# Collaborative Filtering with Entity Similarity Regularization in Heterogeneous Information Networks

**Xiao Yu**[†] **Xiang Ren**[†] **Quanquan Gu**[†] **Yizhou Sun**[‡] **Jiawei Han**[†]
[†] University of Illinois at Urbana-Champaign, Urbana, IL
[‡] Northeastern Univeristy, Boston, MA
[†]{xiaoyu1, xren7, qgu3, hanj}@illinois.edu [‡]{yzsun}@ccs.neu.com

## Abstract

Researchers have been studying hybrid recommender systems which combine user-item rating data with external information in recent years. Some studies suggest that by leveraging additional user and / or item relations, *e.g.*, social network, the performance of the recommendation models can be improved. These studies, nevertheless, mostly utilize a single type of external relationship. Considering the heterogeneity of real-world applications, we propose to position the well-studied recommendation problem in a *heterogeneous information network* context and attempt to incorporate different recommendation factors. We discuss how heterogeneous information network can benefit recommender systems and then propose a matrix factorization based unified recommendation model to take advantage of both rating data and the related information network. Empirical studies show that our approach outperforms several state-of-the-art recommendation methods on explicit rating data.

## 1 Introduction

Heterogeneous information networks, which contain multi-typed entities and relationships, are ubiquitous due to the rapid development of social and information network web services. Researchers begin to incorporate information network analysis techniques into other data mining problems, including classification [Ji *et al.*, 2011], entity search [Sun *et al.*, 2011], clustering and ranking [Sun *et al.*, 2009]. Recent studies have shown that the quality of the recommendation systems can be enhanced by utilizing additional network information such as social network [Ma *et al.*, 2008] or user membership network [Yuan *et al.*, 2011].

However, the above methods mostly only utilize a single type of relationship. It is urgent to develop an unified recommendation framework which can leverage different types of external information due to the heterogeneous nature of the real world. By positioning the recommendation problem in a *heterogeneous information networks* context, it is now possible to study the recommendation problem from different perspectives. The major challenge of utilizing all different

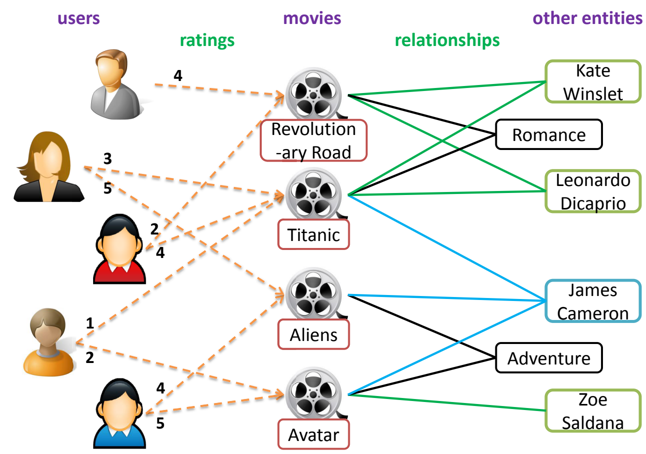



Figure 1: A heterogeneous information network, contains users, movies, actors, and genres as entities, and the corresponding ratings and entity relationships.

types of information in a heterogeneous information network is to (1) combine user rating data with various external information and / or relationships, and (2) learn the importance of each factor. An illustration of heterogeneous information network in the movie domain can be found in Fig. 1. In this example, besides users, movies and the user-movie rating relationships, other types of entities exist in the grander picture as well, such as actors, director, and genres, and relationships between these entities.

In this paper, we adopt the well-studied multi-semantic entity similarity measurements in heterogeneous information network to represent the heterogeneity of the application domains and try to use this technique to facilitate the recommendation process. We propose a matrix factorization based recommendation framework to combine user ratings and various entity similarity matrices defined by meta-path based similarity functions using the information network. We present the objective function as well as the learning algorithm for parameter estimation accordingly. Empirical studies in one real world dataset, IMDb-MovieLens-100K, demonstrate that the proposed recommendation model outperforms several state-of-the-art recommendation methods on this dataset.

