

Learning to Recommend with Hidden Factor Models and Social Trust Ensemble

Dan Zhao

College of Computer Science
Inner Mongolia University
Hohhot, China
csdanna@163.com

Junyi Wang

College of Computer Science
Inner Mongolia University
Hohhot, China
wjyi@imu.edu.cn

Andi Gao

College of Computer Science
Inner Mongolia University
Hohhot, China
csandygao@163.com

Pengfei Yue

College of Computer Science
Inner Mongolia University
Hohhot, China
csypt@imu.edu.cn

Abstract—As one of the most successful approaches to building recommender systems, Collaborative Filtering (CF) uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users. The most successful approaches to CF are latent factor models, Latent Dirichlet Allocation (LDA) models, which directly profile both users and products, and trust-based collaborative filtering models, which analyze the connections among users. This paper introduces some innovations to both approaches. The factor, topic and trust models can now be smoothly merged, to build a more accurate combined model. Further accuracy improvements are achieved by extending the models to exploit both explicit and implicit feedback by the users. The methods are tested on the Yelp data, and the results are better than those previously published on that dataset.

Keywords—recommender systems; latent factor model; latent dirichlet allocation; social network; social trust ensemble

I. INTRODUCTION

With the rapid development of the Internet and the popularity of e-commerce, excessive expansion and redundant information led to people need to spend a lot of time when they looking for the information [3]. Personalized recommendation system can provide an effective mechanism to help users to reduce the time cost of information acquisition process while maintaining the quality of the information obtained, therefore received more and more attention of researchers.

Collaborative filtering is most widely used and successful technology in current recommendation systems [4]. In order to build recommender systems, collaborative filtering algorithm not only need to compare of the user and item factors also need to fuses the users' tastes and their trusted friends' favors together, consider the connections among users. The three main methods of collaborative filtering are factor, topic and trust-based model, respectively. We propose a novel framework, which allows us to fit user accurately and product parameters with only a few feedback and improve the accuracy of recommendations by using collaborative filtering technology.

The remainder of this paper is organized as follows. In Section II presents our work on recommender system with hidden factor models and social trust ensemble. The results of an empirical analysis are presented in Section III, follow by the conclusions and future work in Section IV.

II. PROPOSED FRAMEWORK

The method of collaborative filtering algorithm is forming the numerical score to predict. Although the explicit feedback can meet this requirement, explicit feedback score is usually not sufficient to train an accurate model. For implicit feedback, it does not require additional rating actions, and it's much larger than the explicit feedback. If we find a way can be able to effectively convert implicit feedback to the numerical form in the estimate, which can greatly improve the accuracy of the model which only using the explicit feedback training.

For Yelp dataset, most natural implicit feedback is information of user rating and review behavior history. Other implicit feedback is users' trust friends list. All implicit feedback shows that user preferences for certain items rating and influences of trusted friends' favors on users' tastes. In our experiment, each user u and item i have two sets of data: explicit feedback data set R , contains user-item rating matrix; implicit feedback data set T , contains all the user implicit preference for items of data and the trust relations between users.

A. Models of ratings and reviews

First, consider the user model which titled by 'Hidden factors as Topics', or HFT for short [7]. It aims to combine latent rating dimensions (such as factor models like LFM) with latent review topics (such as topic models like LDA). Unlike supervised topic models, which learn topics are correlated with an output variable, HFT discovers topics that are correlated with the 'hidden factors' of items and users. Topic models operate on documents, so an obvious choice is to define each review as a document D_{ij} (for user u_i and item v_j). For each item v_j we assume that it has a topic distribution $\theta_{j,k}$ (a K -dimensional stochastic vector, each of K topics is discussed across all reviews for that product) and rating 'factors' $V_{j,k}$, which represents the properties that a product possesses. By linking the two factors, we want to get that if a product exhibits a certain property (high $V_{j,k}$), this will correspond to a particular topic being discussed (high $\theta_{j,k}$).

