# A Social Recommendation Framework Based on Multi-scale Continuous Conditional Random Fields

Xin Xin, Irwin King, Hongbo Deng, Michael R. Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{xxin,king,hbdeng,lyu}@cse.cuhk.edu.hk

## ABSTRACT

This paper addresses the issue of social recommendation based on collaborative filtering (CF) algorithms. Social recommendation emphasizes utilizing various attributes information and relations in social networks to assist recommender systems. Although recommendation techniques have obtained distinct developments over the decades, traditional CF algorithms still have these following two limitations: (1) relational dependency within predictions, an important factor especially when the data is sparse, is not being utilized effectively; and (2) straightforward methods for combining features like linear integration suffer from high computing complexity in learning the weights by enumerating the whole value space, making it difficult to combine various information into an unified approach. In this paper, we propose a novel model, Multi-scale Continuous Conditional Random Fields (MCCRF), as a framework to solve above problems for social recommendations. In MCCRF, relational dependency within predictions is modeled by the Markov property, thus predictions are generated simultaneously and can help each other. This strategy has never been employed previously. Besides, diverse information and relations in social network can be modeled by state and edge feature functions in MCCRF, whose weights can be optimized globally. Thus both problems can be solved under this framework. In addition, We propose to utilize Markov chain Monte Carlo (MCMC) estimation methods to solve the difficulties in training and inference processes of MCCRF. Experimental results conducted on two real world data have demonstrated that our approach outperforms traditional CF algorithms. Additional experiments also show the improvements from the two factors of relational dependency and feature combination, respectively.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*


*CIKM'09,* November 2–6, 2009, Hong Kong, China.


## General Terms

Algorithms, Experimentation

## Keywords

Collaborative Filtering, Multi-scale Continuous Conditional Random Fields, Markov Chain Monte Carlo, Social Recommendation

## 1. INTRODUCTION

Recommender system is to suggest relevant items (news, books, movies, images, etc.) attracting particular users, which plays an important role on the Web nowadays since it satisfies both commercial companies and users in daily lives. Traditionally, CF algorithms are used in these systems, assigning each user-item pair a score indicating the user's rating on the item, based on which a ranking list of items is generated to the user as suggestions. Classical CF methods are divided into memory-based methods [4, 12, 14, 20, 30, 34] and model-based methods [7, 13, 32, 33]. Recently, social relations have been considered in many applications, and in this paper, we address the issue of social recommendation. Different from traditional recommender systems, in social recommendation, multiple information and various relational dependencies in social networks should be utilized to improve recommendation results. Traditional CF algorithms, however, suffer from the following two weaknesses.

To illustrate the problem, we use an example showed in Figure 1. In this example, there are four users, denoted by $u_l$ and seven items, denoted by $i_m$. $r_{lm}$ is rating record by $u_l$ to $i_m$. (e.g., scale from 1 to 5, higher value means better satisfaction). The CF algorithms predict values of unrated user-item pairs, denoted as $y_{lm}$ (without loss of generality, not all $y_{lm}$ are shown in the figure), and suggest top ranked items as recommendations.

**Lack of relational dependency within predictions.** In traditional methods, predictions are only relationally dependent on the rated records, while predictions among each other are independent. For example, in Figure 1, suppose $u_3$ and $u_4$ are similar users based on observed ratings, and then $y_{33}$ can be predicted by referring to $r_{43}$, because it is the same item and the two users have high similarity. In the same way, suppose $i_3$ and $i_5$ are observed to have high similarity, and then $y_{45}$ can be predicted by referring to $r_{43}$, because they are similar items by the same user. For simplicity, we suppose no high similarity exists between other items/users pairs, and we do not consider any other relations. In this case, based on traditional CF algorithms, $y_{35}$

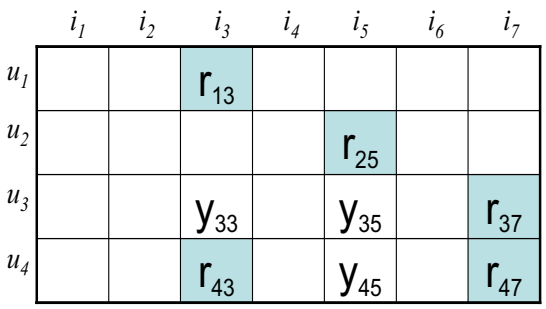



**Figure 1: An illustration example for CF**

cannot be predicted accurately. Because there are no rated items by $u_3$ which is similar to $i_5$ and there is no rating on $i_5$ whose host is similar to $u_3$. Thus no relevant information can be referred to. But if we consider relational dependency within predictions, things are different. As $u_3$ and $u_4$ are similar, $y_{35}$ and $y_{45}$ should be close; as $i_3$ and $i_5$ are similar, $y_{35}$ and $y_{33}$ should be close. So if relational dependency within predictions is utilized, the information of $r_{43}$ can be passed to $y_{35}$ through relational dependency of $y_{33}$, $y_{45}$, and $y_{35}$. In this case, predictions should be generated simultaneously by utilizing the dependency, which let predictions help each other, improving the accuracy. In social recommendations, the data is sparse [30], thus a number of predictions lack of information to refer to, leading to low accuracy. Effectively utilizing relational dependency is indeed important. Previous work, however, did not utilize such information sufficiently. Wang et al.[34] proposed a heuristic method to find $r_{43}$. It has two limitations: (1) It is difficult to measure the similarity between $r_{43}$ and $y_{35}$; and (2) It cannot guarantee the nearness of $y_{35}$ and $y_{33}$ (or $y_{35}$ and $y_{45}$). Ma et al.[20] proposed to firstly predict $y_{33}$ and $y_{45}$, and then to predict $y_{35}$. The problem is that mistakes can propagate from top level to bottom level, which influences the accuracy.

**Being difficult to integrate various features in social network into an unified approach.** In social recommendation, various attributes information and relations have been demonstrated to be effective features. For example, in attribute information, Melville et al.[22] utilized content information (genres, directors, etc.) to boost CF algorithms in movie recommender systems; Nakamoto et al.[23] and Sen et al.[31] employed tag information to improve the accuracy. In relations information, trust relations are utilized effectively in some recent works [1, 3, 9, 21, 24]. These attribute and relation features should be combined to assist predictions in social recommendation. But in traditional CF algorithms, it is hard to combine these features into an unified model. Melville et al.[22] has to convert traditional CF to a classification problem in order to add content features, in which ratings are not predicted. Some of previous work utilized linear integration techniques to smooth feature weights [20]. Consequently, the computing complexity for enumerating values in all spaces to obtain a fitting weight-vector is large when the number of features increases. Thus a framework to globally optimize the weights of multiple effective features should be explored.

