches, denoted by a_i and b_i for $i = 1, 2, \ldots, n$. Let $d_i = a_i - b_i$ denote the difference of a_i and b_i and $\bar{d}$ be the average of the d_i values for $i = 1, 2, \ldots, n$. The null hypothesis is $\bar{d} = 0$ whereas the alternative hypothesis is $\bar{d} > 0$. The *p-value* is computed using the t-statistic:

$$T = \frac{\bar{d}}{s/\sqrt{n}},$$

where s is the standard deviation of d. A small *p-value* (≤ 0.01) indicates the existence of statistically significant evidence against the null hypothesis.

Table 3 shows the *p-values* obtained in our experiments. It is easily observed that TagiCoFi significantly outperforms PMF. Because the main difference between TagiCoFi and PMF lies in the extra tagging information used by TagiCoFi, we can conclude that the tagging information is very useful and TagiCoFi can utilize it very effectively.

Table 3: *p-values* for the significance tests

	$D = 5$	$D = 10$	$D = 20$
20% Training	3.91×10^{-15}	8.27×10^{-17}	1.04×10^{-16}
40% Training	4.11×10^{-13}	2.10×10^{-16}	4.52×10^{-16}
60% Training	1.35×10^{-11}	4.20×10^{-12}	4.15×10^{-12}
80% Training	1.24×10^{-8}	2.85×10^{-12}	5.99×10^{-12}

In order to compare TagiCoFi with PMF more thoroughly, we compare their performance on users with different numbers of observed ratings. The results are shown in Figure 1, from which we can find that TagiCoFi outperforms PMF for all users and the improvement is more significant for users with only few observed ratings. This is a very promising property of TagiCoFi because those users with a small number of ratings are typically new customers who have just started to use the system. If we can provide good recommendation to them, we will have a higher chance to keep them as our long-term customers. Otherwise we will likely lose them.

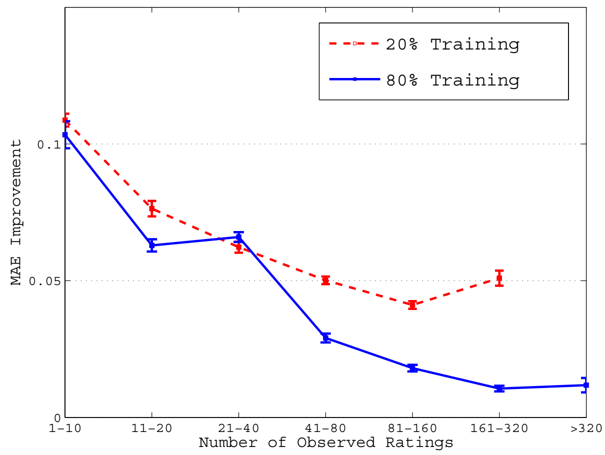


Figure 1: Performance improvement of TagiCoFi over that of PMF on different user rating scales (no users in a 20% training set have more than 320 observed ratings)

4.4 Sensitivity to Parameters

4.4.1 User Similarity Measures

In this section, we conduct a set of experiments to compare the effectiveness of the aforementioned user similarity measures: cosine similarity, Pearson similarity and Euclidean-based similarity. Due to the page limit restriction, we only report results with parameters $\alpha = 1, \beta = 50, D = 10$ in Figure 2. We have also observed the same trend in other parameter settings. From Figure 2, we see that the Pearson similarity always gives the best performance and the Euclidean-based similarity is always the worst. Although the difference between these measures is obvious, Figure 2 shows that the difference decreases as the training set size increases. One may ask if changing the σ parameter in the Euclidean-based similarity measure will help. We have tuned the parameter by trying different values but cannot make it outperform the other similarity measures. Based on this analysis, we adopt

Table 2: MAE comparison between PMF and TagiCoFi ($\times 10^{-2}$)

Training Set Size	User Average	Movie Average	$D = 5$ PMF	$D = 5$ TagiCoFi	$D = 10$ PMF	$D = 10$ TagiCoFi	$D = 20$ PMF	$D = 20$ TagiCoFi
20%	73.16±0.09	74.18±0.17	73.53±0.13	**68.86**±0.12	73.50±0.11	**68.32**±0.13	73.54±0.12	**67.73**±0.14
40%	72.76±0.07	71.98±0.09	68.79±0.08	**66.06**±0.12	68.88±0.06	**65.82**±0.07	68.91±0.08	**65.46**±0.14
60%	72.62±0.04	70.97±0.05	66.46±0.12	**64.70**±0.10	66.71±0.09	**64.65**±0.10	66.85±0.09	**64.59**±0.14
80%	72.53±0.02	70.39±0.04	64.88±0.14	**63.53**±0.10	65.15±0.10	**63.64**±0.10	65.42±0.07	**63.84**±0.11

