2017 10th International Symposium on Computational Intelligence and Design

Similarity based Matrix Factorization for Recommender Systems

Gen Zhang[1], Xu Zhou[2]
[1,2]*College of Computer*
National University of Defense Technology
Hunan, China, 410073
[1]*zhanggen12@hotmail.com*, [2]*zhouxu@nudt.edu.cn*

***Abstract*—In the area of computational intelligence, recommender systems play important roles in both commercial and social activities. Among its implement methods, matrix factorization is one of the most popular approach with various merits. However, traditional matrix factorization methods either roughly aggregate it into final task or ignore users/items similarity, which may easily introduce the cause of over-fitting or under-fitting. To tackle this problem, in this paper, we propose a similarity based matrix factorization for recommender systems by considering both global and local information within and between users and items. Specifically, it applies a novel method to extract the latent information within users and items, and captures the local similarity by a low-dimension manifold. Then, global and local information are integrated through a manifold regularized matrix factorization. Finally, the reconstructed method can be used in recommender systems. Experiments on two benchmark data sets illustrate the outstanding performance of our method in terms of prediction accuracy.**

***Keywords*-Matrix Factorization; Manifold Regularization; Recommender Systems; Similarity**

I. INTRODUCTION

In big data era, as a powerful tool in commercial and social activities, recommender systems (RS) are attracting intensive researchers' attention. Comparing with other RS implement methods, matrix factorization (MF) presents its outstanding characteristics, with its high predictive accuracy and flexibility for modeling various real-life situations[1]. And MF is also adoptable to many data-intensive scenarios because the result of MF is in a compressed version of the original data matrix[2].

Recently a bunch of MF techniques are proposed, including Probabilistic MF[3], Regression-based MF[4], Time-based MF[5]. Although the aforementioned MF methods achieved state-of-art performance and are widely applied, they do not take the low-level information into account, such as user-user similarity. The value of aggregating low-level similarity such as social relations into recommending has been widely discussed in many previous works. In these methods[6–8], and they suggest that the low-level information is very helpful in optimizing the traditional methods. To solve these problems, several effort has been done recently. Typical work include[9] and[10]. Zheng et al.[9] integrated the user's pairwise relationship as well as locations and activities similarities into MF algorithm. However, they took the activities similarity as the number of activities co-search in search engines, which is not robust and cannot be generalized to other scenario. Despite their advantages in each aspect, the above methods own some common weaknesses clearly: (1) most of them ignored the low-level similarity; (2) few MF methods considering such similarity simply used ad-hoc approach to reveal this information and ignored objects intrinsic relationship; (3) they roughly integrated low-level similarity into matrix factorization, which does not have semantic meaning.

In this paper, we propose a similarity based matrix factorization (Similarity MF) to fill the gaps mentioned above. The basic assumption of Similarity MF is that low-level information is embedded in high-dimensional preference matrix extended space. Thus, through a manifold learning technique, well captured low-level information can be appropriately aggregated into reconstructed interest matrix. Our work implements this idea in the following two steps:

(1) It first introduces a generalized and universal similarity measurement method to capture the internal and external relationship between users and items.

(2) Then, it feeds the captured low-level similarity into MF by a manifold regularization. This novel MF process is integrated into a recommender system, and the reconstructed rating matrix is used to make recommending.

The contributions of our work can be collected as follows:

• The internal and external similarity is considered and our method combines this two together to get the integrated similarity. Thus the relationship between each user and between each item is precisely captured based on their attributes. This significantly leverages the relevant information, which is ignored by previous work at most of the time.

• A novel objective function has been presented as well as its effective and efficient optimization process. Thus our model exceeds most existing MF models for its robustness and accuracy.

In order to examine the accuracy of our novel method, experiments in two benchmark data sets demonstrate the outstanding performance of our method in terms of prediction accuracy.