$\theta_{j,k}$ is a K -dimensional stochastic vector, i.e., each of its entries describes a probability that a topic is discussed, while rating factors $V_{j,k}$ can take any value in R^K . So the two cannot be directly converted. By the following conversion formula,

TABLE I. TABLE OF NOTATIONS IN THE HFT

r_{u_i, v_j}	rating of item v_j by user u_i
D_{ij}	review('document') of item v_j by user u_i
$rec(u_i, v_j)$	prediction of the rating for item v_j by user u_i
α	global offset term
β_{u_i}	bias parameter for user u_i
β_{v_j}	bias parameter for item v_j
U_i	K -dimensional latent features for user u_i
V_j	K -dimensional latent features for item v_j
K	number of latent dimensions/topics
θ_i	K -dimensional topic distribution for document d_i
ϕ_k	word distribution for topic k
$w_{u_i, v_j, t}$	t^{th} word of user u_i 's review of item v_j
$z_{u_i, v_j, t}$	topic for the t^{th} word of user u_i 's review of item v_j
N_d	number of words in document d
Θ	$\{\alpha, \beta_{u_i}, \beta_{v_j}, U_i, V_j\}$
Φ	$\{\theta, \phi\}$

which can get monotonic transform between $\theta_{j,k} \in \Delta^K$ and $V_{j,k} \in R^K$:

$$\theta_{j,k} = \frac{\exp(\omega V_{j,k})}{\sum_{k'} \exp(\omega V_{j,k'})}. \quad (1)$$

Where, the denominator enforces that $\theta_{j,k}$ is positive, the numerator enforces that $\sum_k \theta_{j,k} = 1$ and the parameter ω (which fit during learning) to control the 'peakiness' of the transformation. Intuitively, large ω means that users only discuss the most important topic, while small ω means that users discuss all topics evenly.

Now given the hidden factors as topics model, define the objective of a corpus T (rating and reviews) as:

$$y(T|\Theta, \Phi, \omega, z) = \sum_{R_{ij} \in T} \underbrace{(rec(u_i, v_j) - r_{u_i, v_j})^2}_{\text{rating error}} - \mu \underbrace{l(T|\theta, \phi, z)}_{\text{corpus likelihood}} \quad (2)$$

Where, μ is a hyperparameter that trades off the importance of between rating error and corpus likelihood. The function $rec(u_i, v_j)$ is the 'standard' latent-factor model [15]. It is the error of the predicted ratings, which represent as follow:

$$rec(u_i, v_j) = \alpha + \beta_{u_i} + \beta_{v_j} + U_i^T V_j. \quad (3)$$

Where, α is an off-set parameter, β_{u_i} and β_{v_j} are user biases and item biases, and U_i and V_j are K -dimensional vector which represent user factors and items factors (respectively). And the function $P(T|\theta, \phi, z)$ is the 'standard' Latent Dirichlet Allocation model [12]. It is the (log) likelihood of the review corpus, which represent as follow:

$$P(T|\theta, \phi, z) = \prod_{d \in T} \prod_{j=1}^{N_d} \theta_{d, z_{d,j}} \phi_{z_{d,j}, w_{d,j}}. \quad (4)$$

TABLE II. TABLE OF NOTATIONS IN THE RSTE

S	user - neighbor trust matrix
S_{ij}	the trust status of user u_i to neighbor u_j
r_{u_i, v_j}	rating of item v_j by user u_i
r_{\max}	systemprovisions of the highest rating
$\Psi(U', \varepsilon)$	social trust graph
$U' = \{u_i\}_{i=1}^m$	all the users in a social trust network
\mathcal{E}	the trust relations between users
$S' = \{H_{ab}\}$	the $m \times m$ matrix of Ψ
\hat{R}_{ij}	the prediction of the rating of item v_j by user u_i
$T(a)$	the friends set that user u_a trusts
$U \in R^{K \times m}$	latent user feature matrices
$V \in R^{K \times n}$	latent user feature matrices
Γ	$\{R, S', \sigma^2, \sigma_{U'}^2, \sigma_V^2\}$

Where, ϕ_k is word distributions for each topic, θ_d is topic distributions for each document, and $z_{d,j}$ is topic assignments for each word.

