Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017)

Sparse Overlap Cross-Platform Recommendation via Adaptive Similarity Structure Regularization

Hanqing Lu,[1] Chaochao Chen,[2] Qinyue Jiang[3]
[1]College of Computer Science, Zhejiang University
[2]Ant Financial Services Group, China
[3]School of Management, Zhejiang University
{lhq110, zjuccc}@zju.edu.cn, daisy.jiangqy@foxmail.com

Abstract

People often use multiple platforms to fulfill their different information needs, which has opened opportunities for research on the cross-platform recommendation. Existing cross-platform recommendation works either assume no overlapping users on different platforms or require enough overlapping users to reach a good performance. None of them pays attention to the sparse overlap problem, i.e., the number of observed overlapping users of different platforms is very small. In this paper, we propose a cross-platform recommendation framework termed *Adaptive Similarity Structure Regularization Through Connectors* (AdaSTC), which adaptively learns the user similarity structure on different platforms and further uses it to regularize the modeling process of user preference. Experiments conducted on two real-world datasets demonstrate that AdaSTC significantly outperforms the state-of-the-art methods in the sparse overlap situation.

Introduction

With the emergence and popularity of Web2.0, various social media platforms have been more and more popular due to their different types of services. Now it is common for users to engage in different platforms simultaneously. However, though these information platforms host huge and ever-growing data, they are either isolated or their correlations are significantly undermined. Therefore, how to bridge and integrate multi-platforms information has become a hot research area due to its great potential in promoting users' preference modeling.

Several research works have investigated the cross-platform recommendation problem. For example, FRIENDTRSFER (Yan et al. 2013) assumes that users on two platforms are fully overlapped, which is inconsistent with reality. CM^{3}TM (Min et al. 2015) allows no overlapping users between different platforms but assumes two social platforms share some same topics, which is an overly strong assumption and often not available in real application. Although CrossIntegration (Yan, Sang, and Xu 2015) takes users' partially-overlapping structure between two platforms into consideration, it assumes there are a lot of overlapping users. In general, none of these methods pays attention to the *sparse overlap* problem.

Sparse overlap problem: Sparse overlap problem refers to the number of the observed (explicitly known) overlapping users is very small, which is a widely existing problem in real world applications. For example, Yan *et al*. collected about 40K users (Yan et al. 2013), among which only 3,003 users have both *Flickr* and *Twitter* accounts, and the overlap ratio is less than 1%. Though the sparse overlap problem is very important, there are few studies on this problem in the cross-platform recommendation so far.

In this paper, we propose to use the user similarity structure on the two platforms to regularize to modeling process of user preference. Specifically, we observed that, for every user on one platform, there are some users on other platforms who have considerable similarity with him. In preference modeling process, we should ensure that similar users' preferences embedding are close to each other.

Challenges: However, there are two big challenges in exploiting the user similarity structure to solve the sparse overlap problem. $\mathcal{C}\mathbf{1}$. *Different representations*. Information from different platforms should have different patterns, and simply representing them in the same latent space is inappropriate. $\mathcal{C}\mathbf{2}$. *Unreliable user similarity structure*. The explicit known similarity structure on different platforms is not available in reality. In other words, for a given user on one platform, it is very hard for us to know his similar users on other platforms.

Insights: We provide two insights to address the above challenges. $\mathcal{I}\mathbf{1}$. *Indirect comparison strategy through overlapping users*. We can compare two users' preferences on different platforms by using indirect comparison strategy: for a user on one platform, we first calculate the similarity values between his preference with overlapping users' preferences in the same platform, and then we use a similarity embedding vector that consists of these similarity values to represent this user. After we represent these two users by using the same dimensional similarity embedding vectors, we can compute the distance between them. Figure 1 shows a toy example of this strategy: Alice is a user on Platform A whose latent factor dimension is 3, and Bob is a user on Platform B whose latent factor dimension is 4. We can calculate the distance of their preferences through the overlapping users, i.e., Cater and Dean. $\mathcal{I}\mathbf{2}$. *Adaptive user similarity structure*. Instead of using the determined user similarity structure, we propose to perform user preference modeling