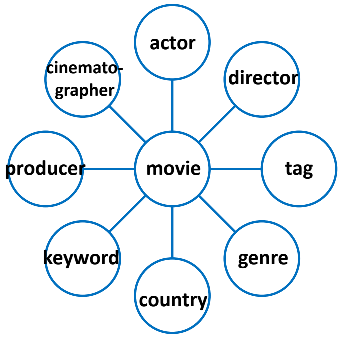



Figure 2: IMDB Network Schema

The major contributions of this paper are summarized as follows:

1. We propose to build a matrix factorization based recommendation framework by considering user ratings as well as a related heterogeneous information network, where different types of information can be used to enhance the performance of the recommender system.
2. We introduce the objective function along with the parameter estimation algorithm to learn the proposed recommendation model.
3. Empirical studies demonstrate the superior performance of our methodology.

The remaining of this paper is organized as follows: We discuss the background and preliminaries of this paper in Section 2. We introduce the proposed unified recommendation model and the learning algorithm in Section 3. Experiments and results are presented in Section 4. Conclusions and future works are presented at the end of the paper.

## 2 Background and Preliminaries

In this section, we present the background and preliminaries of this study. Detailed problem definition is included at the end of this section.

### 2.1 Heterogeneous Information Network

Similar to [Sun *et al.*, 2011] and [Yu *et al.*, 2012], we define information network as follows:

**Definition 1 (Information Network)** *An information network is defined as a directed graph $G = (V, E)$ with an entity type mapping function $\phi : V \to \mathcal{A}$ and a link type mapping function $\psi : E \to \mathcal{R}$. Each entity $v \in V$ belongs to an entity type $\phi(v) \in \mathcal{A}$, and each link $e \in E$ belongs to a relation type $\psi(e) \in \mathcal{R}$.*

We call an information network a *heterogeneous information network* (HIN) when $|\mathcal{A}| > 1$ and / or $|\mathcal{R}| > 1$. To be consistent with recommender system terminology, we refer entities in the information network being recommended as *items*.

Similar to an entity-relation diagram in a relational database, we use an abstract graph (i.e., *network schema*) to represent the entity and relation type restrictions in HIN, denoted by $G_T = (\mathcal{A}, \mathcal{R})$. Examples of heterogeneous information networks and their network schemas can be found in Fig. 2.

### 2.2 Meta-Path-Based Item Similarity

In a network schema of HIN, two entity types can be connected indirectly via different paths, representing relations of different semantic meanings. In order to distinguish path instances, we use the *meta-path* definition from [Sun *et al.*, 2011] to define paths between entity types in network schemas.

**Definition 2 (Meta-Path)** *A meta path $P = A_0 \xrightarrow{R_1} A_1 \xrightarrow{R_2} \ldots \xrightarrow{R_k} A_k$ is a path defined on the graph of network schema $G_T = (\mathcal{A}, \mathcal{R})$, which defines a new composite relation $R_1 R_2 \ldots R_k$ between type $A_0$ and $A_k$, where $A_i \in \mathcal{A}$ and $R_i \in \mathcal{R}$ for $i = 0, \ldots, k$, $A_0 = dom(R_1)$, $A_k = range(R_k)$ and $A_i = range(R_i) = dom(R_{i+1})$ for $i = 1, \ldots, k-1$.*

When measuring similarity between items along different meta-paths, we are able to capture different similarity semantics. For example, in IMDb dataset, to measure similarity between movies, we can utilize meta-paths like movie-actor-movie and movie-keyword-movie, which indicate two different views when considering movie similarity.

