PsRec: Social Recommendation with Pseudo Ratings

Yitong Meng
The Chinese University of Hong Kong
yitongmeng@cse.cuhk.edu.hk

Guangyong Chen
Tencent Quantum Lab
gycchen@tencent.com

Jiajin Li
The Chinese University of Hong Kong
jjli@cse.cuhk.edu.hk

Shengyu Zhang
Tencent Quantum Lab
The Chinese University of Hong Kong
shengyzhang@tencent.com

ABSTRACT

Data sparsity and cold start are two major problems of collaborative filtering based recommender systems. In many modern Internet applications, we have a social network over the users of recommender systems, from which social information can be utilized to improve the accuracy of recommendation. In this paper, we propose a novel trust-based matrix factorization model. Unlike most existing social recommender systems which use social information in the form of a regularizer on parameters of recommendation algorithms, we utilize the social information to densify the training data set by filling certain missing values (handle the data sparsity problem). In addition, by employing different pseudo rating generating criteria on cold start users and normal users, we can also partially solve the cold start problem effectively. Experiment results on real-world data sets demonstrated the superiority of our method over state-of-art approaches.

CCS CONCEPTS

• **Information systems → Social recommendation**;

KEYWORDS

Recommender systems; social network; matrix factorization

1 INTRODUCTION

We aim to utilize social network information to alleviate data sparsity and cold start problems of recommender systems. Data sparsity refers to the case that the rating matrix has only a few observed ratings, while cold start means that there exist new users who rarely rate any items.

1.1 Data Sparsity

One issue brought by data sparsity is overfitting, which often occurs when a model is excessively complex. A model that has been overfitted has poor predictive performance, as it overreacts to minor fluctuations in the training data. A common solution for overfitting is to reduce the ratio of model complexity and training data size.

Most previous work handles the data sparsity problem by reducing the model complexity. The available social information is incorporated into the loss function as a regularization term in the rating matrix factorization $\mathbf{R} \approx \mathbf{UV}$ [8–10, 16, 18]. These regularization terms narrow the solution space of the user-feature matrix $\mathbf{U}$, thereby reducing the model complexity and enhancing the prediction accuracy.

In this paper we take a different approach to handling the data sparsity issue while holding the model complexity unchanged. We densify the training data by filling certain missing values in the rating matrix with *pseudo ratings*, inferred from contextual information provided by the social network. The pseudo ratings are generated based on a simple observation: a user tends to agree with her friends' average taste on an item if most of her friends have a similar taste on it, a phenomenon which has been well observed and studied in social science literature [17].

Our pseudo-rating generation goes beyond the widely used heuristics of recommending the most popular items (largest number of high ratings) to a user [13], which is not personalized, and merely utilizes information in the most popular items. It turns out that the most unpopular items (largest number of low ratings) can also provide useful information about a user's taste, which our algorithm also uses. In addition, the heuristics only gives rating prediction on popular items, while ours uses pseudo ratings to assist matrix factorization in the next step, which gives prediction on all user-item pairs.

1.2 Cold start

The cold start issue happens when some users only give very few ratings. Since the information provided by such users is even less than average, it is more likely to get overfitted. In order to improve prediction accuracy on cold start users, we treat cold start users and normal ones differently in training, by generating more pseudo ratings for cold start users. Experimental results show that this method improves the prediction accuracy on cold start users by a large extent.

To facilitate the idea of filling pseudo ratings, one needs a careful implementation. We construct an f-function to measure the distance between a user's rating and her friends' average rating. The predicted distance serves as a criteria for filling pseudo ratings.

RecSys '18, October 2–7, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5901-6/18/10…$15.00
https://doi.org/10.1145/3240323.3240390

2 RELATED WORK

There are a number of social recommender systems proposed in the past decade, summarized as follows. Ma *et al.* proposed a regularization method SoRec [9] by sharing a common user-latent matrix factorized by ratings and by trusts. SoReg [10] and SocialMF [8] use social information by assuming the user-specific vector to be close to the average of those for her trusted neighbors. TrustMF [18] factorizes the user trust matrix by truster-feature matrix and trustee-feature matrix. TrustSVD [6] incorporates social information in the classic SVD++ model. Shmueli *et al.* use social data to enhance recommendations by computing latent vectors of the users' friends [15].