DOI 10.1109/ISCID.2017.22

$\mathcal{U}$	User set	u	User u
$\mathcal{I}$	Item set	i	Item i
$\mathcal{A}$	Attribute set	a	Attribute a
V_a	Value set	RT	Ratings
P	User matrix	Q	Item matrix
K	Number of factors	R	Similarity
ES	EOS function	NE	Neighbor
L	Objective function	α	Learning rate
N	Number of users	M	Number of items
L	Number of attributes	$\hat{RT}$	Predicted ratings
H	Number of neighbors	IS	IOS function
S	Manifold relationship	$STEP$	Iteration times

Table I: Notations in our model

II. Preliminaries

In this section, we formalize the problem in recommender systems to construct a structure of future work.

$\mathcal{U} = \{u_1, u_2, ..., u_n\}$ and $\mathcal{I} = \{i_1, i_2, ..., i_m\}$ represent the set of users and the set of items respectively. N is the number of users and M is the number of items. $\mathcal{A} = \{a_1, a_2, ..., a_l\}$ is the set of attributes and L is the number of attributes. $V_a = \{v_a^{u_1}, v_a^{u_2}, ..., v_a^{u_m}\}$ or $V_a = \{v_a^{i_1}, v_a^{i_2}, ..., v_a^{i_n}\}$ is the value set of an attribute a for users or items.

RT is the rating matrix for the users and items in recommender systems. $RT(u, i)$ represents user u's rating on item i.

In classic matrix factorization method in recommender systems, predicted rating $\hat{RT}(u, i)$ of user u for item i are calculated as below:

$$\hat{RT}(u, i) = \vec{P}_u \cdot \vec{Q}_i \quad (1)$$

where $\vec{P}_u$ is a $1\times K$ vector for user u and $\vec{Q}_u$ is a $K\times$ 1 vector for item i. K is the number of latent factors of user matrix P and item matrix Q.

Finally, the recommendation problem can be described as $Input = \{\mathcal{U}, \mathcal{I}, \mathcal{A}, V, RT\}$, which consists of users ,items, attributes of users and items and users' ratings on items. And the objective is to calculate $\hat{RT}$ to predict the missing values in recommendation. Table. I illustrates the notations of the problem.

III. The Similarity based Matrix Factorization Model

In this section, we propose a model to implement our improved matrix factorization in recommender systems, with a manifold regularization taken into account. In addition, we also introduce a novel method to extract the latent similarity between each user and each item based on their prior information. We first consider a generalized and universal similarity calculating approach on users and items with their interaction. Then we show how to embed prior information into a low-dimension manifold to capture the local geometric information for both users and items. The reconstructed matrix can be used in recommender systems.

A. A Generalized and Universal Similarity Calculating Approach

This section introduces a new similarity calculating approach based on the known character and attribute of objects, namely GUSCA, which reveal the explicit and implicit factors to the process of other machine learning and analyzing methods such as matrix factorization.

Typically, GURCA works as follows:

Let us denote a n-dimensional space Ω with Λ objects $\{o_1, o_2, ..., o_\Lambda\}$. The similarity between o_ι and o_τ, which is $S(o_\iota, o_\tau)$, is defined as follows:

$$S(o_\iota, o_\tau) = \sum_{a\in\mathcal{A}, o_\iota, o_\tau\in\Omega} F(IS_a(o_\iota, o_\tau), ES_a(o_\iota, o_\tau)) \quad (2)$$

where $\mathcal{A}$ is an attribute space with all objects features involved in it. And $IS_a(o_\iota, o_\tau)$ and $ES_a(o_\iota, o_\tau)$ are functions for IOS (internal object similarity) and EOS (external object similarity) respectively. While $F(x, y)$ is a function combining both IOS and EOS, corresponding to the similarity between two objects in one attribute. According to this GUSCA expression, we can simply recognize the assumption that the similarity between two objects are a combination of two kinds of inferior similarity, internal and external similarity.

B. Modeling the Internal and External Similarity

In this section, we will introduce a novel way to extract explicit and implicit information between objects. And we discuss similarity between objects in two aspects: internal and external. Then we will combine the two separate parts together to form the superior similarity.