Continuous Conditional Random Fields (CCRF) [26] is a desirable approach by going through literatures on solving similar problems mentioned above. CCRF is a recently proposed new model which defines a conditional distribution on predictions of items conditioned on observations. Relational dependency within predictions is modeled in feature functions. CCRF has outstanding advantages comparing to other methods: (1) relational dependency within predictions can be modeled by the Markov property, which is the most general assumption in probabilistic graphical models and has been proven effective in many applications [16]; and (2) feature function weights are globally optimized in CCRF model, which makes it easy to combine various of features. Thus all the two problems aforementioned can be solved based on this approach. Therefore, it is natural to lead us to employ CCRF in social recommendation problems. However, single-scale of CCRF in [26] cannot be directly employed to model different users in recommendations, which will be discussed in detail in Section 3. Therefore in this paper, we extend CCRF model from single-scale to multi-scale in theory, in which each scale corresponds to predictions of a particular user, and apply this new model in social recommendations as a framework to solve the two problems discussed above, which to the best of our knowledge is the first attempt to employ CCRF in recommender systems. The main contributions of this paper include:

1. We formulate the problem of social recommendations and propose a Multi-scale CCRF approach as a framework, extended from single-scale CCRF. In this approach, social relational dependency within predictions is modeled by Markov property. In addition, we combine content and trust relation features into our approach and build an unified model. Experimental results on two real world datasets, Epinions and MovieLens, have demonstrated that our proposed approach performs better than state-of-the-art CF algorithms. Additional experiments are also conducted to show the effectiveness of social relational dependency within predictions and combination of various features.

2. We propose a gradient-based optimization algorithm to train the model and a constrained simulated annealing inference process. Gibbs sampling methods in Markov chain Monte Carlo estimation are employed in both training and inference processes.

The rest of this paper is organized as follows: In Section 2, we introduce the related work. In Section 3, we formulate the problem of social recommendations and present our MCCRF framework. Algorithms are discussed in Section 4 and experimental results together with analysis are given in Section 5. Finally, we summarize this paper in Section 6.

## 2. RELATED WORK

Traditionally, there are two categories of CF methods: memory-based and model-based. The basic idea of memory-based methods (also called neighborhood-based) is that rating predictions for a user depends on other similar users' rated values on the same item or on the current user's previous rated values on other similar items. So approaches are naturally divided into user-based [4, 12, 14] and item-based [5, 18, 30], together with combined approaches [20, 34]. The key point of these methods is the selection of similarity calculation. Typical examples include Pearson Correlation Coefficient (PCC) [27] and Vector Similarity (VS)

[4]. Alternatively, model-based methods [7, 13, 32, 33] are from a probabilistic perspective, which builds a probabilistic model to calculate the expectation of a user's rating on an item. A classical way is to utilize probabilistic latent class [13]. In this approach, latent space exists between users and items, which can be explained as the users' interests or styles. Different latent classes have different distributions on the rating of items, and different users have different distributions on the latent classes. Expectations are calculated as predictions. Some work also combined memory-based and model-based methods into an unified model [25, 36]. Other recent algorithms of CF include [28, 29, 37, 38], etc. The difference of our method from traditional CF methods is that relational dependency within predictions is not utilized sufficiently in most of previous work, but MCCRF models this information using Markov property.

In parallel with the development of CF algorithms is the exploration of effective features in recommendation. At the beginning, only rating information is utilized [4, 5, 12, 13, 30, 33]. However, the data is usually sparse, making it difficult to obtain high accuracy in some cases. Yet some work tried to utilize more features to boost CF algorithms. As mentioned before, the features are divided into attribute features and relational features. Attribute features are descriptions of a single item or a single user, which can be item content (e.g., director, genre in movie recommendations ) [22], tags [23, 31], etc. Relational features, on the other hand, are relationships among item or users, such as user trust information [1, 9, 21, 24]. In previous work, these additional features are combined separately, because under traditional framework of algorithms, it suffers from computing complexity to linearly combine large number of features. But in our approach, since the weights of features are optimized globally, we combine various features into a unified model.

Conditional Random Fields (CRF) is first proposed as a state-of-the-art probabilistic model for segment and labeling sequences data [10, 16]. This model can describe relational dependency in undirected probabilistic graphs, solving the label bias problem. Due to effectiveness in many applications, the theory is widely developed such as Multi-scale CRF [11], Constrained CRF [15, 35], etc. A more detailed tutorial can be found in [8]. Qin et al.[26] first extended conditional random fields from discrete label spaces to continuous label spaces, and applied this CCRF model in "global ranking" tasks. Compared with traditional learning to rank methods relying only on local features of single objects, this method can also model relational dependency among objects to improve ranking. In this paper, we extend this model from single-scale label space to multi-scale label space and apply the new model in social recommendations. We also propose a MCMC-based algorithm to solve the difficulties in training and inference processes.

# 3. SOCIAL RECOMMENDATION FRAMEWORK BASED ON MCCRF

## 3.1 Social Recommendation Formulation

Let $X$ denote observations which can be existing rating records, trust information, similarities between different users/items, profile information of users, etc. Let vector $Y$ denote predictions with $y_{lm}$ denoting the prediction of item $i_m$ by user $u_l$.

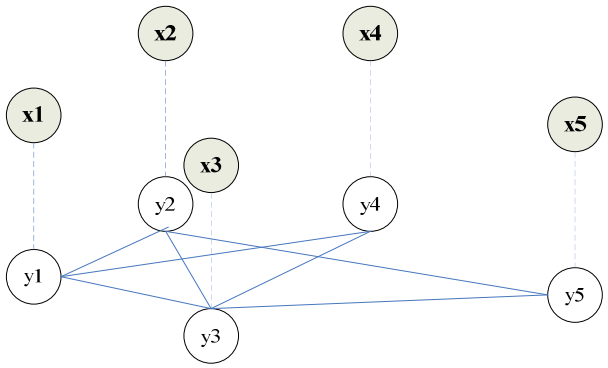



**Figure 2: Probabilistic graph of single-scale CCRF**

We call "local recommendation" or "traditional recommendation", if the problem is formulated as

$$y_{l,m} = f(X).$$

Further more, we call "global recommendation" or "social recommendation", if the problem is formulated as

$$\begin{aligned} Y &= f(X), \ or \\ y_{l,m} &= f(X, y_{-l,-m}), \end{aligned}$$

where $y_{-l,-m}$ denotes all other predictions except $y_{l,m}$.

