Recommendation in Heterogeneous Information Networks with Implicit User Feedback*

Xiao Yu† Xiang Ren† Yizhou Sun‡ Bradley Sturt† Urvashi Khandelwal†
Quanquan Gu† Brandon Norick† Jiawei Han†
† University of Illinois at Urbana-Champaign, Urbana, IL
‡ Northeastern Univeristy, Boston, MA
†{xiaoyu1, xren7, sturt1, khndlwl2, qgu3, bnorick, hanj}@illinois.edu
‡yzsun@ccs.neu.edu

ABSTRACT

Recent studies suggest that by using additional user or item relationship information when building hybrid recommender systems, the recommendation quality can be largely improved. However, most such studies only consider a single type of relationship, *e.g.*, social network. Notice that in many applications, the recommendation problem exists in an *attribute-rich heterogeneous information network* environment. In this paper, we study the entity recommendation problem in heterogeneous information networks. We propose to combine various relationship information from the network with user feedback to provide high quality recommendation results.

The major challenge of building recommender systems in heterogeneous information networks is to systematically define features to represent the different types of relationships between entities, and learn the importance of each relationship type. In the proposed framework, we first use meta-path-based latent features to represent the connectivity between users and items along different paths in the related information network. We then define a recommendation model with such latent features and use Bayesian ranking optimization techniques to estimate the model. Empirical studies show that our approach outperforms several widely employed implicit feedback entity recommendation techniques.

Categories and Subject Descriptors

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

Keywords

Hybrid Recommender System, Information Network

1. INTRODUCTION

Recommender systems, which provide users with recommendations for products or services (generally referred as *items*), have seen widespread implementation in various domains, *e.g.*, e-commerce, video streaming websites, dating services as well as social network services. Existing recommendation approaches utilize the user-item interaction information and / or content information associated with users and items.

Among different recommendation techniques, hybrid approaches build recommendation models by combining user feedback data with additional information, *e.g.*, user or item attributes and relationships. Previous works show that by utilizing additional user or item relationship, the quality of the recommendation models can be improved. Our proposed method falls in the category of such hybrid recommendation systems. The difference between our work and other relationship based hybrid methods is that most previous works only utilize a single type of relationship, *e.g.*, trust relationship [4], friend relationship [6], and user membership [10]. In many real world applications, the entity recommendation problem exists in a heterogeneous environment with different types of information, entity and link attributes and various relationships between users and items. In this paper, we adopt attribute-rich heterogeneous information networks to represent such heterogeneous entity recommendation environment, and propose to study the well established implicit feedback recommendation problem with such data model.

To take advantage of the heterogeneity of the information network, we first diffuse the observed user preferences along different meta-paths to generate possible recommendation candidates under various user interest semantic assumptions. We then employ matrix factorization techniques on the diffused user preferences to calculate the latent representations for users and items accordingly. Each set of latent features represent one recommendation factor with a specific semantic. We then define a recommendation model by combining these recommendation factors and adopt a Bayesian ranking optimization technique to estimate the model accordingly.

The major contributions of this paper are summarized as follows:

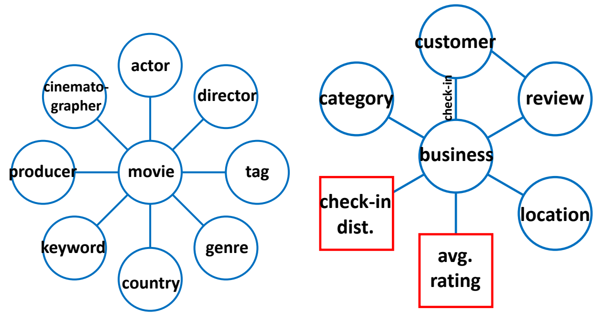
- We propose to build an implicit feedback recommendation framework within an attribute-rich heterogeneous information network, where different types of information and relationships can be used to enhance the quality of the recommender systems.
- We propose a method that can extract latent features to capture different recommendation factors or semantics.
- Empirical studies in two real-world datasets, IMDB-MovieLens-100K and Yelp, demonstrate the power 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 user preference diffusion based latent features and the global recommendation model in Section 3 and 4. Experiments and

*The work was supported in part by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA), the U.S. Army Research Office W911NF-13-1-0193, NSF CNS-0931975, IIS-1017362, DTRA, and DHS.