[Sun *et al.*, 2011] and [Lao and Cohen, 2010] propose link-based similarity functions to quantitatively measure similarities along different meta-paths in HIN. Specifically, suppose we have $n$ users $\mathcal{U} = \{u_1, \ldots, u_n\}$ and $m$ items $\mathcal{I} = \{e_1, \ldots, e_m\}$, such functions can return a similarity score in $[0, 1]$ to indicate how similar items $e_i$ and $e_j$ are under certain similarity semantic (defined by meta-path $P$). By computing similarity scores of all item pairs along meta-path $P$, we can generate a symmetric similarity matrix, denoted as $S^{(l)} \in \mathbb{R}^{n \times n}$. With $L$ different meta-paths, we can calculate $L$ different similarity matrices with different semantics respectively, denoted as $S^{(1)}, S^{(2)}, \ldots, S^{(L)}$.

### 2.3 Low-Rank Matrix Factorization

Matrix factorization techniques have been used to interpret user ratings in previous studies [Ding *et al.*, 2010], by learning the low-rank representations for users and items. Given user rating matrix $R \in \mathbb{R}^{m \times n}$ denoting $m$ users' numerical ratings towards $n$ items, factorization methods seek to approximate the input matrix $R$ with the multiplication of the $d$-dimensional low-rank representations as follows:

$$R \approx UV^T \tag{1}$$

where $U \in \mathbb{R}^{m \times d}$ and $V \in \mathbb{R}^{n \times d}$ with $d < \min(n, m)$.

The recommendation score between $u_i$ and $e_j$ is computed based on estimated low-rank factors as $r(u_i, e_j) = U_i V_j^T$, where $U_i$ and $V_i$ denote the $i$-th rows of matrix $U$ and matrix $V$, respectively.

Traditionally, Non-Negative Matrix Factorization technique (NNMF) can be employed to solve Equation (1) as follows:

$$\min_{U,V} \|R - UV^T\|_F^2 \ \text{ s.t. } \ U \geq 0, \ V \geq 0, \tag{2}$$

Table 1: Notations

| Notation | Description |
|---|---|
| $u \in \mathcal{U}, e \in \mathcal{I}$ | user, item (or entity) |
| $R, G$ | user rating matrix and information network |
| $S^{(l)}$ | item similarity matrix with semantic $(q)$ |
| $U, V$ | low rank representations of users and items |
| $\Theta$ | recommendation model parameters |

where $\|\cdot\|_F$ denotes matrix Frobenius norm.

Moreover, considering the existence of missing values in rating matrix, researchers extend NNMF to Weighted Non-negative Matrix Factorization (WNMF) method [Zhang *et al.*, 2006] to estimate the low-rank representations as follows:

$$\min_{U,V} \|Y \odot (R - UV^T)\|_F^2 \ \text{ s.t. } \ U \geq 0, \ V \geq 0, \tag{3}$$

where $\odot$ is Hadamard product (element-wise multiplication) between matrices, and $Y$ is an indicator matrix with $Y_{i,j} = 1$ if user $i$ rates the item $j$, and otherwise $Y_{i,j} = 0$.

### 2.4 Problem Definition

We define the recommendation problem which we study in this paper as follows:

**Definition 3 (Problem Definition)** *Given a user rating matrix $R$, and a related heterogeneous information network $G$, for a user $u_i \in \mathcal{U}$, we aim to predict the ratings from $u_i$ to unseen items $e_j \in \mathcal{I}$.*

Notations which are used in the rest of the paper can be found in Table 1.

## 3 Matrix Factorization with Meta-Path-Based Entity Similarity

We present the objective function along with the parameter estimation algorithm of the proposed method in this section.

### 3.1 Recommendation Model

We utilize linear regression model to distinguish different preferences on different meta-path semantics, which leads to a composed similarity matrix $S = \sum_{l=1}^{L} \theta_l S^{(l)}$. Intuitively, value of $\theta_l$ indicates the importance of $l$-th meta path among all $L$ meta paths.

In our study, we adopt graph regularization [Smola and Kondor, 2003] to incorporate the composed item similarity matrix $S$, which aims to achieve local consistency between data features respecting given graph structure. The underlying assumption is that two items $e_i$ and $e_j$ with high composed similarity score $S_{ij}$ are more likely to share similar low-rank representation $V_i$ and $V_j$, respectively, thus have similar ratings whether they are observed or unseen. Following this assumption, the local consistency respecting the composed similarity matrix $S$ can be preserved by positioning graph regularization as $\sum_{l=1}^{L} \theta_l S_{ij}^{(l)} \|V_i - V_j\|_2^2$.