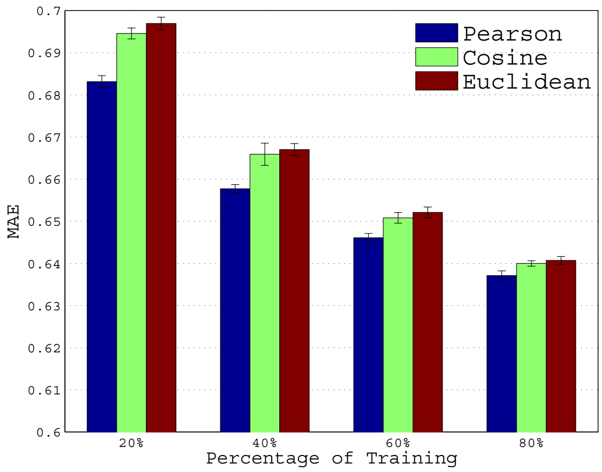


Figure 2: Comparison of similarity measures

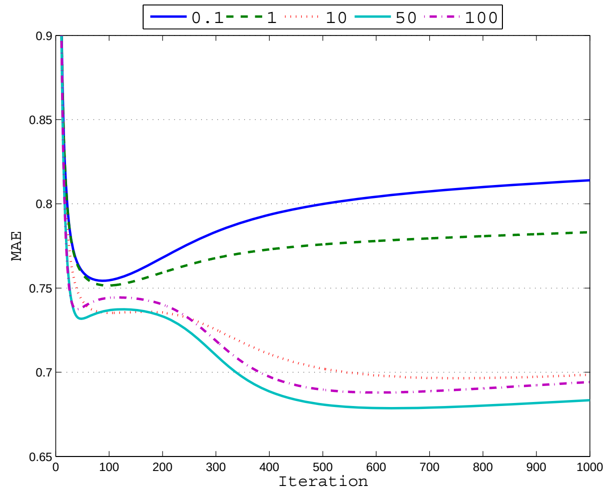


Figure 3: Impact of β

the Pearson similarity as our similarity measure in all other experiments.

4.4.2 Impact of Tagging Information

As we saw in Section 3, the contribution of tagging information is controlled by the parameter β. If $\beta = 0$, we do not use tagging information at all and hence our method degenerates to a special form of PMF; as β increases, we put larger weight on the tagging information. To evaluate the impact of tagging information on collaborative filtering, we carry out a set of experiments by varying the value of β. The MAE curves for different β values on 20% training sets are plotted in Figure 3. The other parameters are set as $\alpha = 1$ and $D = 10$.

As we can see from Figure 3, adopting a larger β value can help to avoid the overfitting problem suffered by most MF-based CF methods [17]. When $\beta \le 1$, the overfitting problem is apparent. If we set $\beta \ge 10$, we do not experience overfitting any more. This phenomenon clearly validates the impact of tagging information, that is, adding more tagging information can improve the generalization ability of the model. Moreover, Figure 3 also shows that the performance might degrade when β is too large. So in practice, we should choose a moderate value of β. Actually, our method is not sensitive to β within a wide range, such as $10 \le \beta \le 50$.

4.4.3 Impact of Number of Latent Features

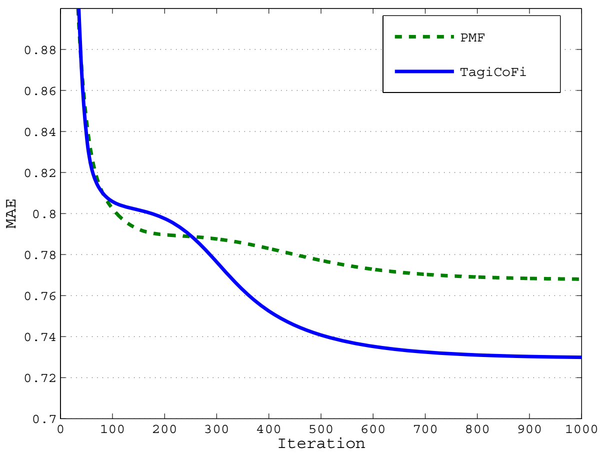
Another important parameter in our method is the number of latent features D. In this section, we conduct a set of experiments on 20% training sets to study how D affects the performance of our model. We use the following parameters: $\alpha = 1, \beta = 50$. The MAE values and their standard deviations are plotted in Figure 4. We also show the percentage decrease in MAE with respect to that for 10 latent features at the points 20, 30, 40 and 50.