RecSys'13, October 12–16, 2013, Hong Kong, China.

ACM 978-1-4503-2409-0/13/10 ...$15.00.
.



(a) IMDb Network Schema (b) Yelp Network Schema

Figure 1: Information network schemas: circles represent entity types while rectangles represent attribute types

results are presented in Section 5. 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.

2.1 Binary User Feedbacks

With m users $\mathcal{U} = \{u_1, ..., u_m\}$ and n items $\mathcal{I} = \{e_1, ..., e_n\}$, we define the binary user feedback matrix $R \in \mathbb{R}^{m\times n}$ as follows:

$$R_{ij} = \begin{cases} 1, & \text{if } (u_i, e_j) \text{ interaction is observed;} \\ 0, & \text{otherwise.} \end{cases}$$

Some previous studies have additional assumptions about the implicit feedback dataset, *e.g.*, user-item interaction frequency. Not to digress from the purpose of this study, we use binary user feedback in its original form as defined above.

2.2 Information Network and Meta-Path

We use the definition of heterogeneous information network (HIN) in [9] and [8], denoted by G. 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 Figure 1. In a network schema of HIN, two entity types can be connected via different paths, representing relations of different semantic meanings.We use *meta-paths* to represent path types as defined in [8].

[8] and [5] proposed link-based similarity functions to quantify the similarities along different meta-paths in HIN. Given two items e_i and e_j and a meta-path P, such functions can return a similarity scores in $[0, 1]$ to indicate how similar they are under certain similarity semantic (defined by meta-path). By measuring the similarity of all item pairs with one meta-path, we can generate a symmetric similarity matrix, denoted as $S \in \mathbb{R}^{n\times n}$. With L different meta-paths, we can calculate L different similarity matrices with different semantics accordingly, denoted as $S^{(1)}, S^{(2)}, ..., S^{(L)}$.

In empirical study, to further utilize the heterogeneity of the network, besides meta-path based item similarity, we also use attribute-based entity similarities, *e.g.*, geographical similarity.

2.3 Problem Definition

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

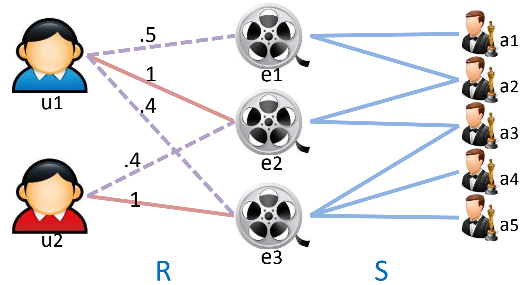


Figure 2: User preference diffusion with meta-path based item similarity matrix (solid red links are observed user preferences while dash links represent diffused preferences)

DEFINITION 1 (PROBLEM DEFINITION). *Given a binary rating matrix R, and a related heterogeneous information network G, for a user $u_i \in \mathcal{U}$, we aim to recommend a ranked list of items $\mathcal{I}_{u_i} \subset \mathcal{I}$ that are of interests to this user.*

3. RECOMMENDATION MODEL

In this section, aiming to alleviate the data sparsity challenge faced by traditional collaborative filtering techniques and utilize the rich yet under-discovered information network, we propose a recommendation approach which combines user feedback and various entity relationship (represented by similarity matrices) derived from the network. We first present a user preference diffusion based feature generation method, and then define a recommendation function with these features. We discuss the recommendation learning algorithm in Section 4.

With the implicit user feedback and item similarity matrices defined in Section 2, we first introduce the diffusion of user preferences based on similarity matrices defined by different meta-paths. Implicit feedback indirectly indicates user preference towards items. Intuitively, if we can find similar items to the ones that the target user was interested in under certain similarity semantic, recommendations to this user could be made by returning the similar items.