Melville *et al.* employed the idea of densifying training data by using contextual information in a Content-Boosted CF algorithm [11]. The contextual information used there is content information of items rather than social information of users.

Golbandi *et al.* handle cold start problem by employing short interviews to solicit new users to provide feedback [4, 5], which requires additional user efforts. However, we handle the cold start problem by just analyzing existing data without requiring new users to pay additional efforts.

3 THE PSREC MODEL

3.1 Problem Definition and Notation

We first introduce notation used in this paper. Let $\mathcal{U} = \{u_1, u_2, ..., u_m\}$ be a set of m users, and $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ a set of n items. The user-item rating matrix is denoted by $\mathbf{R} \in \mathbb{R}^{m\times n}$, where $\mathbf{R}_{ij} > 0$ is the rating given by user u_i to item v_j. We use 0 to represent the missing values in $\mathbf{R}$. Users have social connections among each other, and we use $\mathbf{T}$ to denote user-user trust matrix, where $\mathbf{T}_{ij} := \mathbb{P}(u_i \text{ trusts } u_j)$, the probability that u_i trusts u_j. $\mathbf{T}_{ij} > 0$ if u_j is u_i's friend. Note that the trust matrix $\mathbf{T}$ may be asymmetric.

We are looking for a user-latent matrix $\mathbf{U} \in \mathbb{R}^{d\times m}$ and an item-latent matrix $\mathbf{V} \in \mathbb{R}^{d\times n}$ for some small d, such that they can recover the rating matrix $\mathbf{R}$: $\mathbf{R} \approx \mathbf{U}^T\mathbf{V}$. Hence, the rating of user u_i given to item v_j can be predicted by $\widehat{\mathbf{R}}_{ij} = \mathbf{U}_i^T\mathbf{V}_j$, where $\mathbf{U}_i$ and $\mathbf{V}_j$ are the i-th and j-th column of matrix $\mathbf{U}$ and $\mathbf{V}$ respectively. Formally, we can learn the user- and item-latent matrices by minimizing the mean square error over the observed ratings:

$$\mathbf{U}, \mathbf{V} = \underset{\mathbf{U},\mathbf{V}}{\operatorname{argmin}}\{\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R(\mathbf{R}_{ij} - \mathbf{U}_i^T\mathbf{V}_j)^2 + \lambda||\mathbf{U}||_F^2 + \lambda||\mathbf{V}||_F^2\},$$

where $I_{ij}^R = 1$ if $\mathbf{R}_{ij}$ is an observed rating in $\mathbf{R}$, and 0 otherwise, and $||\cdot||_F^2$ is the Frobenius norm of matrices.

Next we introduce four quantities used in our algorithm.

(1) $\mathbf{F}_{ij} := \mathbb{E}[\sum_{k\neq i}(\mathbb{1}_{\{u_i \text{ trusts } u_k\}} I_{kj}^R)] = \sum_{k\neq i}\mathbf{T}_{ik}I_{kj}^R$, which equals to the sum of trust values of u_i's friends who rates item v_j. Specifically, when $\mathbf{T}_{ij}$ is either 1 or 0, $\mathbf{F}_{ij}$ is exactly the number of u_i's friends who rates item v_j.

(2) $\mathbf{M}_{ij} := \frac{\sum_{k\neq i}(\mathbf{T}_{ik}I_{kj}^R)\mathbf{R}_{kj}}{\mathbf{F}_{ij}}$ is the weighted average rating of u_i's friends on v_j. $\mathbf{M}_{ij}$ is undefined where $\mathbf{F}_{ij}$ is 0. A more trusted friend of u_i contributes more in $\mathbf{M}_{ij}$.

