Information Processing and Management 58 (2021) 102691

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

A selective ensemble learning based two-sided cross-domain collaborative filtering algorithm

Xu Yu [a], Qinglong Peng [a], Lingwei Xu [a], Feng Jiang [a], Junwei Du [a], Dunwei Gong [a,b,*]

[a] *School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China*
[b] *School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China*

ARTICLE INFO

Keywords:
Cross-domain collaborative filtering
Selective ensemble
Ensemble learning
Pareto optimal solutions
Bi-objective optimization problem

ABSTRACT

Recently, various Cross-Domain Collaborative Filtering (CDCF) algorithms are presented to address the sparsity problem, leveraging ratings of auxiliary domains to improve target domain's recommendation performance. Therein, two-sided CDCF algorithms have shown better performance, given the fact that they can extract both user and item information. However, as the auxiliary domains are not all related to the target domain, utilizing information from all the auxiliary domains may not be optimal and would lead to low efficiency. A **T**wo-**S**ided CDCF model based on **S**elective **E**nsemble learning considering both **A**ccuracy and **E**fficiency (TSSEAE) is proposed to balance recommendation accuracy and efficiency. In TSSEAE, user-sided and item-sided auxiliary domains are firstly combined to improve performance of target domain. Then, CDCF problems are converted to ensemble learning problems, with each combination corresponding to a classifier. In this way, the problem of selecting combinations can be converted to that of selecting classifiers, which is a selective ensemble learning problem. Finally, a bi-objective optimization problem is solved to obtain Pareto optimal solutions for the selective ensemble learning problem. The experimental result on Amazon dataset shows the effectiveness of TSSEAE.

1. Introduction

Recently, as the most successful models of artificial intelligence (Liu et al., 2020, 2018, 2015; Mollah et al., 2021; Xi, Wang, Yao, & Zhang, 2021; Xu, Yu and Gulliver, 2021; Xu et al., 2021), recommendation models are widely applied in real life with the purpose of providing personalized services and overcoming information overload. Collaborative filtering (CF) models (Fu, Qu, Yi, Lu, & Liu, 2018; Liu, Feng, Wang, & Zuo, 2021; Yu, Quan, Peng, Yu and Liu, 2021) and content-based models (Balabanović & Shoham, 1997; Oppermann, Kincaid, & Munzner, 2021; Van Den Oord, Dieleman, & Schrauwen, 2013) are two kinds of recommendation models. CB models recommend similar items according to the attribute information of items, but they depend on the contents of items. In contrast, CF models, which are independent of contents, recently have gained dominance. However, CF always suffers from the sparsity problem since customers generally dislike rating items in real applications.

To address the sparsity problem of CF models, cross-domain CF (CDCF) has been recently proposed, which improves performance by exploiting ratings of auxiliary domains. According to information shared among different domains, the existing CDCF models are divided into: user-sided transfer (Berkovsky, Kuflik, & Ricci, 2007; Hu et al., 2013; Loni, Shi, Larson, & Hanjalic, 2014; Yu et al., 2021), item-sided transfer (Singh & Gordon, 2008), two-sided transfer (Pan, Xiang, Liu, & Yang, 2010; Yu, Chu, Jiang, Guo, & Gong, 2018; Yu, Jiang, Du, & Gong, 2019), and non-sided transfer (Li, Yang, & Xue, 2009a, 2009b; Yu et al., 2021; Zhang, Wu, Lu,

* Corresponding author at: School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China.
E-mail address: dwgong@vip.163.com (D. Gong).

https://doi.org/10.1016/j.ipm.2021.102691
Received 8 April 2021; Received in revised form 9 July 2021; Accepted 10 July 2021
Available online 29 July 2021
0306-4573/

Liu, & Zhang, 2017). Among them, user- and item-sided transfers belong to one-sided transfer. In the case of one-sided transfer, the auxiliary domains have the same users or items with the target domain, which are called user- or item-sided auxiliary domains. With respective to two-sided transfer, there exist both user- and item-sided auxiliary domains. In contrast, non-sided transfer requires neither shared users nor shared items. Previous studies have shown that two-sided transfer can achieve good recommendation performances since they exploit both useful user and item information. However, existing CDCF models utilize all the auxiliary domains to improve performance of target domain. Given the fact that auxiliary domains are not all related to target domain, recommendation performances achieved using all the auxiliary domains may not be optimal. Moreover, making recommendations by utilizing all the auxiliary domains also lead to low efficiency. Hence, to improve recommendation performance of CDCF, we have to address the following two challenges.

(1) How to select significant subsets from all the auxiliary domains to improve the recommendation accuracy?

(2) How to balance recommendation accuracy and efficiency?

A **T**wo-**S**ided CDCF model based on **S**elective **E**nsemble learning considering both **A**ccuracy and **E**fficiency (TSSEAE) is proposed to over the two challenges. In TSSEAE, user- and item-sided auxiliary domains are combined to improve performance of target domain. As there are $m \times n$ combinations with respect to m user- and n item-sided auxiliary domains, our aim is to figure out which combinations should be selected to balance recommendation accuracy and efficiency? In TSSEAE, the CDCF problem is converted to an ensemble learning problem, with each combination corresponding to a classifier. In this way, the problem of selecting combinations can be converted to that of selecting classifiers, which can be deemed as a selective ensemble learning problem. Moreover, to balance recommendation accuracy and efficiency, TSSEAE solves a bi-objective optimization problem for the selective ensemble learning problem. Extensively experimental results show that TSSEAE performs better than other state-of-the-art models. The main contributions are as follows.

(1) We investigate that recommendation by utilizing a subset of auxiliary domains performs better than leveraging all of them.

(2) We convert the subset selection problem to a selective ensemble learning problem.

(3) We solve the bi-objective optimization problem to balance accuracy and efficiency.

The remainder is organized in the following. We review related work in Section 2. Section 3 describes the proposed algorithm, TSSEAE, to balance accuracy and efficiency. Section 4 carries out extensive experiments to test TSSEAE and presents detailed experimental analysis. We present the conclusion and future research topics in Section 5.