The major difference of these two formulations is that predictions in social recommendation are dependent on each other conditioned on observations and thus predictions on different items should be generated simultaneously; while in traditional recommendation, predictions are independent. In other words, traditional recommendation is a special case of social recommendation when relational dependency within predictions is removed.

## 3.2 Single-scale CCRF

Single-scale CCRF is proposed by Qin et al.[26], applied in the issue of "global ranking". In this model, a joint conditional probability distribution of a probabilistic graph is defined conditioned on observations. In this section, we explain the model in the application of recommender systems. Please notice single-scale CCRF can only model predictions of a single user and we discuss how to handle multiple users in the next sub-section.

The detailed definition of single-scale CCRF is as follows. Figure 2 gives the probabilistic graph. Let nodes $X(x_1, x_2, ..., x_5)$ denote observations and nodes $Y(y_1, y_2, ..., y_5)$ denote predictions ($y_m$ for item $i_m$). The edge connecting $y_m$ and $y_n$ indicates that relational dependency exits between them in the model. We define the set of nodes connected to $y_m$ by actual line as the "neighbor" of $y_m$, denoted as $neighbor(y_m)$. Since $X$ denotes observations and all values of $Y$ are conditioned on it, we use dotted line to approximately express the relational dependency among $X$ and $Y$. The joint conditional probability density function of predictions $Y$ conditioned on observations $X$ is defined as

$$\begin{aligned} p(Y|X) &= \frac{1}{Z_{sgl}(X)} \exp\left\{ \sum_m \alpha \cdot H(y_m, X) \right. \\ &\quad \left. + \sum_{m,n} \beta \cdot G(y_m, y_n, X) \right\}, \end{aligned}$$

where $H(y_m, X)$ is a local state feature functions vector defined on a local value $y_m$, and $G(y_m, y_n, X)$ is a relational edge feature functions vector defined on the relational dependent values of $y_m$ and $y_n$. $\alpha$ and $\beta$ are function weights vectors to be learned from the training dataset. $Z_{sgl}(X)$ is a normalization factor defined as

$$Z_{sgl}(X) = \int_y \exp\left\{\sum_m \alpha \cdot H(y_m, X) + \sum_{m,n} \beta \cdot G(y_m, y_n, X)\right\} dy.$$

The goal for social recommendation is to find a vector of predictions $Y$ for this user, which can maximize the joint conditional probabilistic distribution of $p(Y|X)$. The feature functions are defined in the quadratic form as:

$$h_{t1}(y_m, X) = -(y_m - x_{m,t1})^2,$$
$$g_{t2}(y_m, y_n, X) = -\frac{1}{2} M_{m,n,t2}(y_m - y_n)^2.$$

In the equations, $t_1$ is state feature function index ranging from 1 to $T_1$ and $t_2$ is edge feature function index ranging from 1 to $T_2$. Here, $x_{m,t1}$ is observed features on item $i_m$ which can be the average rating of $i_m$; $M_{m,n,t2}$ is a relational feature measure which can be the similarity between item $i_m$ and item $i_n$. If we use these two features as an example, it is not difficult to conclude that $p(Y|X)$ will be high if predictions $Y$ fit the following conditions: (1) predictions on item $i_m$ is close to the average rating of item $i_m$; and (2) similar items receive similar ratings predictions. Therefore, relational dependency within predictions for a particular user is described in single-scale CCRF model.

## 3.3 Multi-scale CCRF

Single-scale CCRF cannot model multiple users, because there is only single value for each item, though conditioned relational dependency within predictions is modeled on different items. In this case, all users will be treated the same, which is not reasonable. Besides, what we need to do is not only distinguishing prediction strategies of different users, but also modeling the relational dependency within them. In social recommendation, various relationships (trust information, similarity information, etc) among users are needed to be modeled. Therefore, in this paper, we extend CCRF from single-scale to multi-scale to form a novel model and apply it as a framework in social recommendations to solve aforementioned limitation.

Figure 3 gives the probabilistic graph of MCCRF. In this graph, label space of $Y$ has been extended from single-scale to multi-scale with $y_{l,m}$ denoting prediction on item $i_m$ by user $u_l$. Different scales of $Y$ are drawn in different layers which denote predictions of multiple users. For example, $(y_{11}, y_{12}, y_{13}, y_{14}, y_{15})$ is the rating predictions for user $u_1$, and $(y_{21}, y_{22}, y_{23}, y_{24}, y_{25})$ is for user $u_2$. We still use actual line to denote the relational dependency of predictions $Y$ in the model. In MCCRF, relational dependency exists not only within predictions of the same user (layer), but also within predictions among different users (layers). For example, the prediction of $y_{13}$, has dependent relationship with $\{y_{11}, y_{12}, y_{14}, y_{15}, y_{23}\}$. This example also shows how $neighbor(y_{13})$ (the five dependent nodes) is defined in MCCRF.

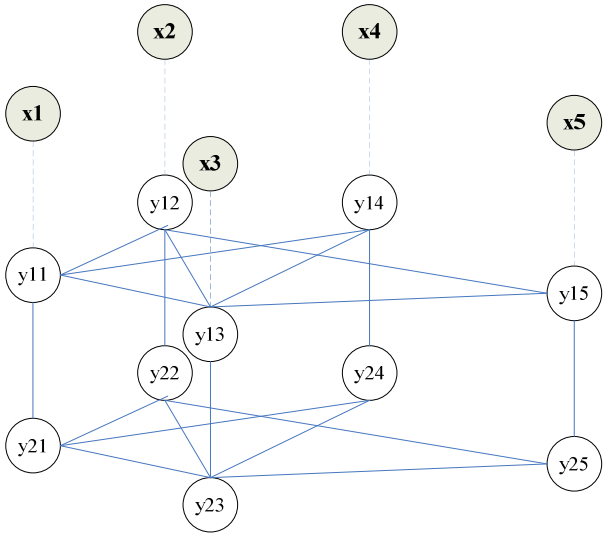



**Figure 3: Probabilistic graph of MCCRF**

In this model, the joint conditional probability density function is defined as

$$p(Y|X) = \frac{1}{Z_{mul}(X)} \exp\left\{\sum_l \sum_m \alpha \cdot H(y_{l,m}, X) + \sum_l \sum_{m,n} \beta \cdot G(y_{l,m}, y_{l,n}, X) + \sum_m \sum_{l,j} \gamma \cdot R(y_{l,m}, y_{j,m}, X)\right\}, \quad (1)$$