With regard to the internal similarity, we only focus on the impact of different values of one attribute on two objects. First, we will introduce a frequency counter function $C(v_a^{o_\iota})$ of a certain value in attribute a of object o_ι, which indicates how many times value $v_a^{o_\iota}$ has appeared in all objects. The internal similarity function of **IOS** is defined as:

$$IS_a(o_\iota, o_\tau) = \frac{\frac{C(v_a^{o_\iota})}{\Lambda} \cdot \frac{C(v_a^{o_\tau})}{\Lambda}}{\frac{C(v_a^{o_\iota})}{\Lambda} + \frac{C(v_a^{o_\iota})}{\Lambda} \cdot \frac{C(v_a^{o_\tau})}{\Lambda} + \frac{C(v_a^{o_\tau})}{\Lambda}} \quad (3)$$

where Λ is the number of all objects in the whole space Ω. As is illustrated in this function, $IS_a(o_\iota, o_\tau) \in [0, \frac{1}{5}]$ and the internal similarity is $\frac{1}{5}$ if the two objects have the same value $\frac{\Lambda}{2}$. And this method specifically captures the internal similarity based on the frequency of values.

The external similarity between objects (o_ι, o_τ) of a certain attribute a is calculated by aggregating the impact of other attributes $a^{'} \in \mathcal{A}(a^{'} \neq a)$. We first define a conditional probability function:

$$CP_{V_{a'}|v_a^{o_\iota}}(V_{a'}, v_a^{o_\iota}) = \sum_{v_{a'}^{o_\kappa}\in V_{a'}, o_\iota, o_k\in\Omega} \frac{C(v_{a'}^{o_\kappa} \cap v_a^{o_\iota})}{C(v_a^{o_\iota})} \quad (4)$$

where $V_{a'}$ is a set of values of attribute $a^{'}(a^{'} \neq a)$.

The external similarity function of **EOS** between two objects (o_ι, o_τ) is defined as follows:

$$ES_a(o_\iota, o_\tau) = \sum_{a^{'}, a \in \mathcal{A}, a^{'} \neq a} \gamma min(CP_{V_{a^{'}}|v_a^{o_\iota}}(V_{a^{'}} v_a^{o_\iota}), \quad CP_{V_{a^{'}}|v_a^{o_\tau}}(V_{a^{'}}, v_a^{o_\tau})) \tag{5}$$

where $a^{'}$ is in attribute set $\mathcal{A}$ and not equal to attribute a. And γ is the weight factor of each part where $\gamma \in (0, 1)$ and we set $\gamma = \frac{1}{\Lambda}$ in our model specifically . As displayed above, $ES_a(o_\iota, o_\tau) \in [0, 1]$.

Finally, the similarity between (o_ι, o_τ) is defined as below:

$$R(o_\iota, o_\tau) = \sum_{a \in \mathcal{A}, o_\iota, o_\tau \in \Omega} \sigma IS_a(o_\iota, o_\tau) + (1-\sigma) ES_a(o_\iota, o_\tau) \tag{6}$$

where σ is the weight factor between IOS and EOS which measures the affect of internal and external similarity to objects and we set $\sigma = 0.5$ in this model typically.

C. Internal and External Similarity in Recommender Systems

There is ubiquitous and intrinsic relationship between objects in recommender systems, namely users and items. Taking Taobao as an example, users are associated to each other based on their age, gender and occupation. And items are interacted according to category, prices and manufactures. The above similarity model can be applied into recommender systems for the reason that users and items are objects in the space of $\mathcal{RS}$ where $R(u_\iota, u_\tau)$ and $R(i_\iota, i_\tau)$ can be calculated by applying the above similarity function between users and items.

In recommender systems, we define the similarity between users $R(u_\iota, u_\tau)$ and items $R(i_\iota, i_\tau)$ as follows:

$$R(u_\iota, u_\tau) = \sum_{a \in \mathcal{A}_u, u_\iota, u_\tau \in \mathcal{U}} \sigma IS_a(u_\iota, u_\tau) + (1-\sigma) ES_a(u_\iota, u_\tau) \tag{7}$$

$$R(i_\iota, i_\tau) = \sum_{a \in \mathcal{A}_i, i_\iota, i_\tau \in \mathcal{I}} \sigma IS_a(i_\iota, i_\tau) + (1-\sigma) ES_a(i_\iota, i_\tau) \tag{8}$$

where $\mathcal{A}_u$ and $\mathcal{A}_i$ are the sets of user and item attributes respectively. And $\mathcal{U}$ and $\mathcal{I}$ are the sets of users and set of items respectively. $IR_a(u_\iota, u_\tau)$, $ER_a(u_\iota, u_\tau)$, $IR_a(i_\iota, i_\tau)$ and $ER_a(i_\iota, i_\tau)$ can be determined by applying Eq. 3 and Eq. 5.