2. The related work

Existing CDCF are divided into user-sided transfer (Berkovsky et al., 2007; Hu et al., 2013; Loni et al., 2014; Yu, Zhan et al., 2021), item-sided transfer (Singh & Gordon, 2008), two-sided transfer (Pan et al., 2010; Yu et al., 2018, 2019), and non-sided transfer (Li et al., 2009a, 2009b; Yu, Hu et al., 2021; Zhang et al., 2017).

As for one-sided transfer, Berkovsky et al. (2007) and Singh and Gordon (2008) presented Neighbor-based CDCF (N-CDCF) and Collective Matrix Factorization (CMF) model, respectively. They are cross-domain version of memory-based and matrix factorization-based CF algorithms, respectively. Hu et al. (2013) presented a tensor factorization based CDCF model, called CDTF.Loni et al. (2014) and Yu, Zhan et al. (2021) proposed cross-domain models based on Factorization Machine (FM) model (Rendle, 2010).

Along the line of non-sided transfer, Li et al. (2009a, 2009b) presented the CBT and RMGM models, which extract the cluster-level rating mode from the rich rating information in the auxiliary domain.

For two-sided transfer, Pan et al. (2010) presented CST, which extracts coordinate systems from auxiliary domains and incorporates them into the target factorization system. However, CST can only address the case of homogeneous items across different domains. Also, Yu et al. (2018, 2019) proposed CTSIF and TSEUIF, in which domain-independent and domain-dependent features are transferred to help the recommendation of the target domain, respectively.

Traditional CDCF (Berkovsky et al., 2007) models utilize all the user- and item-sided auxiliary domains to improve recommendation performance. However, not all the auxiliary domains are closely related to the target domain. Hence, traditional CDCF models by utilizing all the auxiliary domains may not achieve the best recommendation accuracy. In addition, efficiency is also a significant metric for recommendation. Obviously, selecting only a subset from all the auxiliary domains will achieve a high efficiency. Thus it is interesting to figure out which combinations can balance the recommendation accuracy and efficiency.

3. The proposed algorithm

For the case of m user- and n item-sided auxiliary domains, there will be $m \times n$ user and item combinations. As auxiliary domains are not all related with target domain, selecting a number of appropriate combinations rather than all the combinations may be beneficial to the recommendation performances in the target domain. As mentioned previously, in this section, we will study how to select a good subset from $m \times n$ combinations available by considering both recommendation accuracy and efficiency. Recommendation accuracy can be evaluated with MAE or other indicators over a test dataset, and efficiency is related to the size of the subset. A smaller size means a higher efficiency. As a result, the problem of selecting a subset bears two simultaneous objectives, minimizing the MAE value and the size of the subset, simultaneously. Since the two objectives contradict each other, the problem of selecting an optimal subset belongs to a bi-objective optimization problem. The notations we use in this paper are summarized in Table 1.

Table 1
Notations and explanations.

Notation	Explanation
D_0	Target domain
D^u	User-sided auxiliary domain
D^i	Item-sided auxiliary domain
$\mathbf{M}_0$	Rating matric of target domain
$\mathbf{M}^u$	Rating matric of user-sided auxiliary domain
$\mathbf{M}^i$	Rating matric of item-sided auxiliary domain
U_0	User set of target domain
I_0	Item set of target domain
ID_u	ID number of user u
ID_i	ID number of item i
r	Rating
$\mathbf{U}$	Intrinsic feature matric of users
$\mathbf{V}$	Intrinsic feature matric of items
$\mathbf{B}$	Interaction of intrinsic features
H	Set of base classifiers
s	Selection vector
H_s	A pruned ensemble
f	Performance measure

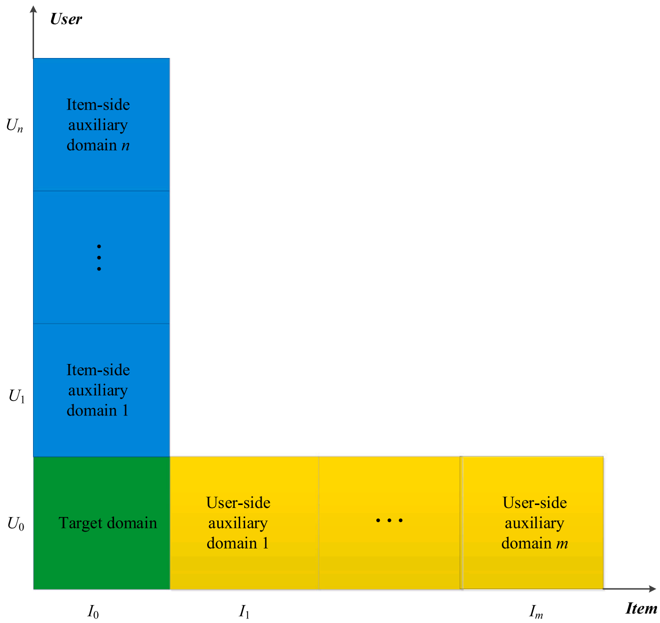

Fig. 1. An illustration of two-side transfer with two or more user- and item-sided auxiliary domains.

3.1. Problem formulation

As mentioned before, we consider the scenario of two-sided transfer with two or more user- and item-sided auxiliary domains. Let D_0, $D^u_1, \ldots, D^u_m$, and $D^i_1, \ldots, D^i_n$ be the target domain, the user- and item-sided auxiliary domains, respectively, $\mathbf{M}_0$, $\mathbf{M}^u_1, \ldots, \mathbf{M}^u_m$, and $\mathbf{M}^i_1, \ldots, \mathbf{M}^i_n$ be the corresponding rating matrices. The scenario can be illustrated in Fig. 1, where there are $m \times n$ combinations in total to be chosen to improve the recommendation performances in D_0.

3.2. Model construction

In this section, we will introduce the proposed model. First, we convert recommendation into classification. Then, user and item features are expanded by Non-negative Matrix Factorization. Finally, we convert the problem of cross-domain recommendation into that of ensemble learning.