More specifically, given a binary user feedback matrix $R \in \mathbb{R}^{m\times n}$ and an item similarity matrix $S^{(q)} \in \mathbb{R}^{n\times n}$, we can diffuse the preference of u_i towards e_j by calculating $\tilde{R}^{(q)}_{ij} = \sum_{t=1}^{n} R_{it} S^{(q)}_{jt}$, which can be rewrite in matrix form for all users as follows:

$$\tilde{R}^{(q)} = RS^{(q)}, \tag{1}$$

where $\tilde{R}^{(q)}$ denotes the diffused user preference matrix. This process propagates user preference along different meta-paths in the item-related HIN and thus is able to find more items a user might be interested in.

We demonstrate the user preference diffusion process with a toy example in Figure 2. In this example, 2 users (u_1 and u_2), 3 movies (e_1, e_2 and e_3), and 5 actors form a small information network. Assume that we observed that u_1 is interested in e_2 and u_2 is interested in e_3 from their previous user log. Under the assumption that users choose movies following meta-path movie-actor-movie, we can claim that u_1 could also be interested in e_1 and e_3 because these two movies are similar to e_2 since they share actors, and u_2 could be interested in e_2 for the same reason. The similarity score between movies along the meta-path movie-actor-movie is calculated with PathSim [8].

With low-rank matrix factorization technique, we derive the low-rank user and item latent features from the diffused matrix $\tilde{R}^{(q)}$ as follows:

$$(\hat{U}^{(q)}, \hat{V}^{(q)}) = \text{argmin}_{U,V} \|\tilde{R}^{(q)} - UV^T\|_F^2 \;\; \text{s.t. } U \geq 0, \; V \geq 0, \tag{2}$$

We apply NMF to solve Equation (2), although more advanced factorization techniques can be used instead. By repeating the above process for all L similarity matrices, we can now generate L pairs of representations of users and items ($\hat{U}^{(1)}$, $\hat{V}^{(1)}$, ..., $\hat{U}^{(L)}$, $\hat{V}^{(L)}$). Each low-rank pair represent users and items under a specific similarity semantics due to the user preference diffusion process. Considering that different similarity semantics could have different importance when making recommendations, we define the recommendation model as follows:

$$\hat{r}(u_i, e_j) = \sum_{q=1}^{L} \theta_q \cdot \hat{U}_i^{(q)} \hat{V}_j^{(q)T} \tag{3}$$

where θ_q is the weight for the q-th user and item low-rank representation pair. Based on the meaning and non-negative property of the features, we add $\theta_q \geq 0$ as a constraint.

4. BAYESIAN RANKING OPTIMIZATION

We discuss the learning process for the recommendation model proposed in Section 3 with user implicit feedback data in this section. Implicit feedback data are often noisy and only contain positive instances. The 0 values in user feedback matrix R are a mixture of negative observations and missing values. The common and traditional approach of parameter estimation is to adopt a classification or learning-to-rank objective function and treats all 1 values in R as positives and 0s as negatives. Theoretically, this approach is incorrect and our empirical study in Section 5 also verifies this claim.

Motivated by [7], we employ a different learning approach by considering the correct item pair orders. We define an objective function aiming to generate the correct order of each item pair in R for each user. The assumption behind this approach is that users are more interested in the items with 1 values in R than the rest of the items, which is a weaker and more plausible assumption compared with the traditional approach.

We use $p(e_a > e_b; u_i|\theta)$ to denote the probability that user u_i prefers e_a over e_b. The Bayesian formulation of the optimization criterion is to maximize the posterior probability as follows:

$$p(\Theta|R) \propto p(R|\Theta)p(\Theta), \tag{4}$$

where $\Theta = \{\theta_1, ..., \theta_L\}$ represents the model parameters, and $p(R|\Theta)$ represent the probability that all item pairs can be ranked correctly for all users according to R, *i.e.*, for each user, items with feedback 1 should be ranked before items with feedback 0.

With the assumption that both user preferences and ordering of item pairs are independent, we can expand the likelihood function $p(R|\Theta)$ as follows:

$$p(R|\Theta) = \prod_{u_i \in \mathcal{U}} \prod_{(e_a > e_b) \in R_i} p(e_a > e_b; u_i|\Theta) \tag{5}$$

where $(e_a > e_b) \in R_i$ represent all item pairs with the correct order in the feedback of u_i.

We define $p(e_a > e_b; u_i|\theta)$ as:

$$p(e_a > e_b; u_i|\Theta) = \sigma(\hat{r}(u_i, e_a) - \hat{r}(u_i, e_b)), \tag{6}$$

where σ is the logistic sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$.