where $l$ and $j$ denote different users; $m$ and $n$ denote different items. $H(y_{l,m}, X)$ is a local state feature functions vector defined on local value $y_{l,m}$; $G(y_{l,m}, y_{j,n}, X)$ is a relational edge feature functions vector defined on relational dependent values within the same layer; $R(y_{l,m}, y_{j,m}, X)$ is a relational edge feature functions vector defined on relational dependent values across different layers. $\{\alpha, \beta, \gamma\}$ is feature function weights vectors to be learned from training data. $Z_{mul}(X)$ is the normalization factor defined as

$$Z_{mul}(X) = \int_y \exp\left\{\sum_l \sum_m \alpha \cdot H(y_{l,m}, X) + \sum_l \sum_{m,n} \beta \cdot G(y_{l,m}, y_{l,n}, X) + \sum_m \sum_{l,j} \gamma \cdot R(y_{l,m}, y_{j,m}, X)\right\} dy.$$

The task for social recommendations under this framework is to find the predictions $Y$ that can maximize the joint probabilistic distributions $p(Y|X)$. Feature functions are still defined in the quadratic form as:

$$h_{t1}(y_{l,m}, X) = -(y_{l,m} - x_{l,m,t1})^2,$$
$$g_{t2}(y_{l,m}, y_{l,n}, X) = -\frac{1}{2} M_{m,n,t2}(y_{l,m} - y_{l,n})^2,$$
$$r_{t_3}(y_{l,m}, y_{l,n}, X) = -\frac{1}{2} U_{l,j,t3}(y_{l,m} - y_{j,m})^2.$$

Here, $x_{l,m,t1}$ is observed features of $i_m$ or $u_l$, which can be the average rating of $u_l$; $M_{m,n,t2}$ is a measure of relational feature in the same layer which can be the similarity of $i_m$ and $i_n$; $U_{l,j,t3}$ is a measure of relational feature across different layers which can be the trust relation of $u_l$ and $u_j$ (e.g. the value of $U_{l,j,t3}$ is 1 of $u_l$ trust $u_j$ and is 0 of not). Under this definition of features as an example, it is not difficult to conclude that $p(Y|X)$ will be high if $Y$ fits the following conditions: (1) predictions of a user are close to average rating of the user; (2) predictions on similar items for the same user are close; and (3) predictions of trusted users on the same item are close. Therefore all kinds of relational dependency within predictions have been modeled.

## 3.4 Features

The feature selection in our work is experiment-based. In CRF, features are divided into state features and edge features. Following are the features combined in our model. We will also show the effectiveness of each feature in experimental section.

State Features (The three kinds of state features are only provided in MovieLens dataset):

1. Average rating of an item within users of similar occupation.
2. Average rating of an item within users of similar age and same gender.
3. Average rating of the same genre.

Edge Features (Trust is only contained in Epinions dataset and the other two are in both datasets):

1. Trust information among users: if one user trusts another user, the latter one will be treated as the former one's neighbor.
2. Similarity of users (please refer to [20] for definition): if the similarity between two users is larger than a threshold, an edge is connected between them denoting they are neighbors of each other. Referring to [20], we set the value of this threshold 0.4 for movieLens dataset and 0.2 for Epinions dataset.
3. Similarity of items (please refer to [20] for definition): if the similarity between two items is larger than a threshold, an edge is connected between them denoting they are neighbors of each other. Referring to [20], we set the value of this threshold 0.4 for movieLens dataset and 0.2 for Epinions dataset.

# 4. ALGORITHMS

In this section, we introduce the details of learning and inference processes of MCCRF.

## 4.1 Learning

Parameters learning is to obtain parameter $\{\alpha, \beta, \gamma\}$ which can maximize the log-likelihood from training data $D = \{(x_k, y_k)\}_{k=0}^{N}$, where $x$ is observations and $y$ is predictions. $(x_k, y_k)$ is a training data sample, the setup of which will be explained in the experimental section. In this paper, Gradient Ascent is chosen as optimization method. For simple denotation, we use vector $\lambda$ to denote feature function weights $\{\alpha, \beta, \gamma\}$, and use vector $F(y_k, x_k)$ to denote the value of feature function vectors $\{H, G, R\}$ given $y_k$ and $x_k$. Then, the log-likelihood can be written in

$$\begin{aligned} L_\lambda &= \sum_{k=0}^{N} \log p_\lambda(y_k|x_k) \\ &= \sum_{k}^{N} \left[\lambda \cdot F(y_k, x_k) - \log Z_\lambda(x_k)\right]. \end{aligned}$$

As discussed in [26], to make the integration $Z$ calculable, we must have $\lambda > 0$. Thus it is substituted in algorithm by another variable in order to employ Gradient Ascent optimization method. Let $\lambda = e^{\lambda'}$, where $e^{\lambda'}$ is set by $e_i^{\lambda'} = e^{\lambda'_i}$. Thus

$$L_\lambda = L'_{\lambda'} = \sum_{k}^{N} [e^{\lambda'} \cdot F(y_k, x_k) - \log Z_{e^{\lambda'}}(x_k)].$$

The gradient of the objective function is

$$\nabla L'_{\lambda'} = e^{\lambda'} \cdot \sum_{k=0}^{N} \left[F(y_k, x_k) - E_{p_{\lambda'}(Y|x_k)}\left(F(Y, x_k)\right)\right]. \quad (2)$$

To calculate the expectation term is expensive. In this paper, we propose an approximate estimation method based on Markov chain Monte Carlo. Particularly, we employ Gibbs sampling technique as our method. The main idea is to first sample a sequence of variables $y$ following the distribution of current $p(y|x)$ (this distribution is defined in Eq. (1) and is decided by current $\lambda$). Then, the feature function values of the sequence data $y$ are averaged as the expectation of feature function value denoted as

$$E_{p_\lambda(Y|x_k)}(F(Y|x_k)) = \frac{1}{S}(\sum_{1}^{S} F(\tilde{y}, x_k)), \quad (3)$$

where $S$ is the length of the sequence.