Figure 4 shows that the MAE decreases as the number of latent features increases. This agrees with our assumption, because the more latent features, the more information can be represented by the latent feature vectors. The figure also shows that the improvement in MAE gets smaller as D continues to increase. When D becomes large enough, there is essentially no significant improvement because the useful information has already been represented well by the existing latent features. From Figure 4, we can see that TagiCoFi can achieve good performance with D taking a wide range of values.

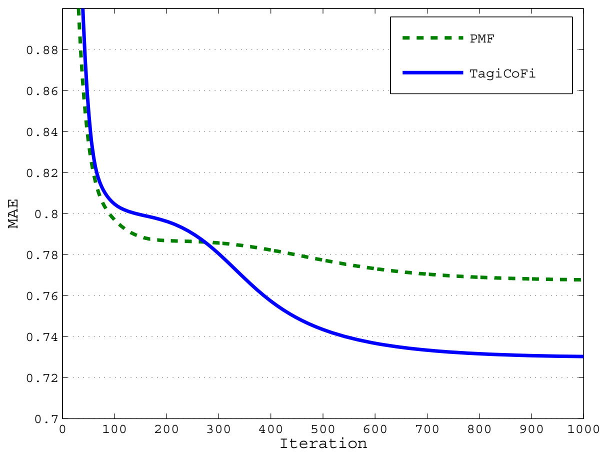
4.5 Cold-Start Setting

One well-known problem of CF systems is the *cold-start* problem, in which recommendations are required for users or items which have no observed ratings [20, 8, 11]. Pure CF methods, such as PMF, cannot work under a *cold-start* setting, since no preference information is available to form any basis for giving recommendation. Suppose tagging information is available, TagiCoFi can solve the cold-start problem by seamlessly integrating the tagging information for recommendation.

To validate the above speculation, we conduct two sets of experiments based on 20% training sets, where we randomly select 50 and 100 users and discard their ratings. These users, called *cold-start* users, are quite commonly found in many recommender systems, such as newly registered users in a system. In the experiments, the parameters of our



(a) 50 *cold-start* users



(b) 100 *cold-start* users

Figure 5: PMF and TagiCoFi in *cold-start* settings

Table 4: MAE comparison in *cold-start* settings

Types	50 *cold-start* users		100 *cold-start* users	
	PMF	TagiCoFi	PMF	TagiCoFi
Cold-start Users	0.7683 ± 0.0012	$\mathbf{0.7500} \pm 0.0009$	0.7623 ± 0.0022	$\mathbf{0.7425} \pm 0.0011$
All Users	0.7680 ± 0.0009	$\mathbf{0.7299} \pm 0.0016$	0.7677 ± 0.0016	$\mathbf{0.7303} \pm 0.0024$

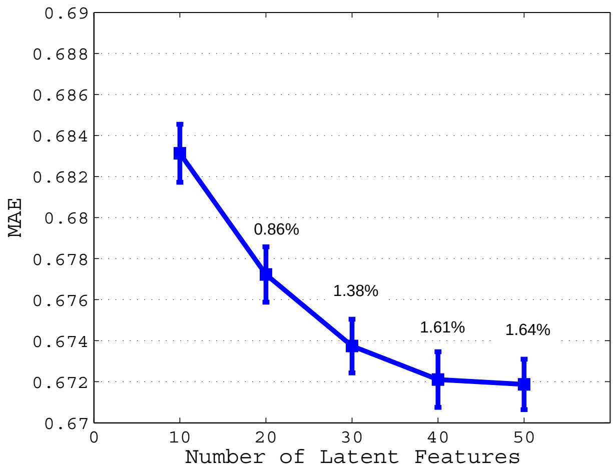


Figure 4: Impact of the number of latent features

model are set as $\alpha = 1$, $\beta = 50$ and $D = 10$. In our implementation of PMF, we use the item average to predict the rating of *cold-start* users for an item, because the original PMF cannot give prediction for those cold-start users. The MAE curves of PMF and TagiCoFi during the first 1000 iterations are plotted in Figure 5.