(3) $\mathbf{S}_{ij} := \sqrt{\frac{\sum_{k\neq i}(\mathbf{T}_{ik}I_{kj}^R)(\mathbf{R}_{kj}-\mathbf{M}_{ij})^2}{\mathbf{F}_{ij}}}$ is the *weighted* standard derivation of u_i's friends' ratings (not including u_i's rating) on item v_j.

(4) $\mathbf{D}_{ij} := |\mathbf{R}_{ij} - \mathbf{M}_{ij}|$ is the distance between a user's rating and his friends'. A small $\mathbf{D}_{ij}$ indicates that u_i has a similar taste with his friends on v_j.

3.2 Model assumption

We make one assumption based on the following general observation: a user tends to agree with his friends' average taste on an item if most of his friends have similar tastes on it.

Assumption 1. *$\boldsymbol{D}_{ij} = f(\boldsymbol{F}_{ij}, \boldsymbol{S}_{ij}) + e_{ij}$, where e_{ij} is a noise term with mean 0, and $f(x, y)$ is a nonnegtive function on $\mathcal{D} = \{(x, y)|x \in (0, +\infty), y \in [0, +\infty)\}$, satisfying that **(1)** $\frac{\partial f}{\partial x} < 0$, **(2)** $\frac{\partial^2 f}{\partial x^2} > 0$, and **(3)** $\frac{\partial f}{\partial y} > 0$ for $(x, y) \in \mathcal{D}$.*

The quantity $f(\mathbf{F}_{ij}, \mathbf{S}_{ij})$ predicts the distance between the pseudo rating and the ground truth on entry (i, j). To gain a quick understanding of the assumption, let us consider the special case of $\mathbf{T}_{ij}$ being either 1 or 0.

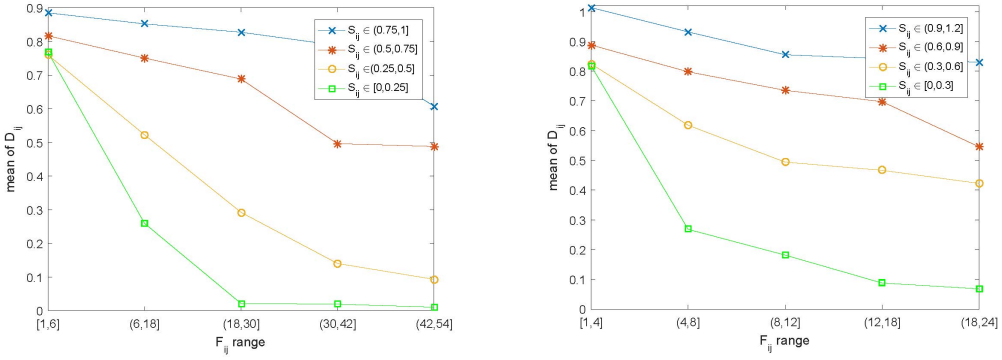
(1) $\frac{\partial f}{\partial x} < 0$: if u_i's friends issued similar ratings on item v_j, then the more number of them, the more likely u_i will agree with his friends' tastes. **(2)** $\frac{\partial^2 f}{\partial x^2} > 0$: the reason is that $\mathbf{F}_{ij}$ can go to infinity theoretically, and $f(x, y)$ is a decreasing function w.r.t. x with a lower bound of 0. Thus it is reasonable to assume $f(x, y)$ is a convex function w.r.t. x, which makes our model more resistant to noise. **(3)** $\frac{\partial f}{\partial y} > 0$: if u_i has a certain number of friends who rates v_j and they all think highly of it, then u_i is very likely to think highly of v_j as well, i.e. $\mathbf{D}_{ij}$ is small . But if ratings of these friends vary a lot, u_i's rating on v_j contains much more uncertainty, thus a higher $\mathbf{D}_{ij}$.