D. Detail of Similarity based Matrix Factorization

In this section, we further introduce the proposed similarity between users and items and a manifold regularization into the classic matrix factorization model. This similarity based matrix factorization method aggregates internal and external similarity and data distribution geometry to capture the global and local information of both users and items.

A basic assumption of manifold regularization is that user and items, which are close in the manifold, are more presumably to have the same features. We define $NE(o_\iota)$ to measure the distance between objects in the manifold based on the similarity function in Eq. 6 as follows:

$$NE(o_\iota) = \{o_\tau | R(o_\iota, o_\tau) > R(o_\iota, o_x), \quad \forall o_x \notin NE(o_\iota), \forall o_\tau \in NE(o_\iota)\} \tag{9}$$

More specifically, each user $u \in \mathcal{U}$ has the same features with its H-nearest neighbors $NE(u)$ and each item $i \in \mathcal{I}$ has the same features with its H-nearest neighbors $NE(i)$ by applying the neighbor function in Eq. 9. After that, we define:

$$S(o_\iota, o_\tau) = \begin{cases} R(o_\iota, o_\tau) & o_\iota \in NE(o_\tau) \ or \ o_\tau \in NE(o_\iota) \\ 0 & otherwise \end{cases} \tag{10}$$

Finally, we define the objective function corresponding to the similarity based matrix factorization as follows:

$$\begin{aligned} L = & \sum_{u \in \mathcal{U}, i \in \mathcal{I}} (RT(u,i) - \hat{RT}(u,i))^2 \\ & + \lambda_1 \sum_{u, u^{'} \in \mathcal{U}} S(u, u^{'}) \parallel \vec{P}_u - \vec{P}_{u^{'}} \parallel^2 \\ & + \lambda_2 \sum_{i, i^{'} \in \mathcal{I}} S(i, i^{'}) \parallel \vec{Q}_i - \vec{Q}_{i^{'}} \parallel^2 \\ & + \beta_1 \parallel \vec{P}_u \parallel^2 + \beta_2 \parallel \vec{Q}_i \parallel^2 \end{aligned} \tag{11}$$

where the $\sum_{u, u^{'} \in \mathcal{U}} S(u, u^{'}) \parallel \vec{P}_u - \vec{P}_{u^{'}} \parallel^2$ and $\sum_{i, i^{'} \in \mathcal{I}} S(i, i^{'}) \parallel \vec{Q}_i - \vec{Q}_{i^{'}} \parallel^2$ are the manifold regularization discussed above.

Accordingly, the objective function Eq. 11 is optimized by minimizing the L with gradient decent method for $\vec{P}_u$ and $\vec{Q}_i$ respectively:

$$\begin{cases} \dfrac{\partial L}{\partial \vec{P}_u} = & -2(RT(u,i) - \hat{RT}(u,i))\vec{Q}_i \\ & + \lambda_1 \displaystyle\sum_{u, u^{'} \in \mathcal{U}} S(u, u^{'})(\vec{P}_u - \vec{P}_{u^{'}}) + \beta_1 \vec{P}_u \\ \dfrac{\partial L}{\partial \vec{Q}_i} = & -2(RT(u,i) - \hat{RT}(u,i))\vec{P}_u \\ & + \lambda_2 \displaystyle\sum_{i, i^{'} \in \mathcal{I}} S(i, i^{'})(\vec{Q}_i - \vec{Q}_{i^{'}}) + \beta_2 \vec{Q}_u \end{cases} \tag{12}$$

Further, $\vec{P}_u$ and $\vec{Q}_i$ are updated through iteration as below:

$$\begin{cases} \vec{P}_u^{'} = \vec{P}_u - \alpha \dfrac{\partial L}{\partial \vec{P}_u} \\ \vec{Q}_i^{'} = \vec{Q}_i - \alpha \dfrac{\partial L}{\partial \vec{Q}_i} \end{cases} \tag{13}$$

where α is the learning rate of the objective function. Finally, the $N \times K$ matrix P and the $K \times M$ matrix Q can be calculated through the above method.