HFT model is based on the idea that the factors V_j should accurately model users' rating (as in eq.3), and also that the review corpus should be 'likely' when these factors are transformed into topics (as in eq.4). This model more accurately predicts product ratings by harnessing the information present in review text.

B. Models of social trust ensemble

Nowadays, lots of recommender systems are based on the assumption that users are independent and identically distributed, and ignores the social trust relationships between users, which is not contain the factor which we normally ask trusted friends for recommendations in real life. Based on this intuition, Ma Hao proposed a trust-based model, which is titled 'Recommendation with Social Trust Ensemble', or RSTE for short [8, 20].

According to the different sources of trust, recommendation method based on trust can divided into recommendation based on explicit trust and recommendation based on implicit trust. Normally, explicit trust stored in user-neighbor trust matrix S . Matrix element S_{ab} denote the trust status of user u_a to neighbor u_b . S_{ab} is 0-1 binary representation, 0 represents the default value and 1 represents has trust between users. Because trust is not combined with the users' similarity index, the similarity of tastes between selected trust friend and target user is low, this lead to reduce of recommended accuracy. By analyzing historical user behavior and other features to infer users' implicit trust relationship between them, get even more detailed trust weights between users to update user-neighbors trust matrix [18]. Implicit trust between users is calculated as:

$$H(u_a, u_b) = 1 - \frac{\sum_{j=1}^n |r_{u_a, v_j} - r_{u_b, v_j}|}{r_{\max} \times n}. \quad (5)$$

Where the r_{u_a, v_j} is user u_a ratings for items v_j and r_{\max} represent system provisions of the highest rating. The param-

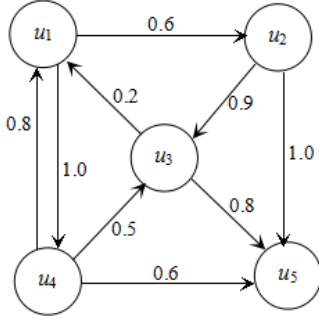


Fig. 1. figure social trust graph

ter n indicates the number of users of common evaluation items.

Suppose we have a directed social trust graph $\Psi = \{U', \varepsilon\}$ (Fig.1), where the vertex set $U' = \{u_i\}_{i=1}^m$ represents all the users in a social trust network and the edge set ε represents the trust relations between users. Let $S' = \{H_{ab}\}$ denote the $m \times m$ matrix of Ψ , which is also called the social trust matrix in this paper. For a pair of vertices, u_a and u_b let $S'_{ab} \in (0,1]$ denote the weight associated with an edge from u_a to u_b , and $S'_{ab} = 0$, otherwise. Note that social trust matrix S' is an asymmetric matrix, since in a trust-based social network, user u_a trusting u_b does not necessary indicate user u_a trusts u_b . Use the following formula to analyze how users' social trust networks affect users' decisions or behaviors:

$$\hat{R}_{aj} = \frac{\sum_{b \in T(a)} R_{bj} S'_{ab}}{|T(a)|}. \quad (6)$$

Where \hat{R}_{aj} is the prediction of the rating that user u_a would give item v_j . R_{bj} is the score that user u_b gave item v_j , $T(a)$ is the friends set that user u_a trusts and $|T(a)|$ is the number of trusted friends of user u_a in the set $T(a)$.

In real life, every user has his/her own taste and at the same time, every user may be influenced by his/her friends he/she trusts. Hence, in order to define the model more realistically, every observed rating in the user-item matrix should reflect both of these two factors. Based on this motivation, model the conditional distribution over the observe ratings as:

$$\begin{aligned} P(U, V | R, S', \sigma^2, \sigma_U^2, \sigma_V^2) &\propto P(R | S', U, V, \sigma^2) P(U | \sigma_U^2) P(V | \sigma_V^2) \\ &= \prod_{a=1}^m \prod_{j=1}^n [N(R_{aj} | g(\alpha' U_a^T V_j + (1-\alpha') \sum_{b \in T(a)} S'_{ab} U_b^T V_j), \sigma^2)]^{I_{aj}^R} \\ &\times \prod_{a=1}^m N(U_a | 0, \sigma_U^2 I) \times \prod_{j=1}^n N(V_j | 0, \sigma_V^2 I) \end{aligned} \quad (7)$$