The objective function presented as follows has two parts. The first part aims to learn the low rank representations $U$ for users and $V$ for items, and the second part is to learn the path strengths $\Theta = [\theta_1, \theta_2, ..., \theta_L]^T$ which generate the composed similarity relationships between items.

$$\begin{aligned}
\min_{U,V,\Theta} \quad & \|Y \odot (R - UV^T)\|_F^2 + \lambda_0(\|U\|_F^2 + \|V\|_F^2) + \\
& \frac{\lambda_1}{2} \cdot \sum_{i,j} \sum_{l=1}^{L} \theta_l S_{ij}^{(l)} \|V_i - V_j\|_2^2 + \lambda_2 \|\Theta\|_2^2, \\
\text{s.t.} \quad & U \geq 0, \ V \geq 0, \ \Theta \geq 0, \ \text{and} \ \sum_{l=1}^{L} \theta_l = 1,
\end{aligned} \tag{4}$$

where $\lambda_0$ is the tuning parameter which controls the regularization to avoid over-fitting when learning $U$ and $V$. Similarly, $\lambda_2$ is the regularization parameter on $\Theta$. $\lambda_1$ controls the trade-off between user ratings and similarity matrices.

To simplify the optimization process, we rewrite the graph regularization term (*i.e.*, $\lambda_1$ term) into trace form. For $l$-th similarity matrix, we define the degree diagonal matrix $D^{(l)}$ whose elements $D_{ii}^{(l)} = \sum_{j=1}^{n} S_{ij}^{(l)}$ for $i = 1, ..., n$, and $L^{(l)}$ is the graph Laplacian defined as $L^{(l)} = D^{(l)} - S^{(l)}$. The trace form of $\lambda_1$ term can be derived as follows:

$$\begin{aligned}
& \frac{1}{2} \cdot \sum_{i,j} \sum_{l=1}^{L} \theta_l S_{ij}^{(l)} \|V_i - V_j\|_2^2 \\
= \ & \sum_{l=1}^{L} \theta_l \cdot \Big\{ \sum_{i,j} V_i S_{ij}^{(l)} V_i^T - \sum_{i,j} V_i S_{ij}^{(l)} V_j^T \Big\} \\
= \ & \sum_{l=1}^{L} \theta_l \cdot \Big\{ \sum_{i} V_i D_{ii}^{(l)} V_i^T - \sum_{i,j} V_i S_{ij}^{(l)} V_j^T \Big\} \\
= \ & \sum_{l=1}^{L} \theta_l \cdot \mathrm{Tr}\left(V^T (D^{(l)} - S^{(l)}) V\right) \\
= \ & \mathrm{Tr}\left(V^T (\sum_{l=1}^{L} \theta_l L^{(l)}) V\right).
\end{aligned} \tag{5}$$

By substituting Equation (5), the objective function presented in Equation (4) can be rewrote as follows:

$$\begin{aligned}
\min_{U,V,\Theta} \quad & \|Y \odot (R - UV^T)\|_F^2 + \lambda_0(\|U\|_F^2 + \|V\|_F^2) + \\
& \lambda_1 \cdot \mathrm{Tr}\left(V^T (\sum_{l} \theta_l L^{(l)}) V\right) + \lambda_2 \|\Theta\|_F^2, \\
\text{s.t.} \quad & U \geq 0, \ V \geq 0, , \ \Theta \geq 0, \ \text{and} \ \sum_{l=1}^{L} \theta_l = 1.
\end{aligned} \tag{6}$$

### 3.2 Parameter Estimation

Following similar optimization strategies in [Gu *et al.*, 2010], we derive the multiplicative updating rules for objective function in Equation (6) w.r.t $U$, $V$ and $\Theta$ in the rest of this section.