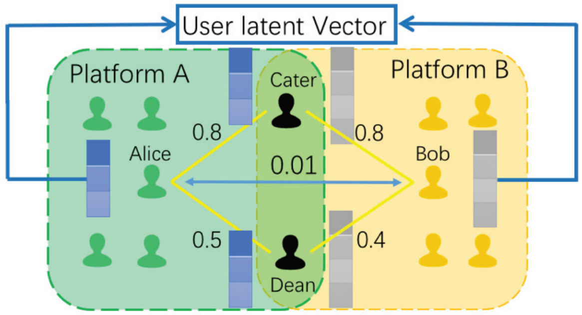


Figure 1: The example of $\mathcal{I}1$: users on Platform A (e.g., Alice) are shown in green, users on platform B (e.g., Bob) are shown in orange, and overlapping users (e.g., Cater) are shown in black. The yellow line between two users is the similarity of their preferences on the same platform, e.g., $Sim(Alice, Cater) = 0.8$. The blue arrow between two platforms' users is their distance, e.g., $Dis(Alice, Bob) = 0.01$. In this case, Alice can be represented by a similarity embedding vector $Vec(Alice) = [0.8, 0.5]^{\mathrm{T}}$, and Bob can be represented by a similarity embedding vector $Vec(Bob) = [0.8, 0.4]^{\mathrm{T}}$. The distance between Alice's and Bob's preferences can be computed by using the following formula: $Dis(Alice, Bob) = Dis(Vec(Alice), Vec(Bob))$.

and user similarity structure learning in a unified framework. That is, we adaptively learn the user similarity structure from the result of user preference modeling and further use it to regularize the learning process of user preference modeling.

Solution: Based on the above two insights, we propose a novel cross-platform user preference learning framework termed *Adaptive Similarity Structure Regularization Through Connectors* (AdaSTC). AdaSTC performs the user similarity structure learning and the user preference modeling simultaneously. That is, AdaSTC directly learns a distance induced probabilistic neighborhood matrix (Nie, Wang, and Huang 2014), where the distance between users' preferences on different platforms is computed with indirect comparison strategy.

We summarize the main contributions of this paper as follows: 1. we identify the sparse overlap problem in the cross-platform recommendation scenario, and propose to exploit cross-platform users' intrinsic similarity structure to alleviate this problem; 2. We conduct extensive experiments on two real-world datasets to validate the effectiveness and efficiency of AdaSTC.

The Method

Notations: Suppose we have two platforms A and B. Platform A has a user set $\mathcal{U}_A = \{u_A^1, ..., u_A^{n_A}\}$ and an item set $\mathcal{V}_A = \{v_A^1, ..., v_A^{m_A}\}$ with n_A and m_A denoting the number of users and items, respectively. Platform B has a user set $\mathcal{U}_B = \{u_B^1, ..., u_B^{n_B}\}$ and an item set $\mathcal{V}_B = \{v_B^1, ..., v_B^{n_B}\}$ with n_B and m_B denoting the number of users and items, respectively. There are n_{AB} observed overlapping users between two platforms A and B, and we denote them by $\mathcal{U}_{AB}$, and usually, $n_{AB} \ll n_A, n_B$, which is the sparse overlap problem. We denote the dimensions of latent factors on two platforms by k_A and k_B respectively. Let $R_A \in \mathbb{R}^{n_A \times m_A}$ and $R_B \in \mathbb{R}^{n_B \times m_B}$ be the user-item rating matrices on two platforms. The corresponding observation indicator matrices are $W_A \in \mathbb{R}^{n_A \times m_A}$ and $W_B \in \mathbb{R}^{n_B \times m_B}$, where $W_A^{ij} = 1$ denotes R_A^{ij} is known and $W_A^{ij} = 0$ denotes R_A^{ij} is missing, and the same as W_B^{ij}. Let $U_A \in \mathbb{R}^{n_A \times k_A}$, $V_A \in \mathbb{R}^{k_A \times m_A}$, $U_B \in \mathbb{R}^{n_B \times k_B}$, and $V_B \in \mathbb{R}^{k_B \times m_B}$ be the latent factor matrices, where each row of U_A represents a user's preference on platform A, each column of V_A represents an item's characteristics on the platform A, and the same meanings as U_B and V_B.