3.2.1. Converting recommendation into classification

Let U_0 and I_0 be user and item sets in D_0, respectively. In Fig. 2, we convert recommendation problem in D_0 as classification, where each user–item interaction, $(u, i, r) \in U_0 \times I_0 \times \{1, 2, 3, 4, 5\}$ can be represented with a feature vector, $(ID_u,\ ID_i)$, and a class

	d_1	d_2	d_3	d_4
t_1	5	?	4	?
t_2	?	5	?	1
t_3	2	?	3	4

ID_t	ID_d	label
1,0,0	1,0,0,0	5
1,0,0	0,0,1,0	4
0,1,0	0,1,0,0	5
0,1,0	0,0,0,1	1
0,0,1	1,0,0,0	2
0,0,1	0,0,1,0	3
0,0,1	0,0,0,1	4

Fig. 2. Converting recommendation into classification.

label, *r*, where ID_u and ID_i are the ID numbers, represented with one-hot encoding, respectively. One-hot encoding is generally better than the discrete numeralization approach due to no significance in the size among the ID feature values.

3.2.2. Expanding user and item features from each combination

As sample set in Fig. 2 has just two trivial location feature vectors, it is difficult to train a good classifier to solve the classification problem. We employ Non-negative Matrix Factorization (NMF) to extract intrinsic users' and items' features from auxiliary domains (Ding, Li, Peng, & Park, 2006; Yu et al., 2018) to expand user and item features. We solve the following optimization problem,

$$\begin{aligned} \min_{\mathbf{U}\geq 0,\mathbf{V}\geq 0,\mathbf{B}\geq 0} \quad & T = \left\|\mathbf{M} - \mathbf{U}\mathbf{B}\mathbf{V}^T\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{U}^T\mathbf{U}=\mathbf{I}, \mathbf{V}^T\mathbf{V}=\mathbf{I} \end{aligned} \tag{1}$$

where **M** denotes the rating matrix, **U** and **V** represent intrinsic feature matrices, and **B** is associated with the interaction of intrinsic features. For **U** and **V**, we expect them to be nonnegative, and the same is true for **B**. In addition, imposing orthogonality on **U** and **V** is beneficial to obtaining a unique solution.

Since **M** in the optimization problem is incomplete, we first adopt the average rating in each row to fill all missing ratings. Then, we solve the optimization problem in the following according to Ding et al. (2006).

$$(\mathbf{V})_{ij} \leftarrow (\mathbf{V})_{ij}\sqrt{\frac{(\mathbf{M}^T\mathbf{U}\mathbf{B})_{ij}}{(\mathbf{V}\mathbf{V}^T\mathbf{M}^T\mathbf{U}\mathbf{B})_{ij}}} \tag{2}$$

$$(\mathbf{U})_{ij} \leftarrow (\mathbf{U})_{ij}\sqrt{\frac{(\mathbf{M}\mathbf{V}\mathbf{B}^T)_{ij}}{(\mathbf{U}\mathbf{U}^T\mathbf{M}\mathbf{V}\mathbf{B}^T)_{ij}}} \tag{3}$$

$$(\mathbf{B})_{ij} \leftarrow (\mathbf{B})_{ij}\sqrt{\frac{(\mathbf{U}^T\mathbf{M}\mathbf{V})_{ij}}{(\mathbf{U}^T\mathbf{U}\mathbf{B}\mathbf{V}^T\mathbf{V})_{ij}}} \tag{4}$$

The Intrinsic Feature Inference (IFI) algorithm is given as follows.

Algorithm 1 IFI

Input: the matrix, **M**, dimensionalities for latent vectors k and l.
Output: intrinsic feature matrices **U**, and **V**.

```
Fill missing values in M
Randomly initialize U^(0), V^(0), and B^(0). Let ΔT=1, s=1
while ΔT > 10^-3 do
    Update U^(s-1), V^(s-1), and B^(s-1) using Eqs. (2-4), and obtain U^(s), V^(s), and B^(s)
    ΔT = T^(s) - T^(s-1)
    s = s + 1
end while
s = s - 1, U = U^(s), and V = V^(s)
```

3.2.3. Converting the problem of cross-domain recommendation into that of ensemble learning

For a combination, (D_s^u, D_t^i), $s = 1, \ldots, m$, $t = 1, \ldots, n$, the one-hot embedding feature vector, (ID_u, ID_i), of a user–item interaction in the target domain, can be extended to $(ID_u, ID_i, U_s^u, V_t^i)$, where U_s^u denotes a vector associated with the latent factor membership of user u, V_t^i is a vector associated with the latent factor membership of item i, D_s^u is user-sided auxiliary domain s, and D_t^i denotes item-sided auxiliary domain t.

In this way, for the above combination, (D_s^u, D_t^i), we can train a classifier corresponding to the constructed training set. It is clear that we can train $m \times n$ classifiers on the constructed training sets corresponding to all the combinations.

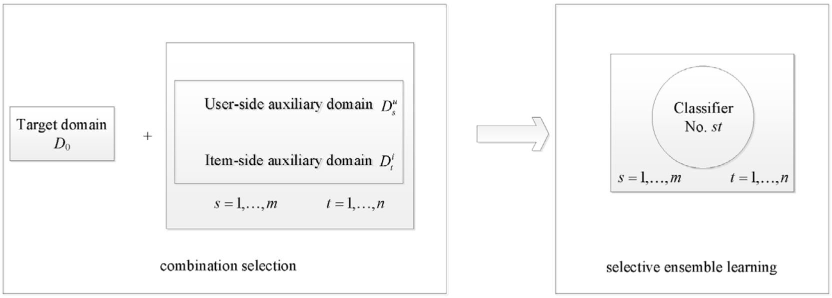

Fig. 3. The relationship between combination selection and selective ensemble learning.

The above process indicates that the problems of cross-domain recommendation in the two-sided transfer scenario can be converted into that of ensemble learning (Hansen & Salamon, 1990). As a result, selecting appropriate combinations is equivalent to selecting appropriate classifiers in the corresponding ensemble learning problem, which belongs to selective ensemble learning (Zhou, Wu, & Tang, 2002). Their relationship can be clearly demonstrated in Fig. 3.

3.3. Model solving

3.3.1. Formulating a bi-objective optimization problem

As mentioned previously, the size of a good subset should be as small as possible, and the recommendation accuracy based on the subset should be as high as possible. In this paper, each combination is associated with a classifier. Hence, maximizing the recommendation accuracy and minimizing the size of the selected subset are equivalent to maximizing the classification accuracy and minimizing the number of selected classifiers, respectively.