Optimizing Equation (6) with respect to $U$ is equivalent to optimizing

$$\begin{aligned}
\min_{U} J(U) \quad &= \quad \|Y \odot (R - UV^{\{k\}T})\|_F^2 + \lambda_0 \|U\|_F^2 \\
\text{s.t.} \quad & \quad U \geq 0.
\end{aligned} \tag{7}$$

Multiplicative updating formula for $U$ can be calculated from the derivative of Equation (7) with respect to $U$ and Karush-Kuhn-Tucker complementary condition for the non-negativity of $U$ as follows:

$$U_{ij}^{\{k+1\}} = U_{ij}^{\{k\}} \frac{[(Y \odot R)V^{\{k\}}]_{ij}}{[Y \odot (U^{\{k\}}V^{\{k\}T})V^{\{k\}} + \lambda_0 U^{\{k\}}]_{ij}} \quad (8)$$

Similarly, to learn $V$ at $(k+1)$-th iteration, we simplify the proposed objective function as follows:

$$\begin{aligned} \min_V H(V) &= \|Y \odot (R - U^{\{k+1\}}V^T)\|_F^2 + \lambda_0 \|V\|_F^2 + \\ &\quad \lambda_1 \operatorname{Tr}\left(V^T L^{\{k\}} V\right), \\ \text{s.t.} &\quad V \geq 0, \end{aligned} \quad (9)$$

where $L^{\{k\}} = \sum_{l=1}^{L} \theta_l^{\{k\}} L^{(l)}$ is the graph Laplacian matrix for composed similarity matrix $S$ at $k$-th iteration.

The derivation with respect to $V$ of Equation (9) is:

$$\begin{aligned} \frac{\partial H(V)}{\partial V} &= \left(Y \odot (U^{\{k+1\}}V^T)\right)^T U^{\{k+1\}} - \\ &\quad (Y \odot R)^T U^{\{k+1\}} + \lambda_0 V + \lambda_1 L^{\{k\}} V. \end{aligned}$$

From Karush-Kuhn-Tucker complementary condition for the non-negativity of $V$, we get:

$$\Big[\left(Y \odot (U^{\{k+1\}}V^T)\right)^T U^{\{k+1\}} - (Y \odot R)^T U^{\{k+1\}} + \lambda_0 V + \lambda_1 L^{\{k\}} V\Big]_{ij} \cdot V_{ij} = 0,$$

and updating formula of $V$ for the $\{k+1\}$-th iteration can be derived as follows:

$$V_{ij}^{\{k+1\}} \quad (10)$$

$$= \frac{V_{i,j}^{\{k\}}[(Y \odot R)^T U^{\{k+1\}} + \lambda_1 L^{\{k\}-} V^{\{k\}}]_{ij}}{[(Y \odot (U^{\{k+1\}}V^{\{k\}T}))^T U^{\{k+1\}} + \lambda_0 V^{\{k\}} + \lambda_1 L^{\{k\}+}]_{ij}}$$

where $L = L^+ - L^-$, $L_{ij}^+ = (|L_{ij}| + L_{ij})/2 \geq 0$ and $L_{ij}^- = (|L_{ij}| - L_{ij})/2 \geq 0$.

When optimize Equation (6) with respect to $\Theta$, the problem becomes as follows:

$$\begin{aligned} \min_\Theta F(\Theta) &= \lambda_1 \sum_{l=1}^{L} \theta_l \operatorname{Tr}(L_l V^{\{k+1\}} V^{\{k+1\}T}) + \lambda_2 \|\Theta\|^2 \\ \text{s.t.} &\quad \Theta \geq 0, \text{ and } \sum_{l=1}^{L} \theta_l = 1. \end{aligned} \quad (11)$$

Equation (11) is a well studied quadratic optimization problem with non-negative bound. We use the standard trust region reflective method to update $\Theta$ at each iteration.