The framework: First, we aim to minimize the error between the known ratings and the predicted ratings on two platforms separately. We choose to model the user preference on two platforms by using matrix factorization (MF) (Koren, Bell, and Volinsky 2009) technique. Next, we bridge the users on two platforms by their similarity structure. That is, in the modeling process, we use a regularization term to make sure that similar users on two platforms have close preferences. Thus, we propose the similarity structure regularization framework as follows:

$$\min_{U_A, V_A, U_B, V_B} \|W_A \otimes (R_A - U_A V_A)\|_F^2 + \tau \|W_B \otimes (R_B - U_B V_B)\|_F^2 + \alpha \sum_{i,j} Dis(U_A^i, U_B^j) H(u_A^i, u_B^j), \tag{1}$$

where α, τ are the trade-off parameters to balance the regularizations and loss functions. We use L2 norms of U_A, V_A, U_B, V_B as regularization terms but omit them for conciseness. $H(u_A^i, u_B^j)$ denotes the intrinsic similarity between users u_A^i and u_B^j, and $Dis(U_A^i, U_B^j)$ denotes the difference between user preferences U_A^i and U_B^j. A bigger value of $H(u_A^i, u_B^j)$ indicates that the distance between user preference vectors U_A^i and U_B^j should be smaller.

Realization of Insight 1: we first calculate the similarity between U_A^i/U_B^j and overlapping users' preference U_A^k/U_B^k ($k \in \mathcal{U}_{AB}$), and then we use a similarity embedding vector consists of these similarity values to represent u_A^i/u_B^j. Let $Vec(u_A^i)$ denotes the similarity embedding vector of u_A^i and $Vec(u_B^j)$ denotes the of similarity embedding vector of u_B^j. Their formal definition are as follows:

$$\begin{aligned} &Vec(u_A^i) = [Sim(U_A^i, U_A^{k(1)}), ...Sim(U_A^i, U_A^{k(n_{AB})})]^{\mathrm{T}}, \\ &Vec(u_B^j) = [Sim(U_B^j, U_B^{k(1)}), ...Sim(U_B^j, U_B^{k(n_{AB})})]^{\mathrm{T}}, \\ &\{k(1), .., k(n_{AB})\} = \mathcal{U}_{AB}, \end{aligned} \tag{2}$$

where $Sim(U_A^i, U_A^k)$ denotes the similarity between user preferences U_A^i and U_A^k. Since both $S(u_A^i)$ and $S(u_B^j)$ are n_{AB}-dimensional vectors, it is easy for us to compute the distance between them. Based on this strategy, we propose the new regularization term in our model as follows:

$$\min_{U_A, U_B} \sum_{i,j} Dis(Vec(u_A^i), Vec(u_B^j)) H(u_A^i, u_B^j) \tag{3}$$