We now turn to the matter of seeking the best subset from all the classifiers. Given a data set, $\mathrm{Data} = \{(x_i, y_i)\}_{i=1}^{m}$, and a set composed of *n* trained base classifiers, $H = \{h_i\}_{i=1}^{n}$, where $h_i : \mathcal{X} \rightarrow \mathcal{Y}$ maps the features to the label. Let H_s be a pruned ensemble, where $s \in \{0,1\}^n$ is the selection vector, and $s_i = 1$ means that h_i is selected. Our aim is to simultaneously maximize the classification accuracy of H_s and minimize size of H_s calculated as $|s| = \sum_{i=1}^{n} s_i$.

To achieve a solution with a high classification accuracy and a small size of selected classifiers, previous ensemble pruning approaches generally solve the above problem by aggregating the two objectives. However, for multi-objective optimization problems, explicit consideration of each objective is quite helpful (Qian, Yu, & Zhou, 2015; Yu, Yao, & Zhou, 2012). Hence, the problem of selecting classifiers is equivalent to the following bi-objective optimization problem,

$$\arg\min_{s\in\{0,1\}^n} \left(f\left(H_s\right), |s|\right) \tag{5}$$

For multi-objective optimization problems, one solution may be better than the other on one dimension, whereas worse than its counterpart on the other dimension(s). On this circumstance, the domination relationship is usually employed to compare two candidates, whose definition in the bi-objective optimization case is given as follows.

Definition 1 (*Domination*)**.** Let $g = (g_1, g_2)$: $S \rightarrow \mathbb{R}^2$ be the objective vector. For two solutions, s, $s' \in S$:
(1) s weakly dominates s', if $g_1(s) \le g_1(s')$ and $g_2(s) \le g_2(s')$, denoted as $s \succeq_g s'$;
(2) s dominates s', if $s \succeq_g s'$ and either $g_1(s) < g_1(s')$ or $g_2(s) < g_2(s')$, denoted as $s \succ_g s'$.

Consequently, a bi-objective optimization problem generally has many *Pareto optimal* solutions.

3.3.2. Solving the optimization problem based on PEP

We employ Pareto Ensemble Pruning (PEP) (Qian et al., 2015), shown as Algorithm 2, to solve the bi-objective ensemble pruning, since it is a good model for selective ensemble learning based on bi-objective optimization.

Algorithm 2 PEP

Input: a set of trained classifiers, $H = \{\mathrm{h}_i\}_{i=1}^{n}$, an objective, *f*, criterion, eval.
Output: the optimal solution, s_{opt}.
Let $g(s) = \left(f\left(H_s\right), |s|\right)$ be objectives
Let s be a randomly selected solution from $\{0,1\}^n$, and $P = \{s\}$
Let $iteration = 1$
repeat
  Randomly select $s \in P$
  Generate s' by flipping each bit of s with the probability of $\frac{1}{n}$

7: **if** $\nexists z \in P$ such that $z \succ_g s'$ **then**
8: Let $P = \left(P - \left\{z \in P \middle| s' \succeq_g z\right\}\right) \cup \{s'\}$ and $Q = VDS\left(f, s'\right)$.
9: **for** $q \in Q$ **do**
10: **if** $\nexists z \in P$ such that $z \succ_g q$ **then**
11: Let $P = \left(P - \left\{z \in P \middle| q \succeq_g z\right\}\right) \cup \{q\}$
12: **end if**
13: **end for**
14: **end if**
15: Let $iteration = iteration + 1$
16: **until** $iteration > n^2 * \log(n)$
17: Let $s_{opt} = \min_{s \in P} eval(s)$

In Algorithm 2, PEP first generates a solution randomly, and puts it into a set, P. Then, a loop is conducted to iteratively improve solutions in P.

In Line 8 of PEP, a local search approach, i.e., variable-depth search (VDS) (Lin & Kernighan, 1973), is used to improve the new candidate solution, which is beneficial to improving the efficiency of PEP. The details of VDS are described in as follows. In Line 16 of PEP, according to the implementation in Ref. Qian et al. (2015), the PEP algorithm will stop when the number of iterations reaches to $n^2 * \log(n)$, where n is the number of base classifiers.

Algorithm 3 VDS

Input: Given a function, f, and a solution, s.
Output: the solutions obtained by a local search according to s.
1: Let $Q = \emptyset$, $L = \emptyset$, $N(\cdot)$ be the set of neighbor solutions of a binary vector with the Hamming distance of 1
2: **while** $V_s = \left\{y \in N(s) \middle| \left(y_i \neq s_i \Rightarrow i \notin L\right)\right\} \neq \emptyset$ **do**
3: Choose $y \in V_s$ with the minimal value of f
4: Let $Q = Q \cup \{y\}$, $L = L \cup \left\{i \middle| y_i \neq s_i\right\}$
5: Let $s = y$
6: **end while**

Since it is difficult to measure the generalization performance directly, we employ the error directly on a validation data set as an alternative way. When we obtain Pareto optimal solutions to the ensemble pruning optimization problem, we will determine the final solution according to the *eval* function in Line 12 of Algorithm 2.

To fulfill this task, selecting the evaluation criterion, *eval*, is related to application. If the algorithm is sensitive to the number of classifiers, the selection should prefer those with a small number; otherwise, it should lean to those with a good performance measure. In this paper, we use a comprehensive criterion in the following form to select one final solution.

$$eval = 20f\left(H_s\right) + |s| \tag{6}$$

3.3.3. The proposed TSSEAE algorithm

According to Algorithms 1 and 2, the proposed TSSEAE algorithm is given as follows. In Line 1, we convert recommendation into classification, and obtain a sample set, T_0. Lines 2–4 compute intrinsic user features according to IFI. Lines 5–6 compute intrinsic item features according to IFI. In Line 12, for the st-th combination, we expand the user and item features, (ID_u, ID_i), to $(ID_u, ID_i, U^u_s, V^i_t)$ for each sample, and obtain a new sample set, T_e. In Lines 13–14, $m \times n$ classifiers are trained with SVMs on the $m \times n$ sample sets. Line 16 solves the ensemble learning problem with PEP, and obtains the best solution. In Line 17, we apply ensemble learning on $H_{s_{opt}}$ to obtain the recommendation results.