One of the key points for Gibbs sampling is to calculate $p(y_{l,m}|y_{-l,-m}, X)$ in sampling the sequence, where $y_{-l,-m}$ denotes all other predictions except $y_{l,m}$. In our case,

$$P(y_{l,m}|y_{-l,-m}, X) = \frac{P(y_{l,m}, y_{-l,-m}|X)}{\int_{y_{l,m}} P(y_{l,m}, y_{-l,-m}|X) dy_{l,m}}. \quad (4)$$

Under the definition of $p(y|x)$ in Eq. (1), it is not difficult to conclude that $p(y_{l,m}|y_{-l,-m}, X)$ is a Gaussian distribution, the mean and variance of which can be calculated by current $y_{-l,-m}$, $x$ and $\lambda$. Thus the Gibbs sampling methods is feasible in this estimation case by using existing Gaussian distribution sampling methods (in this paper, we use DistLib[1]) as tools. Due to space limitation, please refer to [2, 19] for more details about the theory of Gibbs sampling. The detailed learning algorithm is shown in Algorithm 1.

## 4.2 Inference

Inference is to search predictions that can maximize the joint probability density function conditioned on observations, which is formulated as

$$\hat{y} = \arg\max p(y|x).$$

On this problem of MCCRF, exact estimation is hard to calculate, thus we still consider approximate methods. Generally speaking, Gibbs sampling can be directly used to estimate the optimal solution, however, as discussed in [2], this

[1]http://statdistlib.sourceforge.net

**Algorithm 1** Learning Algorithm for MCCRF

**Input**: Training data $D = \{(x_k, y_k)\}_{k=0}^{N}$,
$U$: number of updating iterations
$S$: number of sampling iterations
**Algorithm**:
  **for** $i = 0$ to $N$-1 **do**
    Load features
    Initialize $\lambda$, y
  **end for**
  Gibbs sampling initialization
  **for** $i = 0$ to $U$-1 **do**
    **for** $k = 0$ to $N$-1 **do**
      **for** $j = 0$ to $S$-1 **do**
        **for** each user-item pair $t$ in $(x_k, y_k)$ **do**
          Sample $y_t$ according to Eq. (1) and Eq. (4)
          Update distributions of $y$ for relevant user-item pairs
        **end for**
      **end for**
    **end for**
    Compute the expectation term according to Eq. (3)
    Compute $\nabla\lambda'$ according to Eq. (2)
    Update $\lambda'' = \lambda' + \eta * \nabla\lambda'$
  **end for**
**Output**: Parameter $\lambda$ of MCCRF model.

**Algorithm 2** Inference Algorithm for MCCRF

**Input**: Testing Data
$T_i$: time control sequence
$S$: number of sampling iterations
$\lambda$: function weights vector
**Algorithm**:
  Load features, $\lambda$, constraints
  Fix predictions of relevant user-item pairs
  Initialize predictions
  Gibbs sampling initialization
  **for** $T = T_0$ to $T_{min}$ according to $T_i$ **do**
    **for** $i = 0$ to $S$-1 **do**
      **for** each user-item pair $t$ **do**
        **if** (prediction is not fixed by constraints) **then**
          Sample $y_t$ according to Eq. (1), Eq. (4) and Eq. (5)
          Calculate $\Delta F$ defined in Simulated Annealing
          **if** $(\min(1, \exp(-\Delta F/T)) > random[0, 1])$ **then**
            Accept $y_t$
            Update relevant distributions
          **end if**
        **end if**
      **end for**
    **end for**
  **end for**
**Output**: Predictions of MCCRF.

method is inefficient because random samples can rarely approach the optimal solution unless $p(y|x)$ has large probability mass around the solution. Thus, in this paper, we employ Simulated Annealing. Using this strategy, the joint conditioned probability function of acceptable sampling data sequence can be controlled by the temperature schema as

$$p_i(\tilde{y}|x) = p^{1/T(i)}(\tilde{y}|x), \tag{5}$$

where $T(i)$ is the temperature at time $i$. When temperature falls, probability mass around the optimal solution will increase, making the sampling process approach to the solution faster. More details about simulated annealing in MCMC are shown in [2, 6, 19].

Utilizing MCMC technique as inference method has another advantage: it is easy to add constraints in the inference process to improve the prediction results. In social recommendations, users usually have rating history on some items, and these ratings can serve as constraints in the inference to assist predictions. In our proposed framework, the constraints can be added into the model by fixing the rated scores in the inference process when sampling. Referring to [10, 15, 17], such process will not destroy the Markov property of the Conditional Random Fields model, and the inference result will be the best one in candidates that can fit the constraints. The detailed algorithm for inference of MCCRF is shown in Algorithm 2.

# 5. EXPERIMENTS

Our experiments are conducted on two real world datasets from MovieLens and Epinions. We aim at verifying the following issues:

1. How about the overall performance of our proposed approach comparing with traditional and state-of-the-art CF methods?
2. How does the relational dependency in predictions affect the accuracy of recommendation results?
3. How do the features we combined from previous work affect the recommendation results?
4. How about the computing complexity of MCCRF?

To Issue 1, we compare our approach with traditional and state-of-the-art CF algorithms in Section 5.4; to Issues 2 and 3, additional experiments are conducted to show the effectiveness of relational dependency and combination of various features in Section 5.5 and Section 5.6. We give analysis of Issue 4 in Section 5.7. Experiments setup is introduced in Section 5.1, Section 5.2 and Section 5.3. In the pre-processing, clustering algorithms are employed, and the impact of cluster size is analyzed in Section 5.8.

## 5.1 Datasets

In this paper, we choose two datasets, MovieLens[2] and Epinions[3] in our experiments for social recommendation. MovieLens is a famous dataset in CF tasks. In this dataset, there are $1,682$ movies and 943 users. Ratings are given on the scale of 1 to 5, with higher value indicating better satisfaction. There are totally $100,000$ rating records in this user-item matrix. The density is

$$\frac{100,000}{1,682 * 943} = 6.3\%.$$

For a single user, there are at least 20 ratings. Some of the statistical results are shown in Table 1. Besides rating information, the dataset also provides other content information. For a movie item, content information includes released date,

[2]http://www.cs.umn.edu/Research/GroupLens
[3]http://www.epinions.com/

**Table 1: Statistics of MovieLens Dataset**

| Statistics | User | Item |
|---|---|---|
| Min. Num. of Ratings | 20 | 1 |
| Max. Num. of Ratings | 737 | 583 |
| Avg. Num. of Ratings | 106.04 | 59.45 |

**Table 2: Statistics of Epinions Dataset**

| Statistics | User | Item |
|---|---|---|
| Min. Num. of Ratings | 1 | 1 |
| Max. Num. of Ratings | 1022 | 2018 |
| Avg. Num. of Ratings | 16.55 | 4.76 |

genre, etc; and for a user, age, gender, occupation are provided. In our approach, genre, occupation, age and gender are combined as content features.