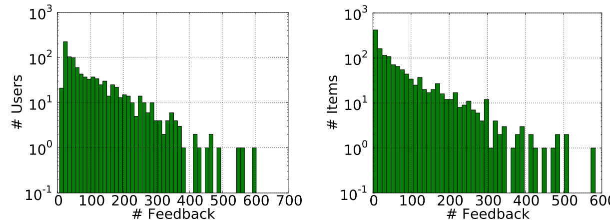
### 3.3 Recommendation By Global Model

Given computed low-rank representation matrices $U$ and $V$, and learned linear model parameters $\Theta$, we can define the global recommendation score for user $i$ on item $j$ as follows:

$$r(u_i, e_j) = U_i V_j^T$$

| Name | #Items | #Users | #Ratings | #Entities | #Links |
|---|---|---|---|---|---|
| IM100K | 943 | 1360 | 89,626 | 60,905 | 146,013 |

(a) Datasets Description




(b) #Ratings vs. #Users  (c) #Ratings vs. Item Popularity

Figure 3: IM100K Dataset

Notice that although we do not directly utilize $\Theta$ when making recommendations, the meta-path similarity matrix selection process embedded in the proposed model serves as regularization terms when learning $U$ and $V$ thus $\Theta$ affects the recommendation results indirectly by adding external information from the related information network to $U$ and $V$. This recommendation function is the same as other matrix factorization based recommendation models.

## 4 Empirical Study

We implemented the proposed recommendation model along with several popularly deployed or state-of-the-art recommendation techniques in this section. We apply these methods on one real-world dataset. We perform a series of experiments to demonstrate the effectiveness of the proposed approach. We present experimental results with discussion and analysis in this section.

### 4.1 Dataset

The dataset we used in empirical studies is built by combining the popular MovieLens-100K dataset and the corresponding IMDb dataset. We name this dataset IMDb-MovieLens-100K (IM100K). When building this dataset, we mapped two datasets using titles and release date of the movies, which could be erroneous on certain movies so the results we presented below are lower-bound of the actual performance. We summarize this dataset in Table 3(a). The distributions of rating numbers can be found in Figure 3.

We random sample training datasets of different sizes (40%, 60% and 80%) and use the rest of the dataset for testing. For each recommendation method, we learn models in trainings dataset and treat ratings in testing sets as unseen.

### 4.2 Competitors and Evaluation Metrics

We implement several widely deployed or state-of-the-art recommendation models as follows:

**UserMean** use mean value of the target user to predict the rating

**ItemMean** use mean value of the target item to predict the rating

Table 2: Performance Comparison

| Metric | MAE | | | RMSE | | |
|---|---|---|---|---|---|---|
| Training Size | 40% | 60% | 80% | 40% | 60% | 80% |
| UserMean | 0.8400 | 0.8409 | 0.8324 | 1.0479 | 1.0482 | 1.0407 |
| ItemMean | 0.8167 | 0.8237 | 0.8130 | 1.0281 | 1.0354 | 1.0235 |
| NNMF (d=40) | 2.1944 | 2.1862 | 2.0162 | 2.4459 | 2.4391 | 2.2915 |
| WNMF (d=10) | 0.7919 | 0.7879 | 0.7589 | 1.0055 | 1.0028 | 0.9677 |
| WNMF (d=20) | 0.7917 | 0.7875 | 0.7591 | 1.0060 | 1.0026 | 0.9681 |
| WNMF (d=40) | 0.7886 | 0.7833 | 0.7569 | 1.0027 | 0.9991 | 0.9655 |
| Hete-MF (d=10) | 0.7838 | 0.7800 | 0.7530 | 0.9950 | 0.9931 | 0.9683 |
| Hete-MF (d=20) | 0.7818 | 0.7802 | 0.7528 | 0.9941 | 0.9938 | 0.9593 |
| Hete-MF (d=40) | **0.7780** | **0.7772** | **0.7400** | **0.9900** | **0.9905** | **0.9503** |

**NNMF** Non-negative matrix factorization function (nnmf) in Matlab on $R$ with d = 40

**WNMF** Weighted non-negative matrix factorization on $R$ with different dimensionality settings (d = 10, 20 and 40) (Equation (3))

We use *Hete-MF* to denote the proposed recommendation model. We utilize 7 different item similarity semantics in this dataset, *e.g.*, movie-actor-movie, movie-keyword-movie, movie-user-movie, movie-genre-movie.