Algorithm 4 TSSEAE

Input: Given rating matrices, $\mathbf{M}_0$, $\mathbf{M}^u_1, \cdots, \mathbf{M}^u_m$, and $\mathbf{M}^i_1, \cdots, \mathbf{M}^i_n$.
Output: Recommendation results.
1: $T_0 = R2C(M_0)$
2: **for** $s = 1 : m$ **do**
3: U_s=IFI([$\mathbf{M}^u_s$)
4: **end for**
5: **for** $t = 1 : n$ **do**
6: V_t=IFI($\mathbf{M}^i_t$)
7: **end for**
8: **for** $s = 1 : m$ **do**
9: **for** $t = 1 : n$ **do**
10: T_e[st]=feature_expansion (T_0, U_t , V_s)
11: **end for**

12: **end for**
13: **for** $k = 1 : mn$ **do**
14: $h[k] = SVMs(T_e[k])$
15: **end for**
16: $s_{opt} = PEP(H, f, eval)$
17: recommendation_result = ensemble_learning($H_{s_{opt}}$)

4. Experiments

Extensive experiments are conducted on Amazon dataset to investigate the performance of TSSEAE. The algorithms involved are implemented with Python 3.6 on one open-source machine learning library: Scikit-learn.

4.1. The setting of the compared methods

Funk-SVD: It is the single-domain latent factor model. In this paper, we set the learning rate to $\frac{1}{\tau}$, so as to improve the convergence speed, where τ is the number of iterations. We also give the explanation in the revised manuscript. The latent factor dimensionality k and the regularization parameter λ are tried in {5,10,15,20,25,30,35,40}, and {0.001,0.01,0.1,1,10,100,1000}, respectively. Best parameter values are computed through 2-fold cross-validation. That is to say, the original data is randomly divided into two groups. Firstly, the first group is used for training, and the second group is used for validation to compute the test metric of the model. Then, the second group is used for training, and the first group is used for validation. Finally, we calculate the average value of the test metric on the two cases, and select the parameters corresponding to the optimal test metric value as the optimal parameters.

CMF: It is the cross-domain model of Funk-SVD, whose parameters are setted in the same way with Funk-SVD.

CDTF: Tensor factorization is used to model the "user–item–domain" triadic relationship. Its parameter setting follows the setting in Ref. Hu et al. (2013).

CTSIF: It is the two-sided cross-domain recommendation algorithm with respect to one user-sided auxiliary domain and one item-sided auxiliary domain. Its parameters follow the settings of Ref. Yu et al. (2018).

TSEUIF: It is the two-sided cross-domain recommendation algorithm for more than one user-sided auxiliary domains and more than one item-sided auxiliary domains. Its parameters follow the setting in Ref. Yu et al. (2019).

ALL: It is the two-sided CDCF algorithm based on ensemble learning on all the combinations.

TSSEA: It is similar to the proposed TSSEAE, which only consider the recommendation accuracy.

TSSEAE: It is the proposed algorithm. The dimensionalities of latent vectors of user and item are tried from {5, 10, 15, 20, 25, 30, 35, 40}. We select support vector machines as the base classifiers, and the parameters, C and γ, follow the same setting as CTSIF_SVMs (Yu et al., 2018).

4.2. Evaluation indicators

MAE, RMSE, Precision, and Recall in the following are selected as evaluate metrics:

$$MAE=(\sum_{i\in T} |r_i - \tilde{r}_i|)/|T| \tag{7}$$

$$RMSE=\sqrt{\sum_{i\in T} (r_i - \tilde{r}_i)^2/|T|} \tag{8}$$

$$Precision = \frac{\sum_{u\in U} |R(u) \cap T(u)|}{\sum_{u\in U} |R(u)|} \tag{9}$$

$$Recall = \frac{\sum_{u\in U} |R(u) \cap T(u)|}{\sum_{u\in U} |T(u)|} \tag{10}$$

where T represents the set of test ratings, r_i and $\tilde{r}_i$ are the ground truth and the predicted ratings, respectively, and $T(u)$ and $R(u)$ denote the true and predicted recommendation lists.

4.3. Data preparation for MAE and RMSE

The experimental data released by Amazon come from the rating data of different users on 24 different kinds of products. The original data include Amazons approximately 142.8 million product reviews and metadata, and we only use the rating data among them.

This experiment requires that the target domain rating data is sparse, and the auxiliary domain rating data is rich. As the rating densities of rating matrices in different domains of the original data are generally low, in order to meet the experimental requirements, we first select subsets of users and items with rich rating data to construct the corresponding auxiliary domain and target domain data. Furthermore, to meet the sparsity of the target domain rating matrix, 25%, 30%, 35%, and 40% of the original target domain data are randomly chosen as training set of target domain, denoted as TR25, TR30, TR35, and TR40, respectively.

Table 2
The selected target and user-sided auxiliary domains.

Domain	Item type
Target domain	Electronics
User-sided auxiliary domain 1	Grocery_and_Gourmet_Food
User-sided auxiliary domain 2	Health_and_Personal_Care
User-sided auxiliary domain 3	Home_and_Kitchen

Table 3
Statistics information in different domains.

Domain		# of users	# of items	Avg. # of ratings for each item		Avg. # of ratings for each user		Density (%)	
Target domain	TR25	193	700	7	2	25	7	3.60%	1.00%
	TR30				2		8		1.20%
	TR35				2		9		1.40%
	TR40				3		10		1.50%
User-sided auxiliary domain 1		193	367	16		30		8.30%	
User-sided auxiliary domain 2		193	623	14		45		7.20%	
User-sided auxiliary domain 3		193	290	8		12		4.30%	
Item-sided auxiliary domain 1		260	700	5		19		2.70%	
Item-sided auxiliary domain 2		313	700	5		18		2.60%	
Item-sided auxiliary domain 3		340	700	5		19		2.80%	

Table 4
The values of MAE.

Algorithm	TE75	TE70	TE65	TE60
Funk-SVD	1.199	1.141	1.068	0.989
CMF	0.907	0.876	0.842	0.821
CDTF	0.871	0.862	0.829	0.813
CTSIF	0.865	0.832	0.818	0.816
TSEUIF	0.781	0.737	0.728	0.711
ALL	0.673	0.639	0.629	0.622
TSSEA	0.656	0.631	0.616	0.609
TSSEAE	0.669	0.635	0.623	0.614

The remaining, denoted as TE75, TE70, TE65, and TE60, are used as the test sets, respectively. The selected target and user-sided auxiliary domains are shown in Table 2. The statistics information of different domains is shown in Table 3.