With probability and likelihood definitions above, we can derive the objective function as follows

$$\min_{\Theta} - \sum_{u_i \in \mathcal{U}} \sum_{(e_a > e_b) \in R_i} \ln \sigma(\hat{r}(u_i, e_a) - \hat{r}(u_i, e_b)) + \frac{\lambda}{2} \|\Theta\|_2^2 \tag{7}$$

where $\frac{\lambda}{2}\|\Theta\|_2^2$ is the L_2 regularization term.

Notice that Equation (7) is differentiable. We can utilize gradient descent based optimization methods or the popular BFGS-B method [2] to learn the parameters. The gradient of Equation (7) with respect to Θ can be found as follows:

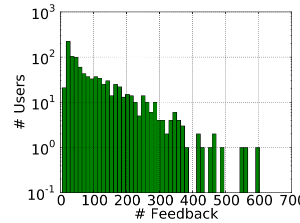
$$\frac{\partial O}{\partial \Theta} = - \sum_{u_i \in \mathcal{U}} \sum_{(e_a > e_b) \in R_i} \frac{e^{-\hat{r}_{i,ab}}}{1 + e^{-\hat{r}_{i,ab}}} \frac{\partial}{\partial \Theta} \hat{r}_{i,ab} + \lambda \Theta,$$

where $\hat{r}_{i,ab} = \hat{r}(u_i, e_a) - \hat{r}(u_i, e_b)$.

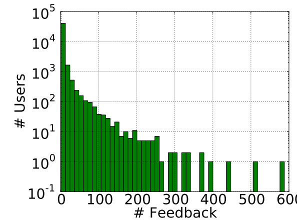
With the above gradient, we employed stochastic gradient descent (SGD) method [1] to estimate the parameters in our empirical studies. The time complexity of this proposed learning process is $O(mn^2)$ where m is the number of users and n is the number of items. In large dataset this can be overwhelming. With SGD, we only need to sample a small number of training instances (around 10^{-5}) and we can still get good results as reported in Section 5.

Name	#Items	#Users	#Ratings	#Entities	#Links
IM100K	943	1360	89,626	60,905	146,013
Yelp	11,537	43,873	229,907	285,317	570,634

(a) Datasets Description



(b) IMDb Feedback Distribution



(c) Yelp Feedback Distribution

Figure 3: IM100K and Yelp Datasets

5. EMPIRICAL STUDY

We implemented the proposed recommendation model along with several popularly deployed or state-of-the-art implicit feedback techniques in this section. We apply these methods on two real-world datasets to demonstrate the effectiveness of the proposed approach. We present experimental results with discussion in this section.

5.1 Data

The first dataset is built by combining the popular MovieLens-100K dataset and the corresponding IMDb dataset. We name this dataset IMDb-MovieLens-100K (IM100K). If a user reviewed a certain movie, we set this feedback as 1, otherwise it would be 0. 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.

The second dataset is the Yelp dataset. When user wrote a review for a restaurant, we set the feedback as 1, otherwise it would be set to 0. Notice that this dataset is much sparser than the IM100K dataset, so the performances of all methods decline accordingly. We summarize these two datasets in Table 3(a). The schema of these two datasets can be found in Figure 1. The distributions of the user feedback can be found in Figure 3.

Both datasets have timestamps with each user item interaction. We split the feedback matrix R of both datasets for each user into training and test based on timestamps, *i.e.*, we use 80% of the "past" feedback to predict 20% of "future" feedback. In Yelp dataset, we filtered out users who have at most 2 reviews since we can not create training and test data for this user.