Epinions dataset comes from a consumer review site Epinion.com. In this system, users can give reviews (scale from 1 to 5) to products, being used for future customers as reference and for companies to receive feedbacks or to recommend items. Different from traditional benchmark datasets, Epinions dataset has social trust information among users besides basic rating records. A user can build a trust/distrust list of other users for personalized products ranking as well as indicating users' reputations in the whole social network. Thus it is a good dataset for social recommendation. The whole dataset contains $40,163$ users who rated a total number of $139,529$ different items at least once, writing $664,824$ reviews. The density is

$$\frac{664,824}{40,163 * 139,529} = 0.01186\%.$$

There are totally $487,183$ trust information records in our dataset. The density of trust relationship is

$$\frac{487,183}{2 * C^2_{40,163}} = 0.0302\%.$$

Other statistics are summarized in Table 2.

In both datasets, we randomly group users into four groups, with three groups as training, and the rest as testing. To observe the performances when active users have different number of ratings as history, experiments are conducted by selecting 5, 10 and 15 as rating history for each active user respectively in MovieLens and 2, 5, and 10 in Epinions. We name them Given2, Given5, Given10, and Given15.

## 5.2 Data Sample Building

In this section, we introduce how we build probabilistic graphs on the two datasets. A probabilistic graph represents a data sample $(x_k, y_k)$ in dataset $D = (x_k, y_k)_{k=1}^N$. For MovieLens, since it is small in size, all users and items can be contained in one probabilistic graph. For Epinions, the size is large. For this problem in memory-based CF, Xue et al.[36] proposed a cluster-based method as a solution. By clustering users into small groups, non-similar users are removed in predicting a particular user's evaluations. Thus not only the scalable problem is solved, the accuracy can also be improved. In this paper, we employ similar ideas in our approach. Both users and items are clustered into subgroups, and a probabilistic graph is built on one group of users and one group of items. Referring to [36], we employ K-means algorithm as our clustering algorithm. $K$ is the number of clusters, which is manually defined. In this algorithm, we first randomly select $K$ nodes (users/items) as centroid. All other nodes are assigned into a cluster whose centroid is closest to current node. During iteration processes, the centroid of each cluster is re-calculated based on current nodes in the cluster, and then other nodes are reassigned to adapt the new centroid configuration. In each iteration, the node which has the smallest average distance to other nodes are selected as centroid. Similar to [36], we employ PCC to measure the distance between two nodes. For users, it is defined as

$$Sim(a,u) = \frac{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \overline{r}_a)(r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \overline{r}_a)^2} \sqrt{\sum_{i \in I(a) \cap I(u)} (r_{u,i} - \overline{r}_u)^2}},$$

where $a$ and $u$ denote two users. $I(a)$ and $I(u)$ are the items they have rated. $r_{a,i}$ is the rating of item $i$ by user $a$. $\overline{r}_a$ is the average rating of user $a$. For items, the definition is similar. Due to space limitation, please refer [20] for the details of the definition. In Section 5.8, we will give analysis on the impact of cluster size $K$ in this task.

## 5.3 Metrics

We use Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as our evaluation metrics. MAE is defined as

$$MAE = \frac{\sum |R_{u,i} - \tilde{R}_{u,i}|}{N},$$

where $\tilde{R}_{u,i}$ is the predicted ratings of item $i$ by user $u$, $R_{u,i}$ is the ground truth, and $N$ is the total number of testing predictions. RMSE is defined as

$$RMSE = \sqrt{\frac{\sum (R_{u,i} - \tilde{R}_{u,i})^2}{N}}.$$

In both metrics, lower value indicates higher accuracy.

## 5.4 Overall Performance

To compare our approach with traditional methods, we choose two algorithms (one memory-based and one model-based) as baselines. In memory-based methods, user-based PCC [4] and item-based PCC [30] are widely used. In our baseline, following the idea in [20] which improves the accuracy, we linearly combine these two methods, denoted as EPCC. For model-based methods, generative models are respective. Specifically, Aspect Model (AM) [13] is chosen as baseline. Since our approach belongs to memory-based methods, we choose two state-of-the-art memory-based methods, Similarity Fusion (Fusion) [34] and EMDP [20], for comparison. As stated before, these methods tried to solve similar problems with our approach, but our model have more advantages for solving the error propagation problem.

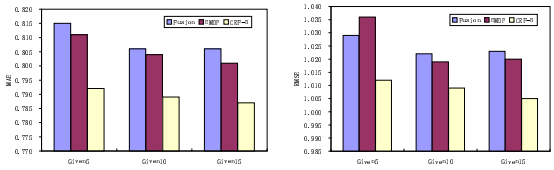
Table 3 and Table 4 shows the overall performance of different methods on MovieLens and Epinions, respectively. Lower MAE and RMSE values indicate better accuracy. On both datasets, we can conclude that MCCRF outperforms traditional and state-of-the-art algorithms. We summarize the improvements from two factors: relational dependency within predictions and combination of various features.