We use Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) to evaluate the performance of different methods. MAE is defined as:

$$MAE = \frac{1}{T}\sum_{i,j}|R_{ij} - \tilde{R}_{ij}| \qquad (12)$$

where $R_{ij}$ denotes the rating $u_i$ gave to $e_j$, and $\tilde{R}_{ij}$ denotes the rating $u_i$ gave to $e_j$ as predicted by a certain method, and $T$ denotes the number of tested ratings. RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{T}\sum_{i,j}(R_{ij} - \tilde{R}_{ij})^2} \qquad (13)$$

### 4.3 Performance Comparison

The performance of of all methods can be found in Table 2. UserMean and ItemMean perform well in this dataset considering the size and the density of the dataset. Based on Figure 3, rating data follow power law distributions, *i.e.*, majority users and majority items only have a small number of ratings. Collaborative filtering based methods usually perform poorly with such users and items while these two simple methods do not suffer from data sparsity issues.

We utilized nnmf function (Non-Negative Matrix Factorization) in Matlab as one of the baseline. We set the dimensionality of the low-rank representations $d = 40$. The performance of this method is worse than UserMean and ItemMean. Without weighted matrix to distinguish observed ratings from unseen ratings and proper regularization terms, matrix factorization can not generate high quality recommendation model.

We also implemented WNMF (Weighted non-negative matrix factorization) (Equation (3)) method. With parameter tuning on regularization terms, the performance of this method can easily beat other baseline methods in all training and testing partitions. One can also notice that a WNMF model with higher dimensionality (d=40) can outperform WNMF model with relatively low dimensionality (d=10) although not by much.

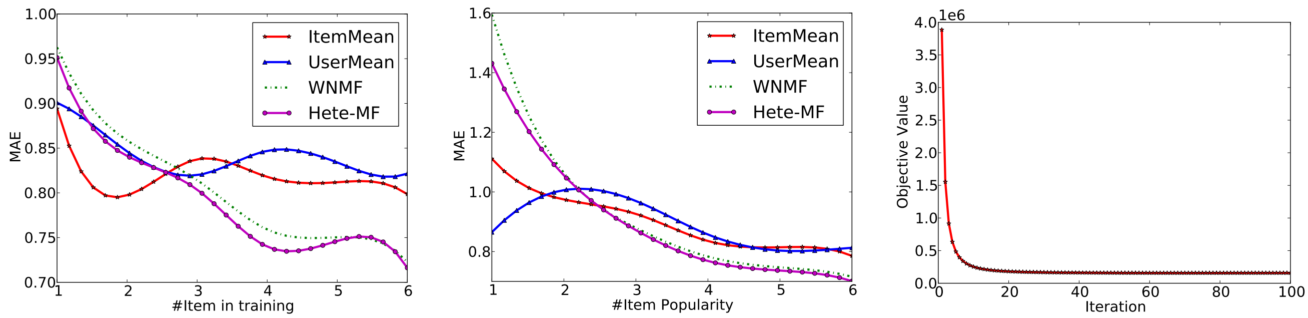
Our proposed Hete-MF recommendation model, which takes advantage of both rating data and the related information network, beats all baseline methods in all experiment settings. This proves our assumption that adding information network as external knowledge with the proposed approach can alleviate the data sparsity issue and improve the recommendation quality.

Besides performance study, we present the objective function convergence process of the proposed method in Figure 4(c). From this figure, one can claim that the proposed objective function converge nicely and the objective value can be reduced rapidly in the first few iterations. This study proves the convergence of Equation (6) and suggests that when estimating parameters, one only need a small number of iterations to obtain high quality recommendation models when using our method.