We check the assumption on two data sets Ciao[1] and Epinions[2] On each dataset, we collect all triples $(\mathbf{F}_{ij}, \mathbf{S}_{ij}, \mathbf{D}_{ij})$, and then categorize them into bins according to $\mathbf{F}_{ij}$ and $\mathbf{S}_{ij}$. For example, triples in Ciao that satisfy $\mathbf{F}_{ij} \in [1, 6]$ and $\mathbf{S}_{ij} \in [0, 0.25]$ fall into the same bin. Then we calculate the mean of $\mathbf{D}_{ij}$'s in each bin. The bins are selected to ensure that there is enough data to average out the noise. The results are shown in Figure 1(a) and Figure 1(b) respectively for the two data sets.

Since the data patterns shown by the two figures are similar, we just explain Figure 1(a) here. There are 4 lines in figure 1(a), each representing a certain level of agreement of a user's friends' tastes on an item. The line at the bottom means the friends has extremely similar tastes, i.e. $\mathbf{S}_{ij}$ is small, and the top line means their tastes are not so similar, i.e. $\mathbf{S}_{ij}$ is large. All four lines show a decreasing trend as $\mathbf{F}_{ij}$ increases, which satisfies the assumption of $\frac{\partial f}{\partial x} < 0$. From a vertical perspective, $\mathbf{D}_{ij}$ goes smaller as $\mathbf{S}_{ij}$ decreases on a certain range of $\mathbf{F}_{ij}$, satisfying the assumption of $\frac{\partial f}{\partial y} > 0$.

Modeling the f function. Neural networks are often used as a function approximation method because it has several merits, such as it is a universal approximator, easily constructed, and differentiable [1, 2, 7]. Recently, Dugas et al. [3] incorporated functional

[1] https://www.cse.msu.edu/ tangjili/trust.html
[2] https://www.trustlet.org/downloaded_epinions.html



(a) Ciao **(b) Epinions**

Figure 1: Mean of D_{ij} in each bin of Ciao and Epinions.

knowledge in neural networks, endowing neural networks with some derivative properties. In order to make the f function satisfy the derivative properties in our assumption, we use Dugas's method to model it:

$$f(x,y) = e^{w_0} + \sum_{i=1}^{H} e^{w_i}\zeta(b_i - e^{v_i}x)h(d_i + e^{z_i}y),$$

where $\zeta(s) = ln(1+e^s)$ and $h(s) = 1/(1+e^{-s})$. w_0, w_i, b_i, v_i, d_i and z_i are model parameters that need to be learned. H controls the model complexity of $f(x,y)$, and should be chosen to balance the trade-off [12] between bias and variance. A normal choice for H is 4. One can easily check that the value of f is necessarily positive, and it satisfies our assumptions of $\frac{\partial f}{\partial x} < 0$, $\frac{\partial^2 f}{\partial x^2} > 0$ and $\frac{\partial f}{\partial y} > 0$.

3.3 Model formulation

The merged rating matrix. Based on our assumption, we can use a user's friends' average rating $\mathbf{M}_{ij}$ to substitute his missing rating on a certain item when $f(\mathbf{F}_{ij}, \mathbf{S}_{ij})$ is relatively small. The resulting rating matrix containing both original observed ratings and pseudo ratings is referred to as the merged rating matrix $\widetilde{\mathbf{R}}$. Formally,

$$\widetilde{\mathbf{R}}_{ij} = \begin{cases} \mathbf{R}_{ij}, & \text{if } I^R_{ij} = 1, \\ \mathbf{M}_{ij}, & \text{if } I^R_{ij} = 0 \wedge \mathbf{F}_{ij} > 0 \wedge \\ & f(\mathbf{F}_{ij}, \mathbf{S}_{ij}) \le \theta_{ps}, \\ 0, & \text{otherwise,} \end{cases}$$

where θ_{ps} is a threshold to indicate $f(\mathbf{F}_{ij}, \mathbf{S}_{ij})$ is small enough.