Where $N(x | \mu, \sigma^2)$ is the probability density function presented by the Gaussian distribution with mean μ and variance σ^2 , and I_{aj}^R is the indicator function, which is equal to 1 if user u_a rated item v_j and equal to 0 otherwise. Let R_{aj} represent the rating of user u_a for item v_j , $U \in R^{K \times m}$ and

$V \in R^{K \times n}$ be latent user and item feature matrices, with column vectors U_a and V_j representing the k -dimensional user-specific and item-specific latent feature vectors of user u_a and item v_j , respectively. The function $g(x)$ is the logistic function $g(x) = 1/(1 + \exp(-x))$, which makes it possible to bound the range of $U_a^T V_j$ within the range $[0,1]$. The users' favors and the trusted friends' favors are smoothed by the parameter α' , which naturally fuses appropriate amount of real world recommendation processes into the recommender systems. The parameter α' controls scale which do users trust themselves or their trusted friends. The sub-formula $P(U | \sigma_U^2)$ and $P(V | \sigma_V^2)$ are zero-mean spherical Gaussian priors on user and item feature vectors.

C. Multi-dimensional recommended model

In Section A, given a potential user interest feature model which called 'Hidden factors as Topics' (HFT), while in Section B, given a user trust model which called 'Recommendation with Social Trust Ensemble' (RSTE). Linear combination of the two models, get a new model, we call it multidimensional recommended model (MRM).

$$f(T, U, V | \Theta, \Phi, \omega, z, \Gamma) = \chi y(T | \Theta, \Phi, \omega, z) + (1 - \chi) P(U, V | \Gamma) \quad (8)$$

In order to solve the data sparseness, cold start and other issues, we propose this model, which is based on the social network, and the history feedback information of users, improvement on the traditional collaborative filtering recommendation method. Next, we will on the experimental data, verify the validity of the model.

III. EXPERIMENTS

In this section, we intended to address the following questions: (1) How does the model parameter χ affect the accuracy of prediction (Sec. D)? (2) How does our approach compare with the published state-of-the-art collaborative filtering (Sec. E)? (3) Can our algorithm achieve good performance even if users have few observed rating records (Sec. F)?

A. Datasets

We choose Yelp datasets for experimentations. The Yelp dataset consists of user reviews on the businesses located in the Phoenix City of the US. It include 1.6M reviews by 366K users for 61K businesses and social network of 366K users for a total of 2.9M social edges from the recently proposed Yelp Dataset Challenge¹.

B. Metrics

We use two metrics, the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE), to measure the prediction on quality of our proposed approach in comparison with other collaborative filtering.

The metrics MAE is defined as:

$$MAE = \frac{\sum_{i,j} |r_{i,j} - \hat{r}_{i,j}|}{N} \quad (9)$$

Where $r_{i,j}$ denotes the rating user i gave to item j , $\hat{r}_{i,j}$

¹http://www.yelp.com/dataset_challenge/

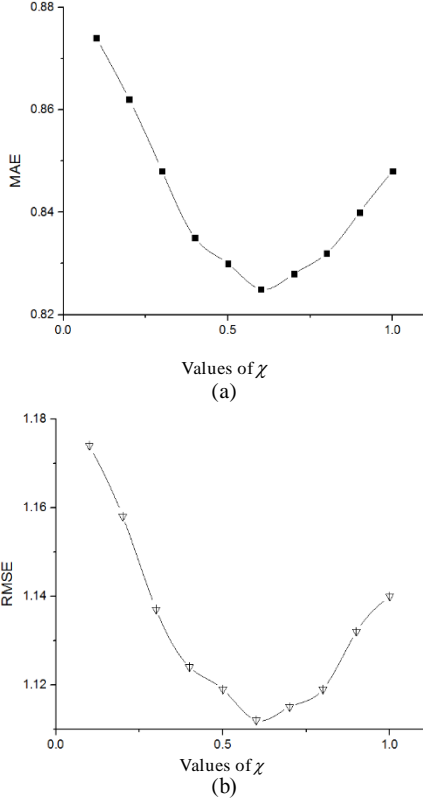


Fig. 2. Impact of parameter χ

denotes the rating user i gave to item j as predicted by a method, and N denotes the number of tested ratings. The metrics RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{i,j} (r_{i,j} - \hat{r}_{i,j})^2}{N}} \quad (10)$$