Table 1: Comparison result, with $\eta = 0.8$ and $x = 90$

Dataset	*movie-tag*				*music-tag*			
	RMSE		MAE		RMSE		MAE	
Item	movie	tag	movie	tag	music	tag	music	tag
PMF	0.8187±0.0013	0.2429±0.0010	0.6471±0.0031	0.1847±0.0005	0.7916±0.0055	0.1958±0.0007	0.6247±0.0053	0.1461±0.0009
CBT	0.9065±0.0002	0.1311±0.0023	0.7200±0.0021	0.1222±0.0008	0.8185±0.0051	0.1993±0.0001	0.6232±0.0041	0.1161±0.0003
CROSS	0.8084±0.0091	0.1553±0.0093	0.6373±0.0022	0.1050±0.0030	0.7836±0.0046	0.1857±0.0055	0.6197±0.0081	0.1347±0.0072
CMF	0.8091±0.0051	0.1503±0.0013	0.6424±0.0048	0.1062±0.0003	0.7839±0.0007	0.1901±0.0014	0.6225±0.0005	0.1368±0.0002
DerSTC	0.7867±0.0019	0.1326±0.0045	0.6175±0.0058	0.0951±0.0031	0.7619±0.0027	0.1599±0.0069	0.6017±0.0011	0.1205±0.0004
AdaSTC	**0.7565±0.0023**	**0.1075±0.0022**	**0.6057±0.0045**	**0.0853±0.0040**	**0.7307±0.0098**	**0.1478±0.0015**	**0.5864±0.0060**	**0.1109±0.0033**

In this paper, for computational simplicity, we simply choose $Dis(a,b) = \|.\|_2^2$ and $Sim(c,d) =< c,d >$, where $\|.\|_2$ denotes the $L2$-norm of vector and $< c,d >$ denotes the inner product of vector c and d. Then the regularization term in Equation (3) can be reformulated as:

$$\min_{U_A,U_B} \sum_{i,j,k} (S_A^{ki} - S_B^{kj})^2 H(u_A^i, u_B^j) \tag{4}$$

where $S_A^{ki} =< U_A^k, U_A^i >$ and $S_B^{kj} =< U_B^k, U_B^j >$.

Realization of Insight 2: we introduce to learn a probabilistic neighborhood matrix (Nie, Wang, and Huang 2014) which reveals the similarity structure between cross-platform users. For each user u_A^i on platform *A*, all the users $\{u_B^j\}_{j=1}^{n_B}$ on platform *B* are considered as the neighborhood of u_A^i with the probability P^{ij}, where $P \in \mathbb{R}^{n_A \times n_B}$ can be determined by solving the following problem:

$$\begin{aligned} &\min_P \sum_{i,j} Dis(U_A^i, U_B^j) P^{ij} + \beta \sum_{i,j} (P^{ij})^2 \\ &s.t. \quad 1_n P = 1_n, P \geq 0 \;\; (i \in \mathcal{U}_A, j \in \mathcal{U}_B, k \in \mathcal{U}_{AB}), \end{aligned} \tag{5}$$

where β is the regularization parameter to avoid overfitting. It can be found that a small distance between U_A^i and U_B^j leads to a large P^{ij}. It further indicates that user u_B^j is very likely to be a close neighbor of u_A^i, which means that u_A^i and u_B^j have a large similarity. With such a nice property, the estimated similarity matrix P can be used for user similarity structure embedding.

Finally, we obtain the following objective function based on Equations (1) and (5):

$$\begin{aligned} &\min_{P,U_A,V_A,U_B,V_B} \|W_A \otimes (R_A - U_A V_A)\|_F^2 + \\ &\tau \|W_B \otimes (R_B - U_B V_B)\|^2 + \alpha \sum_{k,i,j} (S_A^{ki} - S_B^{kj})^2 P^{ij} \\ &+ \beta \sum_{i,j} (P^{ij})^2 \\ &s.t. \quad 1_n P = 1_n, P \geq 0 \;\; (i \in \mathcal{U}_A, j \in \mathcal{U}_B, k \in \mathcal{U}_{AB}), \end{aligned} \tag{6}$$

where 1_n denotes a $1 \times n_A$ vector with all elements equal to 1.