The weights of ratings. Since the pseudo ratings are inferred data rather than ground truth data, they should not be valued as much as the original observed ratings. If pseudo ratings and observed rating are treated equally in matrix factorization, the large amount of noise in pseudo ratings will overwhelm the useful information brought by them, thus reduce the prediction accuracy. In addition, we have different confidence in pseudo ratings too. The pseudo ratings with a smaller value of $f(\mathbf{F}_{ij}, \mathbf{S}_{ij})$ should be trusted more than those with a larger $f(\mathbf{F}_{ij}, \mathbf{S}_{ij})$. Therefore, we assign different weights to observed ratings and pseudo ratings, representing our confidence in that rating. Formally,

$$\mathbf{W}_{ij} = \begin{cases} 1, & \text{if } I^R_{ij} = 1, \\ e^{-f(\mathbf{F}_{ij}, \mathbf{S}_{ij})}, & \text{if } I^R_{ij} = 0 \wedge \widetilde{\mathbf{R}}_{ij} > 0, \\ 0, & \text{otherwise,} \end{cases}$$

which makes each of the observed ratings having a weight of 1, while each of the pseudo ratings a weight of value in $(0, 1)$.

Matrix factorization. With the newly generated pseudo ratings and their corresponding weights, we can use optimization method to find the user-feature matrix $\mathbf{U}$ and the item-feature matrix $\mathbf{V}$ by solving the following optimization problem:

$$\mathbf{U}, \mathbf{V} = \underset{\mathbf{U},\mathbf{V}}{\text{argmin}} \{ \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{W}_{ij}(\widetilde{\mathbf{R}}_{ij} - \mathbf{U}_i^T \mathbf{V}_j)^2 + \lambda||\mathbf{U}||^2 + \lambda||\mathbf{V}||^2 \}.$$

Although our model has a very similar form with the Probabilistic Matrix Factorization Model [14], it has incorporated social information in pseudo ratings and the weight matrix $\mathbf{W}$, thus becomes much more powerful in prediction, as will be shown in experiments.

Addressing cold start problem. Since the ratings provided by cold start users are much less than that of normal users, in order to avoid overfitting on cold start users, we generate more pseudo ratings for cold start users by sacrificing the confidence on them. Namely, the threshold of $f(\mathbf{F}_{ij}, \mathbf{S}_{ij})$ for selecting pseudo ratings of cold start users is looser than that of normal users, i.e.,

$$\widetilde{\mathbf{R}}_{ij} = \begin{cases} \mathbf{R}_{ij}, & \text{if } I^R_{ij} = 1, \\ \mathbf{M}_{ij}, & \text{if } I^R_{ij} = 0 \wedge \mathbf{F}_{ij} > 0 \wedge \\ & (f(\mathbf{F}_{ij}, \mathbf{S}_{ij}) \le \theta_{ps} \vee \\ & (u_i \text{ is a cold start user} \wedge \\ & f(\mathbf{F}_{ij}, \mathbf{S}_{ij}) \le \alpha \cdot \theta_{ps})), \\ 0, & \text{otherwise,} \end{cases}$$

where α is a coefficient larger than 1. By doing so, more pseudo ratings with lower confidence level are fed to the cold start users. Although these ratings with low confidence bring more noise (bias) to the prediction of cold start users, the prediction variance is far more reduced.

Table 2: Statistics of two data sets.

dataset	Epinions	Ciao
users	40163	7375
items	139738	99746
ratings	664824	280391
rating density	0.051%	0.03%
trusters	33960	6792
trustees	49288	7297
trusts	487183	111781
trust density	0.029%	0.23%

Table 1: Performance comparison on all users and cold start users, where * indicates the best performance among previous methods, boldface indicates the best among all, and the column 'Improve' indicates the relative improvements of our approach against the * results.