C. Evaluation procedure

We randomly select 90% of the ratings from Yelp dataset as the training data to predict the remaining 10% of ratings. In the model HFT, offset and bias terms $\alpha, \beta_{u_i}, \beta_{v_j}$ are initialized by averaging ratings and residuals; other parameters are initialized uniformly at random. Parameters for HFT model are fit using L-BFGS [19], which run for 2,500 iterations. Multiple performing gradient descent training, the parameter settings of model RSTE are $\alpha' = 0.4$ for training data, $U_i = V_j = 0.001$. We will multiple performing iterations to find the best parameters χ , let the MAE and RMAE for our model whose error on the test set is lowest.

D. Impact of Parameter χ

In our method proposed in this paper, the parameter χ balances the information from the history feedback information of users and the influence of the social trust network. If $\chi=1$, we only consider the history feedback information like rating and review of item v_j by user u_i . If $\chi=0$, we only fuse information from the user-item rating matrix and the user social trust network for probabilistic matri-

TABLE III. PERFORMANCE COMPARISONS

Metrics	K=5		
	RSTE	HFT	MBM
MAE	0.8744	0.8438	0.8235
RMSE	1.1721	1.1358	1.1116

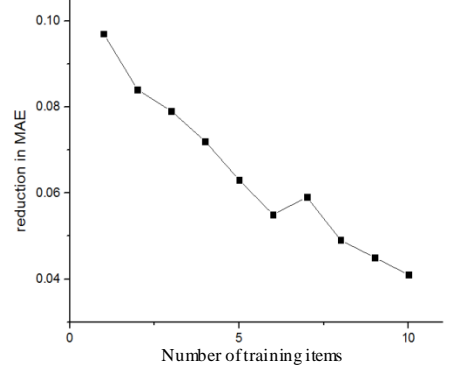


Fig. 3. Recommending new products

x factorization. In other cases, we fuse information from the user's own characteristics, history feedback, and his/her social trust network, furthermore, to predict ratings for users.

Recommending new products Fig.2 shows the impacts of χ on MAE and RMSE. We observe that the value of χ impacts the recommendation results significantly which demonstrates that fusing the history feedback information of users and the effect of the social trust network greatly improves the recommendation accuracy. As χ increases, the MAE and RMAE decrease (prediction accuracy increases) at first, but when χ surpasses a certain threshold, the MAE and RMSE increase with further increase of the value of χ . This phenomenon confirms with the intuition that purely using the history feedback information of users or the users' social network for recommendations cannot generate better performance than fusing these two aspects together.

We observe that, our MRM method achieves the best performance when χ is around 0.6, while smaller values like $\chi=0.2$ or larger values like $\chi=0.7$ can potentially degrade the model performance. So the following experiment, we fixed the parameters χ for 0.6.

E. Comparison

In this section, in order to show the performance improvement of our MBM model, we compare our method with the following approaches.

1) *HFT*: This method is proposed by Julian McAuley and Jure Leskovec in [5]. It only combines ratings with review text for product recommendations.

2) *RSTE*: This is the method proposed in [6]. It is a model which naturally fuses the users' tastes and their trusted friends' favors together.

The experimental results using 5 dimensions to represent

the latent features are shown in Table III. From Table III, we can observe that our approach MRM outperforms the other methods on both MAE and RMSE. We gain improvements of up to 5.8%, 2.4% over RSTE methods and HFT methods, respectively. This demonstrates that our interpretation on the formation of the ratings is reasonable.

F. Recommending new products

The ‘cold-start’ problem is a common issue in recommender system. In particular, when a product or a user is new and one does not have enough rating data available, it is very hard to train a recommender system and make predictions.