For Funk-SVD, we test it on the target domain because it belongs to single domain CF algorithms. Since CMF, CDTF are one-sided transfer algorithms, we test them on Target domain, User-sided auxiliary domains 1–3. Since CTSIF is a two-sided transfer algorithm and can only address one user-sided auxiliary domain and one item-sided auxiliary domain, we test it on the target domain, the user-sided auxiliary domain 1, and the item-sided auxiliary domain 1. Since TSEUIF, ALL, TSSEA, and the proposed TSSEAE algorithm can deal with multiple user-sided and multiple item-sided auxiliary domains, we test them on Target domain, and all auxiliary domains.

4.4. Data preparation for precision and recall

We use the same training set as mentioned previously. To compute precision and recall values, test sets are constructed as follows. We first select users with more than 10 ratings from TE75, TE70, TE65, and TE60. Then, for each user with more than 10 ratings, we randomly select 10 items rated to form the test sets. Finally, ratings 1–5 are mapped to two categories, 'liked' and 'disliked'. Items whose ratings are ≥ 3 are labeled as 'liked'; otherwise, 'disliked'. We choose items with top N ratings for recommendation. In our experiment, we set N = 2, 4, 6, so we can compute precision and recall values.

4.5. Experimental results and analysis

We determine the optimal parameters through cross-validation (Kohavi et al., 1995). Figs. 4 and 5 show the average MAE values for different parameters of IFI and SVM algorithms. Note that, for simplicity, Fig. 5 only lists the results for the combination of user-sided auxiliary domain 1 and item-sided auxiliary domain 1. Also, the iteration processes of IFI algorithm corresponding to 6 auxiliary domains are shown in Fig. 6. From Fig. 6, we can find that the values of objective functions decrease rapidly in the first 10 iterations. After that, the values of objective functions decrease slowly. Finally, the model converges around 50 iterations.

Tables 4 and 5 list the experimental results on MAE and RMSE, respectively. In addition, Fig. 7 reports the results associated with precision and recall. The selected combinations are shown in Table 6.

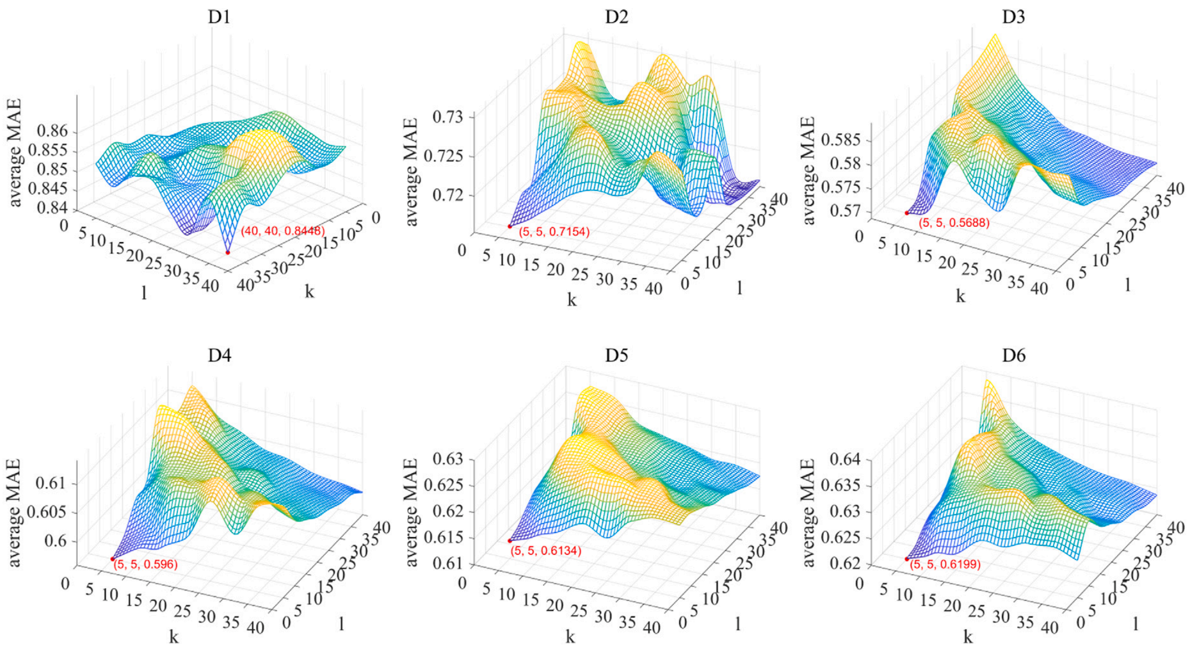

Fig. 4. The average MAE results corresponding to different k and l for IFI algorithm.