		Previous work						Our methods			
All users	Metrics	PMF	SoRec	SoReg	SocialMF	TrustMF	TrustSVD	PsRec$_{w1}$	PsRec$_{no_cold}$	PsRec	Improve
Ciao	MAE	0.822	0.761	0.815	0.749	0.753	0.723*	0.684	0.680	**0.672**	7.05%
	RMSE	1.078	1.01	1.076	0.976	1.014	0.956*	0.913	0.906	**0.898**	6.07%
Epinions	MAE	0.909	0.884	0.932	0.826	0.819	0.805*	0.791	0.789	**0.785**	2.48%
	RMSE	1.197	1.142	1.232	1.082	1.095	1.044*	1.065	1.050	**1.043**	0.10%
Cold Start	Metrics	PMF	SoRec	SoReg	SocialMF	TrustMF	TrustSVD	PsRec$_{w1}$	PsRec$_{no_cold}$	PsRec	Improve
Ciao	MAE	0.926	0.73	0.949	0.741	0.77	0.721*	0.701	0.711	**0.698**	3.19%
	RMSE	1.191	1.031	0.214	0.978	1.096	0.962*	0.927	0.955	**0.915**	4.89%
Epinions	MAE	1.153	0.846*	1.139	0.857	0.853	0.868	0.820	0.843	**0.814**	3.78%
	RMSE	1.432	1.180	1.437	1.152	1.176	1.108*	1.081	1.107	**1.078**	2.71%

4 EXPERIMENTS

4.1 Experiment Setups

Data sets. Ciao and Epinions are two commonly used datasets in evaluating social recommender systems [6]. Both data sets contain user ratings for items and also social relations specified by the users. The ratings are integers from 1 to 5. The trust relation is represented as binary values of either 1 or 0, where 1 means trust and 0 means unobserved (we assume unobserved entries to be 0). Some data statistics are illustrated in Table 2.

Cross-validation. we employ 5-fold cross-validation for learning and testing. We randomly divide the rating data into five equal parts. In each time, four folds are used as training set and the remaining fold as the test set.

Evaluation metrics. Mean absolute error (MAE) and rooted mean square error (RMSE) are employed to evaluate prediction accuracy, defined by:

$$MAE = \frac{\sum_{i,j} |\widehat{\mathbf{R}}_{ij} - \mathbf{R}_{ij}|}{N}, RMSE = \sqrt{\frac{\sum_{i,j} (\widehat{\mathbf{R}}_{ij} - \mathbf{R}_{ij})^2}{N}}.$$

Testing views. Two data set views are created for testing. First, the *All* view indicates that all ratings are used as the test set. Second, the *Cold Start* view means that only the users who rate less than five items will be involved in the test set.

4.2 Comparison Methods

To comparatively evaluate the performance of our PsRec model, we select six representative methods from previous work as competitors, listed in Table 3 together with their optimal parameter settings. We also have 3 different variants of our own model:

(1)PsRec, which is exactly the model we described above and also our full-fledged one, involves different weights for pseudo ratings and observed ratings, and also different pseudo rating selection criteria to handle cold start problem. α is set to be 1.5 and θ_{ps} is selected such that $\widetilde{\mathbf{R}}$ has around 2 times nonzero entries as much as training data $\mathbf{R}$, denoted as $\theta_{ps} = \theta_{ps}\{2\}$. The values of θ_{ps} and α are selected through cross validation such that they give best performance.

(2)PsRec$_{w1}$, which is the same as PsRec, except that the weights of either pseudo ratings or observed ratings are all set to be 1. This variant is used to demonstrate the importance of the weighting strategy of our model.

(3)PsRec$_{no_cold}$, which is the same as PsRec, except that we use exactly the same threshold to select pseudo ratings for both cold start users and normal users, i.e. $\alpha = 1$. This variant is used to justify our technique to handle cold start problem.

Table 3: Comparison methods and parameter settings.

Methods	Optimal Parameters
PMF	$\lambda = 0.001$.
SoRec	$\lambda_c = 1.0$ for Epinions and 0.001 for Ciao.
SoReg	$\beta = 0.1$.
SocialMF	$\lambda_t = 1$.
TrustMF	$\lambda_t = 1$.
TrustSVD	$\lambda = 0.9$, $\lambda_t = 0.5$ for Epinions; $\lambda = 0.5$, $\lambda_t = 1$ for Ciao.
PsRec	$\alpha = 1.5$ and $\theta_{ps} = \theta_{ps}\{2\}$.
PsRec$_{w1}$	All parameters are the same as PsRec.
PsRec$_{no_cold}$	All parameters are the same as PsRec, except $\alpha = 1$.