When given only a few ratings for a particular user u_i or product v_j , a latent factor recommender cannot possibly estimate U_i or V_j accurately due to lack of data. However, review text and social trust network provides significant additional information and our hope is that by including review text and social trust network the relationship between products, users and ratings can be more accurately modeled. Our hypothesis is that when training data for a product is scarce (e.g. when a new product appears in the corpus), the benefit gained from using review text and social trust network will be greatest.

In Fig.3 we compare the amount of training data (x-axis) to the improvement gained by using MBM (y-axis). Specifically, we report the MAE of a latent-factor recommender system minus the MAE of MBM; thus a positive value indicates that MBM has better performance. Notice that our model gains significant improvement when training data is particularly scarce.

IV. CONCLUSIONS AND FUTURE WORK

We have presented MBM, a model that fusing the history feedback information of users and the effect of the social trust network for product recommendations. This model not only to improve the accuracy of recommendations also allows us to accurately fit user and product parameters with only a few reviews and a little social trust network, which existing models cannot do using only a few ratings. In this paper, we only evaluate user similarities between directly connected users. We can also further analyze the similarities between users who are multi-hop connected. We believe this will give us another point of view on understanding the correlations between social relationships and user interest similarities.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their valuable suggestions. This work is in part supported by the China National Natural Science Foundation under award number 61063018.

REFERENCES

- [1] X. Hu, X. Meng, Y. Zhang, and Y. Shi. Recommendation algorithm combining item features and trust relationship of mobile users. *Ruan Jian Xue Bao/Journal of Software*, 2014, 24(8):1817-1830.
- [2] H. Ma, On Measuring Social Friend Interest Similarities in Recommender Systems. *SIGIR*, 2014.
- [3] Q. Lu, R. Wang, and J. Zhang et al. Recommender algorithm combined with neighborhood model and LFM. *Computer Engineering and Applications*, 2013,49(1):100-103.
- [4] J. Ding, and J. Liu. Recommendation Algorithm Based on User Nearest Neighbor Model and Matrix Factorization. *Industrial Control Computer*, 2015.
- [5] J. Vosecky, K. Wai-Ting Leung, and W. Ng. Collaborative Personalized Twitter Search with Topic-Language Models. *SIGIR*, 2014.
- [6] Y. Shi, X. Zhao, J. Wang, M. Larson, and A. Hanjalic. Adaptive Diversification of Recommendation Results via Latent Factor Portfolio. *SIGIR*, 2012.
- [7] J. McAuley and J. Leskovec. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. *RecSys*, pages 165-172, 2013.
- [8] H. Ma, I. King, and M. R.Lyu. Learning to Recommend with Social Trust Ensemble. *SIGIR*, 2009.
- [9] T. Zhu, P. Harrington, J. Li, and L. Tang. Bundle Recommendation in eCommerce. *SIGIR*, 2014.
- [10] J. Lin, K. Sugiyama, K. Min-Yen, and C. Tat-Seng. New and Improved: Modeling Versions to Improve App Recommendation. *SIGIR*, 2014.
- [11] Z. Huang, W. Chung, T. H. Ong, and H. Chen. A graph-based recommender system for digital library. In *Proc. of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 65-73, 2011.
- [12] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993-1022, 2003.
- [13] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Matrix Factorization and Neighbor based Algorithms for the Netflix Prize Problem. In *Proc. of the 2nd ACM Conference on Recommender Systems (RecSys'08)*, pages 267-274, 2008.
- [14] Y. Koren, R. Bell, and C. Volinsky. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer*, 42:30-37, 2009.
- [15] Y. Koren and R. Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*. Springer, 2011.
- [16] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Explicit Factor Models for Explainable Recommendation based on Phrase-level Sentiment Analysis. *SIGIR*, 2014.
- [17] X. Wei and W. Bruce Croft. LDA-Based Document Models for Ad-hoc Retrieval. *SIGIR*, 2006.
- [18] N. Lathia, S. Hailes, and L. Capra. Trust-Based Collaborative Filtering. *Trust Management II*, 2008.
- [19] J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 1980.
- [20] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.