### 4.4 Analysis and Discussion

We analyze the performance of UserMean, ItemMean, WNMF and Hete-MF methods under different scenarios. We ran the following analytical experiments on $80\%$ training data with $d = 40$ for both factorization based methods.

We first study the correlation between the performances of these methods with the training data size of each user. We split all users into 6 groups based on their training data size. Users in group 1 provided very limited number of feedback (average size is 13) while users in group 6 provided the most amount of feedback (average size is 224). We apply all 4 methods in each group. The results of this study can be found in Figure 4(a). One can notice that for users with small number of training data, UserMean and ItemMean methods outperform factorization based methods on average. Collaborative filtering (CF) based methods catch up when users have more ratings. This study proves the existence of cold-start problem and suggests that by adding rating priors into collaborative filtering methods, we could get even better results. Although both CF methods suffer from cold start, our method




(a) Performance Change with User Training Size (b) Performance Change with Item Popularity (c) Converge Rate

Figure 4: Performance Analysis

performs better than WNMF in almost all user groups.

We then study the correlation between performances of the 4 methods and the popularity of the items that we try to predict. We split items into 6 groups based on the average popularity of the movies in the training dataset. Items in group 1 have the least frequency in training dataset while items in group 6 have the highest frequency. The results of this study can be found in Figure 4(b). Similar to the previous study, UserMean and ItemMean perform well for less popular items while CF methods suffer from data sparsity issues when trying to recommend these items. However when items have sufficient ratings, WNMF and Hete-MF can provide high quality recommendation results as expected. When recommending items with insufficient ratings, the proposed method Hete-MF performs better than WNMF. Although both methods suffer from cold-start problem, based on this study, we can observe that adding information network data as external knowledge can alleviate this issue.

## 5 Conclusion and Future Work

In this paper, we study recommendation in the scope of heterogeneous information network (HIN). We propose a recommendation method on explicit rating data by taking advantage of different item similarity semantics in the network. We compared the proposed approaches with several widely employed or state-of-the-art recommendation techniques, and empirical study demonstrate the effectiveness of our method. We also analyzed the performance of these methods under different scenarios and explained the reasons of the performance drift. Interesting future works include adding user and / or item rating priors to the proposed method to alleviate cold start problem and on-line version of the method to incorporate newly generated ratings.

## Acknowledgements


The work was supported in part by the U.S. Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA), NSF IIS-0905215, CNS-0931975, U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265, and DHS.


## References


[Ding *et al.*, 2010] Chris HQ Ding, Tao Li, and Michael I Jordan. Convex and semi-nonnegative matrix factorizations. *PAMI*, 32(1):45–55, 2010.

[Gu *et al.*, 2010] Quanquan Gu, Jie Zhou, and Chris Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *SDM*, 2010.

[Ji *et al.*, 2011] M. Ji, J. Han, and M. Danilevsky. Ranking-based classification of heterogeneous information networks. In *SIGKDD*, 2011.

[Lao and Cohen, 2010] Ni Lao and William W. Cohen. Fast query execution for retrieval models based on path-constrained random walks. In *Proceedings of SIGKDD*, 2010.

[Ma *et al.*, 2008] H. Ma, H. Yang, M.R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, 2008.

[Smola and Kondor, 2003] Alexander J Smola and Risi Kondor. Kernels and regularization on graphs. In *Learning theory and kernel machines*, pages 144–158. Springer, 2003.

[Sun *et al.*, 2009] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD*, 2009.

[Sun *et al.*, 2011] Y. Sun, J. Han, X. Yan, S. P. Yu, and T. Wu. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. In *VLDB*, 2011.

[Yu *et al.*, 2012] X. Yu, Q. Gu, M. Zhou, and J. Han. Citation prediction in heterogeneous bibliographic networks. In *SDM*, 2012.

[Yuan *et al.*, 2011] Quan Yuan, Li Chen, and Shiwan Zhao. Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation. In *RecSys*, 2011.

[Zhang *et al.*, 2006] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. Learning from incomplete ratings using non-negative matrix factorization. *SDM*, 2006.