Local Ensemble across Multiple Sources for Collaborative Filtering

Jing Zheng
Beijing Key Lab of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications
Beijing, China 100876
ollichooo@gmail.com

Fuzhen Zhuang*
Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS
Beijing, China 100190
zhuangfz@ics.ict.ac.cn

Chuan Shi
Beijing Key Lab of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications
Beijing, China 100876
shichuan@bupt.edu.cn

ABSTRACT

Recently, Transfer Collaborative Filtering (TCF) methods across multiple source domains, which employ knowledge from different source domains to improve the recommendation performance in the target domain, have been applied in recommender systems. The existing multi-source TCF methods either require overlapping objects in different domains or simply re-weight domains to merge them together. In this paper, we propose a novel *LO*cal *EN*semble framework across multiple source domains for collaborative filtering (called LOEN for short), where weights of multiple sources for each missing rating in the target domain are determined according to their corresponding local structures. Compared with the previous TCF methods, LOEN does not require overlapping data and considers the divergence of sources through exploiting the local structures of ratings, which allows LOEN to be more general and effective. Experiments conducted on real datasets validate the effectiveness of LOEN, especially for knowledge transfer across unrelated source domains.

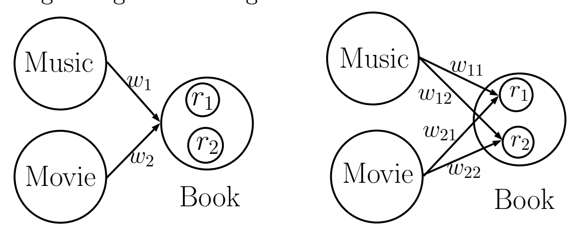
KEYWORDS

Transfer collaborative filtering, recommender system, local ensemble

1 INTRODUCTION

Recommender system has attracted amounts of interest [3, 10]. Recently, Transfer Collaborative Filtering (TCF), which transfers knowledge from source domains to the target domain for collaborative filtering, has been applied to alleviate the data sparsity problem in recommender systems [1, 5, 8].

Most previous TCF methods only focus on transferring knowledge from one single source domain to one target domain [5], which might not fully make use of the abundant information from multiple sources. Recently, some efforts have been made to learn knowledge from multiple sources to improve the recommendation performance [6, 7]. In general, there are two strategies to exploit the knowledge from multiple sources. The first one is to joint multiple sources according to the overlapping users or items [6]. This strategy may limit the applicability of the proposed method. Data of overlapping users or items in different applications are few and hard to be collected in real world. The second one is to re-weight different sources to boost the recommendation performance [7]. For example, considering the relationship of source domains and the target domain, TALMUD [7] assigns weights to different sources in a global manner. For each source domain, TALMUD assigns the same weight for all missing ratings in the target domain while ignoring the difference of their local structures. Indeed, the same source domain may contribute totally different affect to different missing ratings in the target domain.



(a) TALMUD (b) LOEN

Figure 1: Difference of TALMED and LOEN

In order to address the limitations of the existing multi-source TCF methods, we propose a novel *LO*cal *EN*semble method across multiple source domains for collaborative filtering (called LOEN). LOEN focuses on each missing rating and re-weights the source domains according to their local structures. Figure 1 illustrates the difference of LOEN and TALMUD. When transferring knowledge from source domains of Music and Movie to the target domain Book, TALMUD simply assigns weights to these two source domains to integrate knowledge together, while LOEN re-weights these two domains for each missing rating. LOEN takes measures to obtain the top-k similar observed ratings of each missing ratings to capture its local structure first. Then, LOEN

*Fuzhen Zhuang is the corresponding author and he is also with University of Chinese Academy of Sciences, Beijing 100190, China.

CIKM'17, , November 6–10, 2017, Singapore.

ACM ISBN ISBN 978-1-4503-4918-5/17/11...$15.00
https://doi.org/http://dx.doi.org/10.1145/XXXXXX.XXXXXX

assigns different weights to domains for each missing rating according to the similarity of their local structures. Experiments on two real datasets illustrate the superiority of LOEN compared with the-state-of-art methods. Particularly, LOEN shows its superiority in utilizing unrelated sources and robustness for more domains.