The Optimization

We derive an alternative iterative algorithm to solve the Equation (6). First, when U_A, U_B, V_A, and V_B are fixed, the update of P can be formulated as follows:

$$\begin{aligned} &\min_P \alpha \sum_{k,i,j} (S_A^{ki} - S_B^{kj})^2 P^{ij} + \beta \sum_{i,j} (P^{ij})^2 \\ &s.t. \quad 1_n P = 1_n, P \geq 0 \;\; (i \in \mathcal{U}_A, j \in \mathcal{U}_B, k \in \mathcal{U}_{AB}). \end{aligned} \tag{7}$$

Let $C \in \mathcal{R}^{n_A \times n_B}$ be a matrix with $C^{ij} = \frac{\alpha \sum_k (S_A^{ki} - S_B^{kj})^2}{-2\beta}$, and then the above problem can be rewritten as n decoupled sub problems as the following form:

$$\min_{P^i} \frac{1}{2} \|P^i - C^i\|^2 \quad s.t. \;\; 1_n P^i = 1, P \geq 0, \tag{8}$$

where P^i is the i-th column of P.

The above inequality constrained minimization problem can be efficiently solved by Algorithm 1.

Algorithm 1 The optimization algorithm of Equation (8)

Input: C^i
Output: P^i;
1: sort C^i into b where $b_1 \geq b_2 \geq, ..., b_{n_{AB}}$
2: find $\rho = \max\{t \mid b_t + \frac{1}{t}(1 - \sum_{k=1}^t b_k) > 0, 1 \leq t \leq n_{AB}\}$
3: define $z = \frac{1}{\rho}(1 - \sum_{k=1}^{\rho} b_k)$
4: $P^{ki} = \max(C^{ki} + z, 0), 1 \leq k \leq n_{AB}$

Next, when P is fixed, the optimization problem in Equation (8) becomes an unconstraint problem which minimizes:

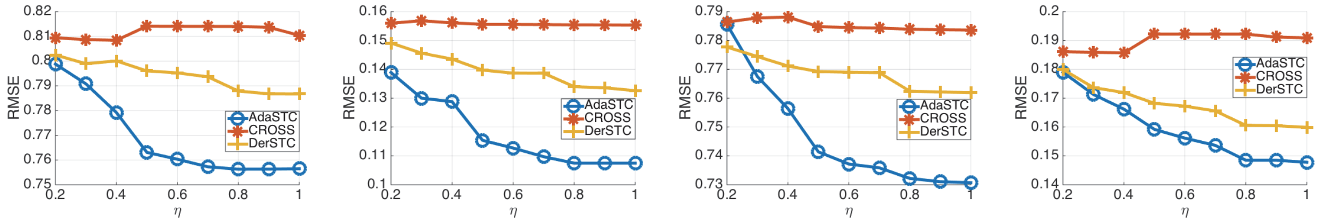
$$\begin{aligned} F = &\|W_A \otimes (R_A - U_A V_A)\|_F^2 + \tau \|W_B \otimes (R_B - U_B V_B)\|^2 \\ &+ \alpha \sum_{k,i,j} (S_A^{ki} - S_B^{kj})^2 P^{ij}. \end{aligned} \tag{9}$$

Because Equation (9) is convex, its solution can be obtained by using the Stochastic Gradient Descent (SGD).

Experiments

Settings

The datasets are crawled from *Douban* and *Sina Weibo*. *Douban* is an online recommendation community in China, which provides the rating and the recommendation service for movies and songs. *Sina Weibo* is a microblogging platform in China, which allows users add personal tags and post tweets. *Sina Weibo* and *Douban* provide APIs which enable users to link their accounts on two platforms. By using the linkage information, we first extract *Sina Weibo* accounts of the *Douban* users, then crawl their tagging information. We process the data into two cross-platform datasets, i.e., the *music-tag* dataset and the *movie-tag* dataset. We randomly select $x\%$ ratings and tags as the training set and report the prediction performance on the remaining 10% test set, where



(a) Predicting movie on *movie-tag* (b) Predicting tag on *movie-tag* (c) Predicting music on *music-tag* (d) Predicting tag on *music-tag*

Figure 2: Results on *Douban-movie* and *Douban-music*, with η varying from 0.2 to 1.

x varies in $\{10, 30, 50, 70, 90\}$. We also randomly select $\eta\%$ users as the observed overlapping users on *Douban* and *Sina Weibo*, where η varies from 0.2 to 1.