Algorithm 1 Our proposed Similarity based MF

Input: $\mathcal{A}_u$: user attribute in user space U
$\mathcal{A}_i$: item attribute in item space I
RT: known ratings in $\mathcal{RS}$
$STEP$: the iteration times
α : the learning rate
β_1: the weight of normal user regularization
β_2: the weight of normal item regularization
λ_1: the weight of user manifold regularization
λ_2: the weight of item manifold regularization
K: the number of latent factors
ϵ : the threshold to suspend the iteration
Output: The predicted ratings $\hat{RT}$

Construct the relationship matrix R for both users $u \in \mathcal{U}$ and items $i \in \mathcal{I}$ with Eq. 7 and Eq. 8
Construct the neighbor matrix N for both users $u \in \mathcal{U}$ and items $i \in \mathcal{I}$ with Eq. 9
Calculate the matrix S for both users $u \in \mathcal{U}$ and items $i \in \mathcal{I}$ with Eq.10 with regard to a manifold regularization
Initial P and Q with random values
while $step < STEP or L_{step} - L_{step-1} < \epsilon$ **do**
    for $u \in [1, N]$ **do**
        for $i \in [1, M]$ **do**
            if $R(u, i) > 0$ **then**
                $\vec{P}_i' = \vec{P}_i - \alpha \frac{\partial L}{\partial \vec{P}_i}$
                $\vec{Q}_u' = \vec{Q}_u - \alpha \frac{\partial L}{\partial \vec{Q}_u}$
            end if
        end for
    end for
end while
$\hat{RT} = P \cdot Q$
return $\hat{RT}$

IV. Experiments

In this section, we evaluate our similarity based matrix factorization by comparing it with other recommendation algorithms on two data sets in terms of RMSE and MAE and thoroughly analyze the experiment results.

A. Data Sets

Few data sets with users and items attributes are available to evaluation our method. We conduct the experiment on the MovieLens/GroupLens and the BookCrossing data sets. Both of the data sets have been widely applied in most of existing recommender systems research.

B. Evaluation Methodology

We select several different methods to perform as the baselines to compare with: classic MF, MF with Hamming similarity and MF with Jacard similarity and MF with our similarity measurement GUSCA but without manifold regularization.

Five repeat experiments on each of the data sets are conducted on similarity MF and the four baselines respectively. And we use mean absolute error (MAE) and root mean square error (RMSE) as the evaluation metrics shown in Eq. 14. The smaller MAE and RMSE is, the more accurate the prediction is. The result is demonstrated as: $ave \pm std$, where ave is the average of five experiments and std is the standard deviation.

$$\begin{cases} MAE = \frac{\sum_{u,i \in TEST} |RT(u,i) - \hat{RT}(u,i)|}{|TEST|} \\ RMSE = \sqrt{\frac{\sum_{u,i \in TEST} (RT(u,i) - \hat{RT}(u,i))^2}{|TEST|}} \end{cases} \tag{14}$$

C. Parameters Setting

We also examine the impact of parameters $\{\lambda_1, \lambda_2, STEP, K, \alpha, \beta_1, \beta_2\}$ in our method. λ_1 and λ_2 control the contribution of the manifold regularization of users and items. $STEP$ and α control the rate of learning. β controls the contribution of the regularization of P, Q. In this experiment, we set the model parameters as: $\lambda_1 = 0.01$, $\lambda_2 = 0.01$, $STEP = 150$, $K = 20$, $\alpha = 0.02$, $\beta_1 = \beta_2 = 0.1$.