2 PRELIMINARY

2.1 Problem Definition

In this section, we introduce the notations and definitions used in this paper. Suppose there are n rating matrices from source domains (denoted as $R^s = \{R^{(1)}, R^{(2)}, ..., R^{(n)}\}$) and a rating matrix R^t from target domain. R^t includes two parts: the observed ratings R^o and the missing ratings $R^p = \{R_1^p, R_2^p ..., R_c^p\}^\top$ (p denotes the value to be predicted in R^t, and c is the number of R^p). That is, $R^t = R^o \cup R^p$. Our goal is to obtain different prediction results from multiple sources and integrate them for the prediction of R^p.

2.2 Transfer Knowledge From Single Domain

We firstly introduce the basic TCF framework CBT [5], which transfers knowledge from one domain to the target domain.

Given a target domain with a rating matrix R^t and a source domain with a rating matrix $R^{(1)}$. CBT firstly learns the shared cluster-level rating pattern from $R^{(1)}$ as follows,

$$\begin{aligned} &\min_{U^{(1)}\geq 0, V^{(1)}\geq 0, S\geq 0} \left\| R^{(1)} - U^{(1)} S V^{(1)\top} \right\|^2 \\ &s.t.\ U^{(1)\top} U^{(1)} = I, V^{(1)\top} V^{(1)} = I, \end{aligned} \tag{1}$$

where $U^{(1)}$, $V^{(1)}$ are the indictor matrices of users and items, respectively. Each row of $U^{(1)}$ (or $V^{(1)}$) indicates the cluster for this user (or item) so that each row should have only one nonnegative entry. $\|\cdot\|$ denotes the Frobenius norm. Next, we can obtain the rating pattern B of $R^{(1)}$ as follows,

$$B = [U^{(1)\top} R^{(1)} V^{(1)}] \oslash [U^{(1)\top} \mathbf{1}\mathbf{1}^\top V^{(1)}], \tag{2}$$

where $\oslash$ denotes element-wise division and $\mathbf{1}$ denotes all-one matrix. Then, CBT transfers B to the target domain, and the optimization process is as follows,

$$\begin{aligned} &\min_{U_t\geq 0, V_t\geq 0} \left\| R^t - U_t B V_t^\top \right\|^2 \\ &s.t.\ U_t^\top U_t = I, V_t^\top V_t = I, \end{aligned} \tag{3}$$

where U_t, V_t are the indicator matrices of users and items in the target domain. The rating pattern construction and transferring process are detailed in [5]. Thus, we can construct the prediction matrix $X^{(1)}$ as follows,

$$X^{(1)} = U_t B V_t. \tag{4}$$

2.3 Transfer Learning for Multiple Source Domains

In this section, we introduce the TALMUD, which is one of the most representative multi-source TCF methods. TALMUD employs CBT in different source domains to obtain multiple prediction results for the target domain and merge the results together by re-weighting them as follows,

$$\min_{U_t\geq 0, V_t\geq 0, B\geq 0} \left\| R^t - \sum_{i=1}^{n} \alpha_i U_t^{(i)} B^{(i)} V_t^{(i)\top} \right\|^2, \tag{5}$$

where $B^{(i)}$ is the rating pattern of $R^{(i)}$ and α_i is the weight for the prediction results generated from the ith domain. $U_t^{(i)}$, $V_t^{(i)}$ are the indictor matrices of users and items for R^t with $B^{(i)}$, respectively.

TALMUD simply re-weights the prediction results from different sources on the basis that the data distribution in various sources are independent. However, the data in real world are always non-i.i.d. In order to avoid the problems discussed above, we proposed the LOEN method.

3 LOCAL ENSEMBLE ACROSS MULTIPLE SOURCES

In this section, we introduce the LOEN, which utilizes the local structure of ratings to improve the recommendation performance. We first calculate the top-k similarity ratings of each missing rating. Then, we employ CBT to transfer knowledge from multiple sources for prediction in the target domain. Finally, we assign weights to each predicted rating according to its distance from the corresponding missing rating in the target domain, which can be obtained indirectly by measuring the similarity of their local structures.

3.1 Rating Similarity

In this section, we introduce the method to measure the similarity of R^p and R^o in the target domain.