5.2 Competitors and Evaluation Metrics

We implement several widely deployed recommendation models as follows:

Table 1: Performance Comparison

Method	IM100K				Yelp			
	Prec1	Prec5	Prec10	MRR	Prec1	Prec5	Prec10	MRR
Popularity	0.0731	0.0513	0.0489	0.1923	0.00747	0.00825	0.00780	0.0228
Co-Click	0.0668	0.0558	0.0538	0.2041	0.0147	0.0126	0.01132	0.0371
NMF	0.2064	0.1661	0.1491	0.4938	0.0162	0.0131	0.0110	0.0382
Hybrid-SVM	0.2087	0.1441	0.1241	0.4493	0.0122	0.0121	0.011	0.0337
HeteRec	**0.2094**	**0.1791**	**0.1614**	**0.5249**	**0.0165**	**0.0144**	**0.0129**	**0.0422**

Popularity Recommend the most popular items to users;

Co-Click Estimate conditional probabilities between items and recommend with an aggregated conditional probability calculated with the training data of the target user;

NMF Non-negative matrix factorization on R;

Hybrid-SVM Use SVM based objective function [3] to learn the ranking model, and treat feedback with 1 value as positive and 0 as negative.

We use *HeteRec* to denote the proposed recommendation model. We utilize 9 different item relationships with different similarity semantics in each dataset, *e.g.*, movie-actor-movie, movie-key-movie, biz-user-biz, and biz geographical similarity. Additionally, we add identity matrix as a similarity matrix as well, *i.e.*, items are only similar to themselves. We apply the same feature set in Hybrid-SVM.

For explicit feedback recommendation evaluation, measures like *root mean square error* (RMSE) are the standard evaluation metric. However, these metrics do not suit the definition of implicit feedback problem. In this study, we test all methods as ranking models and use the well-studied information retrieval metrics including precision-at-position and top-10 *mean reciprocal rank* (MRR, Equation (8)) to evaluate and compare the performance of these methods. The MRR metric we used can be found as follows:

$$\text{MRR}_K = \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{e\in test(u_i)}\frac{1}{rank(u_i,e)}\right) \tag{8}$$

5.3 Performance Comparison

The performance of of all 5 methods in the two datasets can be found in Table 1.

Based on Figure 3, user feedback data follow power law distributions, *i.e.*, a very small number of items have interaction with a large number of users. Due to this property, recommending the popular items to users has a decent performance (MRR=0.1923 in IM100K). Co-click method, as one of the most widely deployed technique, achieves MRR = 0.2041 in IM100K and has a similar performance as the NMF method in Yelp (MRR=0.0371).

We implemented the NMF as the CF baseline. We set the dimensionality of the low-rank representations $d = 20$ in IMD100K and $d = 60$ in Yelp. We use the same method and setting when generating features for the proposed approach with Equation (2). With parameter tuning and additional information, NMF may performance better than the results we present in Table 1. However, the same performance improvement can be achieved in our methods accordingly by replacing the NMF solver in Equation (2) with a more advanced technique. As presented in Table 1, NMF outperforms other baselines methods in terms of precision and MRR.

Hybrid-SVM method is a hybrid recommendation approach which uses the same amount of information as our proposed methods. However, if we simply treat all 1 feedback as positive training data and 0 as negative instead of using the objective function proposed in Section 4, the quality of the learned ranking model can be compromised (the overall performance is worse than NMF).

Our proposed HeteRec recommendation model, which takes advantage of both user feedback and the related information network, beats all baseline methods in both datasets. 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. Moreover, the MRR gain of HeteRec compared with NMF in the relatively dense IM100K dataset is 6.1% while it becomes 10.4% in the sparer Yelp dataset. This observation also fits our intuition that when feedback dataset is sparser, the informative knowledge network can improve the performance even more. When training, we employ a uniform sample rate 10^{-5} in SGD and we apply the same rate to all supervised approaches in this experiment. As discussed in Section 4, although the proposed learning method has a high time complexity, if we evaluate the model with a good sampling rate, we can achieve a good balance of effectiveness and efficiency.

6. CONCLUSION AND FUTURE WORK

In this paper, we study entity recommendation problem in HIN. We propose a recommendation model for implicit feedback by taking advantage of different item similarity semantics in the network. We compared the proposed approaches with several widely employed implicit feedback recommendation techniques, and demonstrate the effectiveness of our methods. Interesting future works include personalized recommendation models, similarity feature selection and on-line recommendation model update.

7. REFERENCES

[1] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*. 2010.

[2] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[3] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[4] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *SIGKDD*, 2010.

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

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

[7] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[8] 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.

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

[10] Q. Yuan, L. Chen, and S. Zhao. Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation. In *RecSys*, 2011.