**Table 3: Performance on MovieLens dataset**

| Methods | MAE | | | RMSE | | |
|---|---|---|---|---|---|---|
| | Given5 | Given10 | Given15 | Given5 | Given10 | Given15 |
| EPCC | 0.835 | 0.830 | 0.815 | 1.065 | 1.059 | 1.033 |
| AM | 0.827 | 0.819 | 0.816 | 1.041 | 1.031 | 1.025 |
| Fusion | 0.815 | 0.806 | 0.805 | 1.029 | 1.024 | 1.022 |
| EMDP | 0.811 | 0.804 | 0.801 | 1.036 | 1.019 | 1.020 |
| MCCRF | **0.784** | **0.781** | **0.778** | **0.995** | **0.994** | **0.988** |

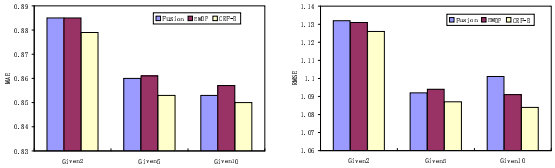
**Table 4: Performance on Epinions dataset**

| Methods | MAE | | | RMSE | | |
|---|---|---|---|---|---|---|
| | Given2 | Given5 | Given10 | Given2 | Given5 | Given10 |
| EPCC | 0.887 | 0.867 | 0.858 | 1.136 | 1.105 | 1.092 |
| AM | 0.893 | 0.885 | 0.863 | 1.132 | 1.131 | 1.101 |
| Fusion | 0.885 | 0.860 | 0.853 | 1.132 | 1.092 | 1.101 |
| EMDP | 0.885 | 0.861 | 0.857 | 1.131 | 1.094 | 1.091 |
| MCCRF | **0.871** | **0.845** | **0.837** | **1.115** | **1.078** | **1.067** |



(a) (b)

**Figure 4: Dependency effectiveness on MovieLens**



(a) (b)

**Figure 5: Dependency effectiveness on Epinions**

## 5.5 Effectiveness of Relational Dependency

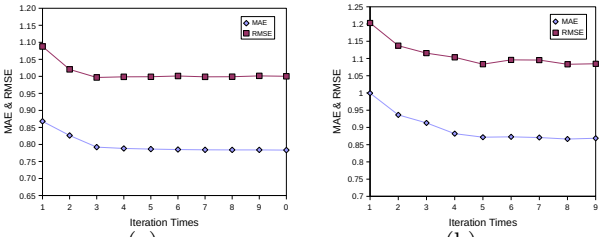
To evaluate the effectiveness of relational dependency in predictions, we conduct experiments with only basic features (CRF-B) of user/item similarities. This means we use the same information comparing with previous work, and the main difference of our approach is that we add relational dependency in predictions. The two state-of-the-art memory-based methods, Fusion method and EMDP method, are chosen for comparisons. Figure 4 and Figure 5 show the experimental results on the two datasets.

From these two figures we can conclude that relational dependency within predictions can improve recommendation results. This is because predictions of user-item pairs can help each other without error propagation. As the data is very sparse in real recommendation systems, utilizing relations in social network sufficiently can improve the accuracy.

## 5.6 Effectiveness of Various Features

To evaluate the effectiveness of various features, we conduct experiments by adding features separately to basic features of user/item similarity. In MovieLens, we conduct experiments by adding occupation features (CRF-BO), age and gender features (CRF-BA), and genre features (CRF-BG). We compare the results with only basic features (CRF-B) and all features (CRF-All). In Epinions, we compare models with (CRF-T) and without (CRF-B) trust information. Figure 6 shows the results in the two datasets (left two: MovieLens, right two: Epinions).

We can observe that each feature we combined (CRF-BO, CRF-BA, CRF-BG, CRF-T) can improve the prediction accuracy comparing to CRF-B. The combination of all features (CRF-ALL, CRF-T) can outperform models with single additional feature.

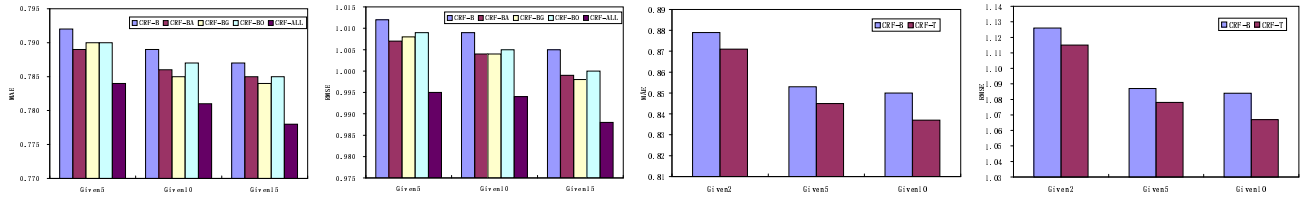



(a) (b)

**Figure 7: Result samples of different iteration times**

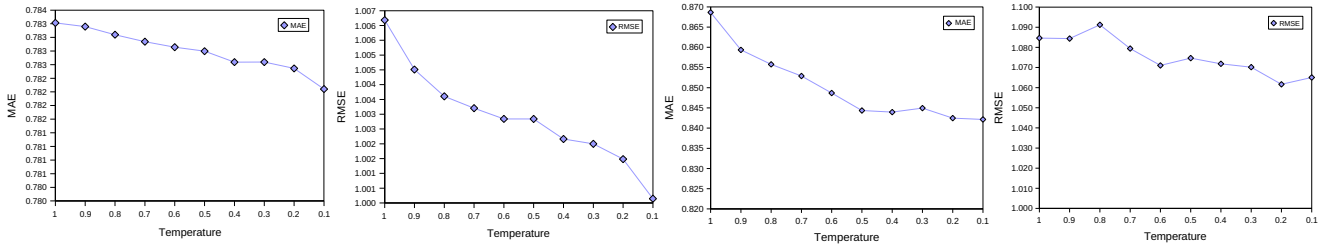
## 5.7 Computing Complexity Analysis

The main computation in our model lies in the sampling process in both training and inferencing. The number of sampling times is the key factor. It is determined by the number of sampling iterations at each temperature and the temperature control schema. Figure 7 shows the results of different iterations in the initialized temperature on two datasets (left: MovieLens; right: Epinions). We can observe after four iterations, the change is not obvious. Figure 8 shows the results in different temperatures (left two: MovieLens; right two: Epinions). According to these results, we set iteration number be 4 and temperature schema from 1.0 to 0.2 with interval of 0.2. Suppose there are $m$ items and $n$ users, the sampling times is $O(m*n)$. Another computation comes from the the updating process of Gaussian distribu-



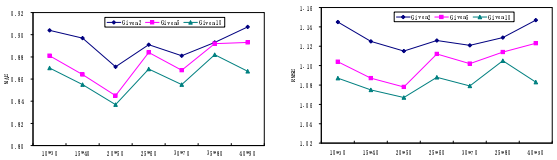
(a) (b) (c) (d)

**Figure 6: Effectiveness of various features**




(a) (b) (c) (d)

**Figure 8: Result samples of different temperature schema**



(a) (b)

**Figure 9: Results on different cluster sizes in Epinions (x-axis: userSize*itemSize)**

tions of user-item pairs. This is decided by the neighbor size of current user-item pair. The neighbor size $s$ can be controlled by adjusting the threshold mentioned in Section 3.4. The updating times for each sample of user-item pair is $O(s)$.