First, we get the latent feature vectors of users P and items Q in the target domain by employing the low-rank matrix factorization method [9]. Next, we can calculate the similarity of users S^P and items S^Q via cosine similarity,

$$\begin{aligned} S^P(P_i, P_j) &= \frac{P_i \cdot P_j^\top}{|P_i| \times |P_j|}, \\ S^Q(Q_i, Q_j) &= \frac{Q_i \cdot Q_j^\top}{|Q_i| \times |Q_j|}. \end{aligned} \tag{6}$$

Then, we can measure the rating similarity S^{PQ} between two ratings (R_{ij} and R_{ab}) as follows,

$$S^{PQ}(R_{ij}, R_{ab}) = S^P(P_i, P_a) \cdot S^Q(Q_j, Q_b). \tag{7}$$

For each missing rating, we record the index of its top-k similar observed ratings (denoted as $I_i \in \mathbb{R}^{1\times k}(i = 1, 2, ..., c)$). Thus, for all missing ratings in the target domain, we can construct the index matrix $I \in \mathbb{R}^{c\times k}$ of their top-k similar observed ratings.

3.2 Integrating Prediction Results

Through employing CBT to transfer knowledge from n sources individually, we can get n predicted rating matrices $X = \{X^{(1)}, X^{(2)}, ..., X^{(n)}\}$ and the predicted rating vector $X^{(j)p} \in \mathbb{R}^{c\times 1}$ in the jth source corresponding to R^p. Then, we integrate predicted ratings learnt from different source domains for each missing rating in the target domain.

With the similarity index matrix I of R^p, we can measure the similarity of the predicted ratings in different prediction results and the corresponding missing rating. The weights $W^{(j)} \in \mathbb{R}^{c\times 1}$ of ${X^{(j)}}^p$ can be obtained,

$$W_i^{(j)} = \frac{Sim(R^{I_i}, {X^{(j)}}^{I_i})}{\sum_{j=1}^{n} Sim(R^{I_i}, {X^{(j)}}^{I_i})}, \tag{8}$$

where $W_i^{(j)}$ denotes the weight of predicted rating ${X^{(j)}}_i^p$. I_i denotes the index vector corresponding to the top-k similar ratings of R_i^p. R^{I_i} is the top-k similar ratings of R_i^p in R^t. ${X^{(j)}}^{I_i}$ is the vector corresponding to R^{I_i} in jth domain. $Sim(R^{I_i}, {X^{(j)}}^{I_i})$ denotes the similarity of R^{I_i} and ${X^{(j)}}^{I_i}$ measured via cosine similarity. The integrated prediction results X_{tgt} for R^p can be constructed as follows,

$$X_{tgt} = \sum_{i=1}^{n} W^{(i)} \odot {X^{(i)}}^p, \tag{9}$$

where $\odot$ denotes the dot product and the complete rating matrix can be composed of X_{tgt} and R^o.

4 EXPERIMENTS

4.1 Datasets

We crawl the datasets from two well-known websites Douban[1] and Dianping[2] in China. The Douban dataset contains data of three domains while the Dianping dataset is collected from five cities in China. The rating range of these two datasets are both from 1 to 5. The datasets are detailed in Table 1. For each dataset, we randomly select one domain as the target domain and the rest as the sources. Thus, we construct eight TCF problems from these two datasets.

For convenience, we randomly select the same number of users and items from the five cities on Dianping dataset. Note that users and items in different domains are independent with each other. LOEN does not require the same number or overlapping of users or items in different domains.

Table 1: Details of Douban & Dianping Datasets

Datasets	Domains	# of users/items/ratings	Density
Douban	Movie	3022/6971/195493	0.93%
	Music	5672/6850/550469	1.42%
	Book	9224/9968/699038	0.76%
Dianping	Shanghai	5000/5000/952225	3.81%
	Shenzhen	5000/5000/264721	1.06%
	Hangzhou	5000/5000/605607	2.42%
	Suzhou	5000/5000/492753	1.97%
	Beijing	5000/5000/577665	2.31%

4.2 Experimental Setting

We compare LOEN with the following representative methods.

- NMF [4]: a non-negative matrix factorization algorithm.
- PMF [9]: a probabilistic matrix factorization algorithm.
- CBT [5]: a TCF algorithm transferring the rating pattern from a single source to the target domain.
- CLFM [2]: a TCF method learning the common rating pattern shared across sources and the domain-specific rating patterns in each domain.
- TALMUD [7]: a multi-source TCF method assigning weights dynamically to the prediction rating matrices learnt from different sources.