4.3 Experiment Results

We now validate the performance of our proposed PsRec method by comparing the results on the two data sets with their competitors, which is presented in Table 1. The comparison is conducted on all users and cold start users respectively. For both all users and cold start users, our proposed PsRec method performs the best of all in terms of both MAE and RMSE on the two data sets. This demonstrates that our method has very good ability to handle sparsity and cold start problems in social recommender systems.

PsRec and PsRec$_{no_cold}$. From the lower part of Table 1, we can see that our method PsRec has a better performance on cold start users than all previous methods. This indicates that generating more pseudo ratings on cold start users by sacrificing a little bit confidence is essential to its success on cold start users.

PsRec and PsRec$_{w1}$. One can also see that PsRec has a better performance than that of PsRec$_{w1}$, which implies that only doing factorization with pseudo ratings is not enough to build a robust social recommender system. By putting proper weights on the pseudo ratings and observed ratings, we can arrive at a balanced point between the noise and informativeness, showing the effectiveness of our weighting strategy.

5 FUTURE WORK

There are several directions worth investigating in future. First, there may exist other better methods which bear more interpretability than a neural network to model the f function. Second, machine learning method could be employed to learn the hyperparameter θ_{ps} and α so as to eliminate parameter tuning.

REFERENCES

[1] A.R. Barron. 1993. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory* 39(3) (1993), 930–945.

[2] G. Cybenko. 1988. Continuous valued neural networks with two hidden layers are sufficient. *Technical report, Department of Computer Science, Tufts University, Medford, MA* (1988).

[3] C. Dugas, Y. Bengio, and Francois Belisle. 2009. Incorporating functional knowledge in neural networks. *Journal of Machine Learning Research* 10 (2009), 1239–1262.

[4] N. Golbandi, Y. Koren, and R. Lempel. 2010. On bootstrapping recommender systems. *Proceedings of the 19th ACM international conference on Information and knowledge management* (2010), 1805–1808.

[5] N. Golbandi, Y. Koren, and R. Lempel. 2011. Adaptive bootstrapping of recommender systems using decision trees. *Proceedings of the fourth ACM international conference on Web search and data mining* (2011), 595–604.

[6] G. Guo, J. Zhang, and N. Yorke-Smith. 2015. TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings. *AAAI* (2015).

[7] K. Hornik, M. Stinchcombe, and H. White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2 (1989), 359–366.

[8] M. Jamali and M. Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys)* (2010), 135–142.

[9] H. Ma, H. Yang, M. Lyu, and I. King. 2008. SoRec: social recommendation using probabilistic matrix factorization. *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR)* (2008), 931–940.

[10] H. Ma, D. Zhou, C. Liu, M. Lyu, and I. King. 2011. Recommender systems with social regularization. *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM)* (2011), 287–296.

[11] Prem Melville, Raymond J. Mooney, and Ramadass Nagarajan. 2002. Content-Boosted Collaborative Filtering for Improved Recommendations. *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-2002)* (2002), 187–192.

[12] J. Moody. 1994. *Prediction Risk and Architecture Selection for Neural Networks*. Springer.

[13] A.M. Rashid. 2002. Getting to know you: learning new user preferences in recommender systems. *Proceedings of the 7th international conference on Intelligent user interfaces. ACM.* (2002).

[14] R. Salakhutdinov and A. Mnih. 2008. Probabilistic Matrix Factorization. *Advances in Neural Information Processing Systems* 20 (2008), 1257–1264.

[15] E. Shmueli, A. Kagian, Y. Koren, and R. Lempel. 2012. Care to comment?: recommendations for commenting on news stories. *Proceedings of the 21st international conference on World Wide Web* (2012), 429–438.

[16] J. Tang, X. Hu, H. Gao, and H. Liu. 2010. Exploiting local and global social context for recommendation. *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys)* (2010), 135–142.

[17] S. Wasserman. 1994. Social Network Analysis. *Cambridge University Press* (1994).

[18] B. Yang, Y. Lei, D. Liu, , and J Liu. 2013. Social collaborative filtering by trust. *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)* (2013), 2747–2754.