Comparison Results and Analysis

We compare AdaSTC with the following recommendation approaches:

PMF (Salakhutdinov and Mnih 2007) is a classic single-platform recommendation method, and we use it to do recommendation on two platforms separately.

CBT (Li, Yang, and Xue 2009) assumes that two domains share the same rating patterns with no overlapping users.

CROSS (Yan, Sang, and Xu 2015) takes user partially overlapping structure on two platforms into consideration, and it models overlapping users' preference from the information on both platforms.

CMF (Singh and Gordon 2008) simultaneously factorizes several matrices, and shares parameters among factors when an entity participates in multiple relations.

DerSTC is a variant of our method without the realization of $\boldsymbol{\mathcal{I}}\mathbf{2}$. That is, it uses the determined similarity structure calculated by using users' profiles.

We first use the grid search to determine the optimal hyperparameters for all the baselines and AdaSTC, and we set $\tau = 0.2, k_A = 15$, and $k_B = 10$ for AdaSTC. Then we conduct the experiment for 10 times and report the average RMSE and MAE on the test set with variance. The comparison results with $x = 90$ are shown in Table 1. From them, we have the following observations: 1. DerSTC achieves the best performance among all the baseline algorithms on both datasets, which demonstrates the effectiveness of the proposed user similarity structure regularization framework; 2. AdaSTC achieves the best performance in terms of the RMSE and the MAE on both datasets, which further demonstrates that our method's strength.

Sparse Overlap Problem Experiments

Since only CROSS and DerSTC can handle the partial overlapping structure between cross-platform users, we compare AdaSTC with these baseline methods to test their capability of handling the sparse overlap problem. We vary η from 0.2 to 1 to study the influence of the number of overlapping users on our model behavior. The results are shown in Figure 2, and from it, we have the following observations: 1. with different overlap ratios, AdaSTC always achieves the best performance, which demonstrates that our method is efficient to solve the sparse overlap problem; 2. with η varying from 0.2 to 0.8, the increasing number of overlapping users will boost the performance of AdaSTC and DerSTC, while it does not have significant influence on CROSS. This result indicates that more overlapping users will help conduct a better indirect comparison.

Conclusion

In this paper, we first identified the sparse overlap problem. Then we proposed a novel cross-platform recommendation algorithm termed AdaSTC to solve it. Specifically, AdaSTC adopts the user similarity structure on different platforms to regularize the modeling process of user preference. Experimental results conducted on two real-world datasets demonstrate that our proposed AdaSTC significantly outperforms the state-of-the-art methods under the sparse overlap situation.

References

Koren, Y.; Bell, R. M.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer*.

Li, B.; Yang, Q.; and Xue, X. 2009. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *IJCAI*, 2052–2057.

Min, W.; Bao, B.; Xu, C.; and Hossain, M. S. 2015. Cross-platform multi-modal topic modeling for personalized inter-platform recommendation. *IEEE Transactions on Multimedia* 1787–1801.

Nie, F.; Wang, X.; and Huang, H. 2014. Clustering and projected clustering with adaptive neighbors. In *SIGKDD*, 977–986.

Salakhutdinov, R., and Mnih, A. 2007. Probabilistic matrix factorization. In *NIPS*, 1257–1264.

Singh, A. P., and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *SIGKDD*, 650–658.

Yan, M.; Sang, J.; Mei, T.; and Xu, C. 2013. Friend transfer: Cold-start friend recommendation with cross-platform transfer learning of social knowledge. In *ICME*, 1–6.

Yan, M.; Sang, J.; and Xu, C. 2015. Unified youtube video recommendation via cross-network collaboration. In *ICMR*, 19–26.