For each problem, we randomly sample 80%, 60%, 40% and 20% of data in the target domain for training. The random selection is carried out 10 times independently. For single-source TCF methods (i.e., CBT and CLFM), the best results are selected as the final results. Optimal parameters are set for all baselines. We employ widely used Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to evaluate the prediction performance of all algorithms. Smaller values of MAE and RMSE mean better performance.

4.3 Effectiveness Experiments

Due to space limitations, we only illustrate the MAE results in Figure 2. From the results, we observe the followings.

LOEN outperforms all baselines which validates the effectiveness of LOEN. It is reasonable since LOEN utilizes knowledge from multiple domains and integrates the prediction results for each missing rating personally.

The single-source TCF methods (i.e., CBT and CLFM) perform worse than NMF and PMF when data in the source domain is more sparse than that in the target domain (e.g., Dianping-Shanghai). Compared with single-source TCF methods, two multi-source TCF methods (i.e., TALMUD and LOEN) achieve good performances in most cases, since they exploit more information from multiple domains.

TALMUD performs worse than CBT and CLFM on Douban-Movie and Douban-Music. Douban datasets are from three quite different domains. We think the bad performances of TALMUD on Douban datasets are caused by the divergence of sources, since TALMUD simply re-weights source domains without considering the diversity of data. The good performances of LOEN on Douban datasets verify the benefits of assigning personalized weight to each predicted rating.

4.4 Knowledge Transfer across Unrelated Sources

To validate the superiority of LOEN when transferring knowledge across unrelated sources, we employ TALMUD and LOEN on Dianping-Shenzhen dataset with 40% training ratio. We first record their performances on all four source domains (denoted as None in Figure 3), and then we successively replace one of the domains with Douban-Music, Douban-Book and Douban-Movie datasets (denoted as 1src, 2src, and 3src in Figure 3, respectively). As results shown in Figure 3, although these two methods generally perform worse with the increase of the replaced sources, LOEN always performs better than TALMUD. It further confirms the superiority of LOEN to transfer knowledge from diverse sources, since it assigns personalized weight to each predicted rating according to the similarity of their local structures.

[1] https://www.douban.com/
[2] https://www.dianping.com/

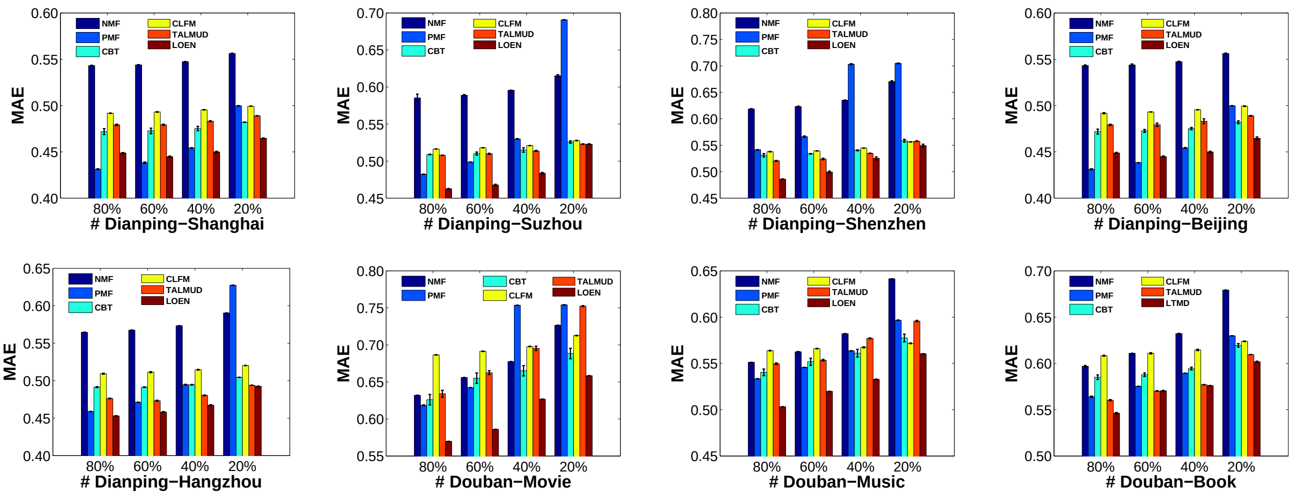
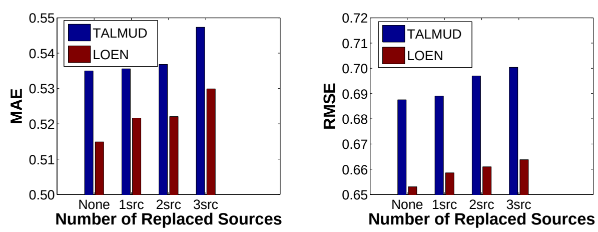