In our experiments, the testing hardware environment is on two Windows workstations with four dual-core 2.5GHz CPU and 8GB physical memory each. The approximate total time for inference in Epinions dataset is 9 hours.

## 5.8 Impact of Cluster Size

As discussed before, we employ clustering techniques as pre-processing. We conduct experiments on different settings to see the impact of cluster size. Figure 9 shows the experimental results. The accuracy increases first and then falls down. This is because at the beginning, there are not enough reference resources. But as the size of a cluster enlarges, non-relevant users/items are included, which influences the accuracy. In our experiments, items are clustered into 50 groups and users are clustered into 20 groups.

# 6. CONCLUSION

In this paper, we have investigated the problem of social recommendation based on CF. Different from traditional recommender systems, various information should be considered in social recommendations. According to limitations of traditional CF algorithms, we extend single-scale CCRF in theory and propose a new model MCCRF as a framework for social recommendation. We also propose MCMC-based methods for training and inference of the model. Experimental results on real world datasets, MovieLens and Epinions, have demonstrated: (1) Markov property in MCCRF is an effective technique to model the relational dependency within predictions. In sparse data, utilizing this kind of dependency can improve recommendation results. (2) Combination of various features into the model can enhance the ability of prediction, which is also the original intention of social recommendations.

# 7. ACKNOWLEDGMENTS

The work described in this paper is supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No.: CUHK 4128/08E and CUHK 4158/08E). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

# 8. REFERENCES

[1] R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In *Proceeding of WWW '08*, pages 199–208, NY, USA, 2008. ACM.

[2] C. Andrieu, N. De Freitas, A. Doucet, and M. Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

[3] P. Bedi, H. Kaur, and S. Marwaha. Trust based recommender system for the semantic web. In *Proceedings of IJCAI 2007*, pages 2677–2682, 2007.

[4] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of UAI '98*, 1998.

[5] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[6] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.

[7] E. Savia. K. Puolamaki, S. Kaski. *Latent grouping models for user preference prediction. Machine Learning*, 74:75–109, 2009.

[8] C. Elkan. Log-linear Models and Conditional Random Fields. In *Tutorial notes at CIKM '08*, 2008.

[9] J. Golbeck. Generating predictive movie recommendations from trust in social networks. In *Proceedings of iTrust '06*, pages 93–104. Springer, 2006.

[10] J. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. *Unpublished manuscript*, 1971.

[11] X. He, R. S. Zemel, and M. A. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *Proceedings of CVPR '04*, volume 2, pages 695–702, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[12] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of SIGIR '99*, pages 230–237, New York, NY, USA, 1999. ACM.

[13] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, 2004.

[14] R. Jin, J. Y. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *Proceedings of SIGIR '04*, pages 337–344, New York, NY, USA, 2004. ACM.

[15] T. T. Kristjansson, A. Culotta, P. A. Viola, and A. Mccallum. Interactive information extraction with constrained conditional random fields. In *Proceedings of AAAI '04*, pages 412–418, 2004.

[16] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML 2001*, pages 282–289, 2001.

[17] S. Li. Markov random field models in computer vision. *Lecture Notes in Computer Science*, 1994.

[18] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, pages 76–80, 2003.

[19] J. Liu. *Monte Carlo strategies in scientific computing*. Springer, 2001.

[20] H. Ma, I. King, and M. Lyu. Effective missing data prediction for collaborative filtering. In *Proceedings of SIGIR '07*, pages 39–46, 2007.

[21] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *Proceeding of SIGIR '09*, pages 203–210, 2009.

[22] P. Melville, R. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of AAAI*, pages 187–192, 2002.

[23] R. Nakamoto, S. Nakajima, J. Miyazaki, S. Uemura, and H. Kato. Investigation of the Effectiveness of Tag-Based Contextual Collaborative Filtering in Website Recommendation. *Lecture Notes in Electrical Engineering*, 4:309, 2008.

[24] J. O'Donovan and B. Smyth. Trust in recommender systems. In *Proceedings of IUI '05*, pages 167–174, New York, NY, USA, 2005. ACM.

[25] D. Pennock, E. Horvitz, S. Lawrence, and C. Giles. Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach. In *Proceedings of UAI*, pages 473–480. Stanford, California, 2000.

[26] T. Qin, T. Liu, X. Zhang, D. Wang, and H. Li. Global Ranking Using Continuous Conditional Random Fields. In *Proceedings of NIPS '08*, 2008.

[27] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Group lens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, 1994.

[28] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Proceedings of NIPS '07*, 2007.

[29] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of ICML '07*, pages 791–798, New York, NY, USA, 2007. ACM.

[30] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of WWW*, pages 285–295. ACM New York, NY, USA, 2001.

[31] S. Sen, J. Vig and J. Riedl. Tagommenders: Connecting Users to Items through Tags. In *Proceedings of the WWW 2009*. Madrid, Spain, 2009.

[32] L. Si and R. Jin. Flexible mixture model for collaborative filtering. In *Proceedings of ICML 2003*, volume 20, page 704, 2003.

[33] L. Ungar and D. Foster. Clustering methods for collaborative filtering. In *AAAI Workshop on Recommendation Systems*, pages 112–125, 1998.

[34] J. Wang, A. De Vries, and M. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of SIGIR '06*, pages 501–508, 2006.

[35] X. Xin, J. Li, J. Tang, and Q. Luo. Academic conference homepage understanding using constrained hierarchical conditional random fields. In *Proceeding of CIKM '08*, pages 1301–1310, New York, NY, USA, 2008. ACM.

[36] G. Xue, C. Lin, Q. Yang, W. Xi, H. Zeng, Y. Yu, and Z. Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of SIGIR '05*, pages 114–121. ACM New York, NY, USA, 2005.

[37] Y. Zhang and J. Koren. Efficient bayesian hierarchical user modeling for recommendation system. In *Proceedings of SIGIR '07*, pages 47–54. ACM New York, NY, USA, 2007.

[38] S. Zhu, K. Yu, and Y. Gong. Stochastic relational models for large-scale dyadic data using MCMC. In *Proceedings of NIPS*, 2008.