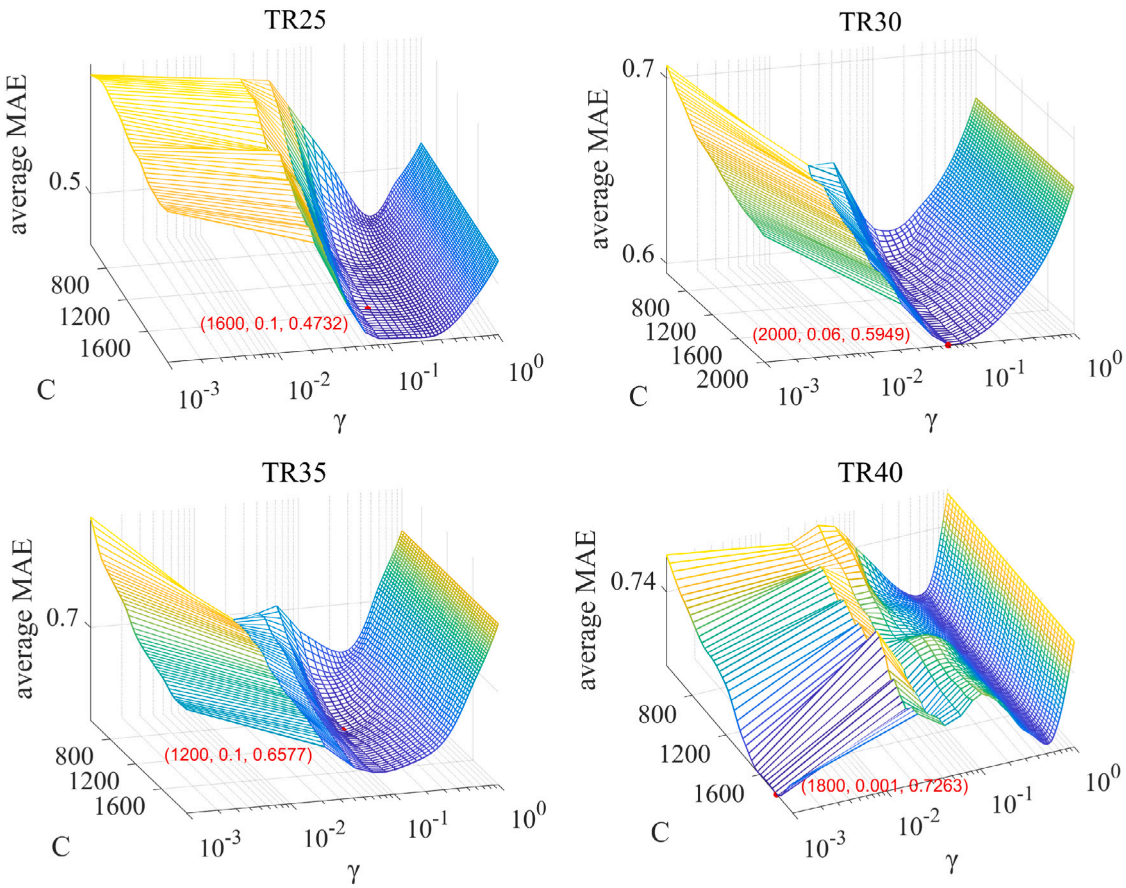

Fig. 5. The average MAE results corresponding to different C and γ for SVM algorithm.

Table 5
The values of RMSE.

Algorithm	TE75	TE70	TE65	TE60
Funk-SVD	1.597	1.528	1.461	1.397
CMF	1.315	1.292	1.238	1.222
CDTF	1.294	1.271	1.216	1.205
CTSIF	1.241	1.221	1.203	1.191
TSEUIF	1.188	1.125	1.113	1.102
ALL	1.081	1.042	1.015	1.021
TSSEA	1.052	1.028	0.998	1.001
TSSEAE	1.069	1.034	1.007	1.013

Table 6
The selected combinations and running time.

Dataset	Algorithm	1	2	3	4	5	6	7	8	9	Running time (s)
TE75	TSSEA				✓	✓	✓				33.93
	TSSEAE					✓	✓				22.45
	ALL	✓	✓	✓	✓	✓	✓	✓	✓	✓	115.54
TE70	TSSEA					✓	✓				23.38
	TSSEAE					✓					11.99
	ALL	✓	✓	✓	✓	✓	✓	✓	✓	✓	118.09
TE65	TSSEA			✓		✓	✓	✓			54.18
	TSSEAE			✓			✓				30.95
	ALL	✓	✓	✓	✓	✓	✓	✓	✓	✓	126.02
TE60	TSSEA	✓	✓								42.08
	TSSEAE	✓									20.97
	ALL	✓	✓	✓	✓	✓	✓	✓	✓	✓	141.02

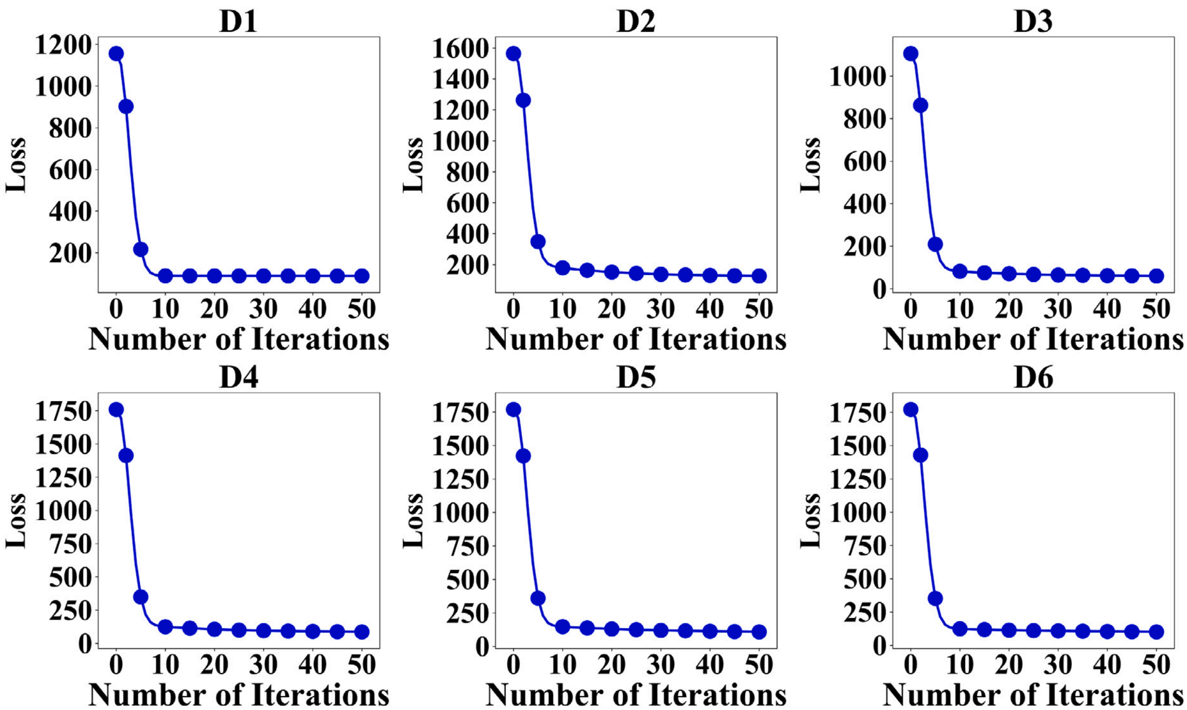

Fig. 6. The iteration processes of IFI algorithm corresponding to 6 auxiliary domains.