Figure 5 shows that TagiCoFi significantly outperforms PMF, validating our speculation that tagging information could be used to perform recommendation for *cold-start* users. Table 4 shows the MAE values of PMF and TagiCoFi together with their corresponding standard deviations on *cold-start* users and all users respectively in two different settings. It is clear that TagiCoFi performs better than PMF at all levels.

5. CONCLUSION

We have proposed a novel framework, TagiCoFi, to seamlessly incorporate tagging information into collaborative filtering for item recommendation. To the best of our knowledge, TagiCoFi is the first work that incorporates tagging information into a model-based CF system for item recommendation. One promising property of TagiCoFi is that it can overcome the overfitting problem suffered by most MF-based CF methods. Moreover, TagiCoFi can also solve the *cold-start* problem for novel users. Experimental results on real data demonstrate that TagiCoFi can significantly outperform state-of-the-art collaborative filtering algorithms, such as PMF, which discard the tagging information.

One of our future research directions is to extend TagiCoFi by incorporating into it the tagging history of items. Furthermore, we plan to extend TagiCoFi to incorporate additional sources of information to further improve the performance of recommender systems.

6. ACKNOWLEDGMENTS

We thank the GroupLens research lab at the University of Minnesota for their dataset. The research reported in this paper is supported by research grant HIA98/99.EG01 from the Hong Kong University of Science and Technology.

7. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE*

Transactions on Knowledge and Data Engineering, 17(6):734–749, June 2005.

[2] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[3] J. Bennett and S. Lanning. The Netflix prize. In *Proceedings of KDD Cup and Workshop*, volume 2007, 2007.

[4] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[5] F. Chung. *Spectral Graph Theory*. Number 92 in Regional Conference Series in Mathematics. American Mathematical Society, 1997.

[6] M. de Gemmis, P. Lops, G. Semeraro, and P. Basile. Integrating tags in a semantic content-based recommender. In *Proceedings of the 2nd ACM Conference on Recommender Systems*, pages 163–170, New York, NY, USA, 2008. ACM.

[7] N. Garg and I. Weber. Personalized, interactive tag recommendation for flickr. In *Proceedings of the 2nd ACM Conference on Recommender Systems*, pages 67–74, New York, NY, USA, 2008. ACM.

[8] H. Guo. Soap: live recommendations through social agents. In *Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering*, 1997.

[9] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 230–237, New York, NY, USA, 1999. ACM.

[10] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 426–434, New York, NY, USA, 2008. ACM.

[11] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, pages 208–211, New York, NY, USA, 2008. ACM.

[12] K. Lang. Newsweeder: learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339. Morgan Kaufmann, 1995.

[13] W.-J. Li and D.-Y. Yeung. Relation regularized matrix factorization. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 2009.

[14] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, Jan/Feb 2003.

[15] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the 5th ACM Conference on Digital Libraries*, pages 195–204, New York, NY, USA, 2000. ACM.

[16] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 8th ACM Conference on Computer Supported Cooperative Work*, pages 175–186, New York, NY, USA, 1994. ACM.

[17] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1257–1264. MIT Press, Cambridge, MA, 2008.

[18] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Infomation Processing Management: an International Journal*, 24(5):513–523, 1988.

[19] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pages 285–295, New York, NY, USA, 2001. ACM.

[20] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–260, New York, NY, USA, 2002. ACM.

[21] S. Sen, S. K. Lam, A. M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F. M. Harper, and J. Riedl. Tagging, communities, vocabulary, evolution. In *Proceedings of the 20th ACM Conference on Computer Supported Cooperative Work*, pages 181–190, New York, NY, USA, 2006. ACM.

[22] S. Sen, J. Vig, and J. Riedl. Tagommenders: connecting users to items through tags. In *Proceedings of the 18th International Conference on World Wide Web*, pages 671–680, New York, NY, USA, 2009. ACM.

[23] A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2nd ACM Conference on Recommender Systems*, pages 259–266, New York, NY, USA, 2008. ACM.

[24] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *Proceedings of the 2nd ACM Conference on Recommender Systems*, pages 43–50, New York, NY, USA, 2008. ACM.

[25] K. H. L. Tso-Sutter, L. B. Marinho, and L. Schmidt-Thieme. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of the 2nd ACM Symposium on Applied Computing*, pages 1995–1999, New York, NY, USA, 2008. ACM.

[26] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49, New York, NY, USA, 1999. ACM.

[27] V. Zanardi and L. Capra. Social ranking: uncovering relevant content using tag-based recommender systems. In *Proceedings of the 2nd ACM Conference on Recommender Systems*, pages 51–58, New York, NY, USA, 2008. ACM.