Figure 2: The Results on Douban & Dianping

4.5 Impact of Transfer from Multiple Sources

Further, we explore the impact of transferring knowledge from multiple sources on Dianping datasets. We take one of the five cities in Dianping as the target domain by turns to construct 5 groups of experiments. In each group, we successively add a source domain until all four sources are added. The average results of LOEN on 80% training data are shown in Figure 4. The performance of LOEN improves steadily as more sources are added, which indicates the robustness of LOEN and the effectiveness of utilizing data from multiple domains for improving the experimental performance. Moreover, the overall performance in Shanghai are better than that in Shenzhen, which indicates the importance of data density in the target domain for experimental performance.

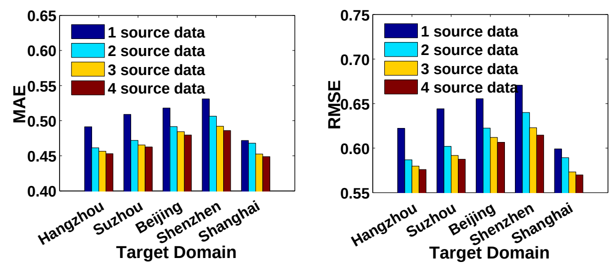


(a) MAE (b) RMSE

Figure 3: Knowledge Transfer from Unrelated Sources

5 CONCLUSIONS

In this paper, considering the divergence of sources, we propose a local ensemble method across multiple sources for collaborative filtering (LOEN), where weights of multiple sources for each missing rating are determined according to their corresponding local structures. Experiments on real datasets validate the effectiveness of the LOEN method.



(a) MAE (b) RMSE

Figure 4: Impact of Transfer from Multiple Sources

ACKNOWLEDGMENTS

This work is supported in part by National Key Basic Research and Department (973) Program of China (No. 2013CB-329606), the National Natural Science Foundation of China (Nos. 61375058, 61772082, 61773361), and the Co-construction Project of Beijing Municipal Commission of Education.

REFERENCES

[1] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *SCIR*. 24.
[2] Sheng Gao, Hao Luo, Da Chen, Shantao Li, Patrick Gallinari, and Jun Guo. 2013. Cross-domain recommendation via cluster-level latent factor model. In *SIGKDD*. 161–176.
[3] Zhao Kang, Chong Peng, and Qiang Cheng. 2016. Top-N Recommender System via Matrix Completion. In *AAAI*. 179–185.
[4] Daniel D Lee and H Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *NIPS*. 556–562.
[5] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction.. In *IJCAI*, Vol. 9. 2052–2057.
[6] Zhongqi Lu, Erheng Zhong, Lili Zhao, Evan Wei Xiang, Weike Pan, and Qiang Yang. 2013. Selective transfer learning for cross domain recommendation. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 641–649.
[7] Orly Moreno, Bracha Shapira, Lior Rokach, and Guy Shani. 2012. Talmud: transfer learning for multiple domains. In *CIKM*. 425–434.
[8] Weike Pan. 2016. A survey of transfer learning for collaborative recommendation with auxiliary data. *Neurocomputing* 177, C (2016), 447–453.
[9] Ruslan Salakhutdinov and Andriy Mnih. 2011. Probabilistic matrix factorization. In *NIPS*, Vol. 20. 1–8.
[10] Zihan Zhang, Xiaoming Jin, Lianghao Li, Guiguang Ding, and Qiang Yang. 2016. Multi-Domain Active Learning for Recommendation. In *AAAI*. 2358–2364.