D. Method Performance

As for MovieLens data set, we respectively evaluate our method with the four baselines, in terms of MAE and RMSE. As is illustrated in Table. II, similarity MF outperforms the classic MF by 9.52% on MAE, 9.34% on RMSE. Compared with the Hamming MF and Jacard MF, similarity MF improves by 6.17% on MAE, 6.73% on RMSE and 3.80% on MAE, 3.00% on RMSE. And the comparison between similarity MF and GUSCA MF illustrates 2.56% on MAE, 2.02% on RMSE improvement.

With regard to the evaluation on BookCrossing data set shown in Table. III, the similarity MF reaches 3.96% on MAE, 5.95% on RMSE respectively over the classic MF. The similarity MF beats the Hamming and Jacard MF by 2.67% on MAE, 4.05% on RMSE and 1.80% on MAE, 3.27% on RMSE. The similarity MF is 0.91% on MAE, 1.66% on RMSE over the GUSCA MF.

In summary, based on the experiment results, our method definitely beats the four baselines in terms of MAE and RMSE. The classic MF ignores the relationship between users and items and is not capable of revealing the ground truth. And our method exceeds the Hamming and Jacard similarity measurement by applying a GUSCA to measure

	Classic MF	Hamming MF	Jacard MF	GURCA MF	Similarity MF
MAE	0.84±0.02	0.81±0.01	0.79±0.02	0.78±0.02	**0.76±0.02**
Improve	9.52%	6.17%	3.80%	2.56%	-
RMSE	1.07±0.03	1.04±0.03	1.00±0.03	0.99±0.03	**0.97±0.01**
Improve	9.34%	6.73%	3.00%	2.02%	-

Table II: Performance of our method and the baselines on MovieLens

	Classic MF	Hamming MF	Jacard MF	GURCA MF	Similarity MF
MAE	4.55±0.01	4.49±0.03	4.45±0.01	4.41±0.02	**4.37±0.01**
Improve	3.96%	2.67%	1.80%	0.91%	-
RMSE	5.04±0.04	4.94±0.01	4.90±0.02	4.82±0.01	**4.74±0.03**
Improve	5.95%	4.05%	3.27%	1.66%	-

Table III: Performance of our method and the baselines on BookCrossing

the similarity between users and items more precisely. Further, the manifold regularization can lead the gradient descent to the ideal direction because manifold method only assumes that users and items, which are close in the manifold, are more presumably to have the same features. By applying a manifold, our method is capable of aggregating the low-level information to a higher aspect.

V. Conclusion

This work introduces a generalized and universal similarity calculating approach to measure the similarity between users and items in recommender systems, and proposes a manifold method to aggregate the local, global information as well as the intrinsic relationships of data. Then, it designs a novel objective function with manifold regularization for matrix factorization and provides a high effective optimization method. As the experiment results demonstration, our similarity based matrix factorization method outperforms other existing MF models in recommender systems in both MAE and RMSE evaluation metrics.

Acknowledgement

This work is partially supported by the The National Key Research and Development Program of China (2016YFB0200401), by program for New Century Excellent Talents in University, by National Science Foundation (NSF) China 61402492, 61402486, 61379146, by the laboratory pre-research fund (9140C810106150C81001).

References

[1] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[2] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, pp. 556–562, 2001.

[3] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *Proceedings of the 25th international conference on Machine learning*, pp. 880–887, ACM, 2008.

[4] D. Agarwal and B.-C. Chen, "Regression-based latent factor models," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 19–28, ACM, 2009.

[5] Y. Koren, "Collaborative filtering with temporal dynamics," *Communications of the ACM*, vol. 53, no. 4, pp. 89–97, 2010.

[6] H. Ma, "An experimental study on implicit social recommendation," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pp. 73–82, ACM, 2013.

[7] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with explicit and implicit social relations," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, p. 29, 2011.

[8] X. Yang, H. Steck, and Y. Liu, "Circle-based recommendation in online social networks," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1267–1275, ACM, 2012.

[9] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Towards mobile intelligence: Learning from gps history data for collaborative recommendation," *Artificial Intelligence*, vol. 184, pp. 17–37, 2012.

[10] R. M. Bell and Y. Koren, "Scalable collaborative filtering with jointly derived neighborhood interpolation weights," in *IEEE International Conference on Data Mining*, pp. 43–52, IEEE, 2007.