From Tables 4–6 and Fig. 7, we can find that the following observations.

(1) All the CDCF algorithms perform better than Funk-SVD, because Funk-SVD is a single-domain CF algorithm which has difficulties in dealing with the sparsity problem.

(2) The four two-sided cross-domain recommendation algorithms perform much better than CMF and CDTF, due to taking advantage of two-sided auxiliary information.

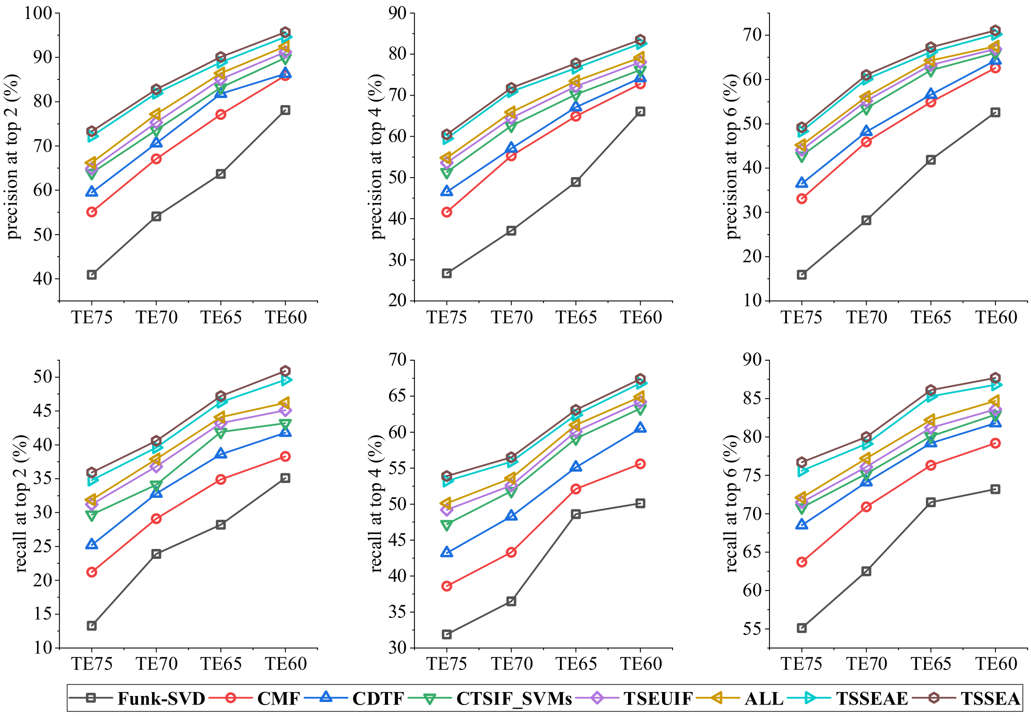

Fig. 7. The results associated with precision and recall.

(3) TSSEA has good prediction accuracies than ALL, which indicates that utilizing a subset from auxiliary domains can achieve better performance than all.

(4) Though TSSEA performs a little better than TSSEAE, it needs to use more auxiliary domains and thus its efficiency is much lower than TSSEAE.

Therefore, the proposed TSSEAE can achieve not only a good recommendation accuracy but also a high efficiency.

5. Conclusion

This paper presents a **T**wo-**S**ided CDCF model based on **S**elective **E**nsemble learning considering both **A**ccuracy and **E**fficiency (TSSEAE). We first convert the original problem of two-sided CDCF into that of selective ensemble learning, which is further formulated as a bi-objective optimization problem. Then, Pareto Ensemble Pruning is employed to determine which combinations of user- and item-auxiliary domains should be selected. To the best of our knowledge, this is the first study to consider both recommendation accuracy and efficiency. The experimental results show that TSSEAE significantly outperforms all the state-of-the-art algorithms at various experimental settings. Also, our study reveals that it would be better to select important combinations than utilize all the combinations. Compared with TSSEA, which considering only recommendation accuracy, TSSEAE can match its recommendation accuracy with a high running efficiency.

In this paper, we only explore the values of ratings in each domain. In our future work, we will integrate user's reviews to further improve the recommendation performance. Moreover, other evaluation indicators, such as diversity and serendipity, will be adopted to fully test the recommendation performance.

CRediT authorship contribution statement

Xu Yu: Conceptualization, Methodology, Writing. **Qinglong Peng:** Experiment, Software. **Lingwei Xu:** Validation, Software. **Feng Jiang:** Validation, Investigation. **Junwei Du:** Visualization, Investigation. **Dunwei Gong:** Investigation, Reviewing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is jointly sponsored by National Natural Science Foundation of China (Nos. 61402246, 61773384), Natural Science Foundation of Shandong Province (Nos. ZR2019MF014, ZR2020QF003, ZR2018MF007), Doctoral Found of QUST, China (Nos. 1203043003480, 12030430010806). All the authors listed have approved the manuscript that is enclosed.

References

Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, *40*(3), 66–72.
Berkovsky, S., Kuflik, T., & Ricci, F. (2007). Cross-domain mediation in collaborative filtering. In *The 11th international conference on user modeling* (pp. 355–359).
Ding, C., Li, T., Peng, W., & Park, H. (2006). Orthogonal nonnegative matrix t-factorizations for clustering. In *The 12th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 126–135).
Fu, M., Qu, H., Yi, Z., Lu, L., & Liu, Y. (2018). A novel deep learning-based collaborative filtering model for recommendation system. *IEEE Transactions on Cybernetics*, *49*(3), 1084–1096.
Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(10), 993–1001.
Hu, L., Cao, J., Xu, G., Cao, L., Gu, Z., & Zhu, C. (2013). Personalized recommendation via cross-domain triadic factorization. In *The 22nd international conference on world wide web* (pp. 595–606).
Kohavi, R., et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *The international joint conference on artificial intelligence* (pp. 1137–1145).
Li, B., Yang, Q., & Xue, X. (2009a). Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *The twenty-first international joint conference on artificial intelligence* (pp. 2052–2057).
Li, B., Yang, Q., & Xue, X. (2009b). Transfer learning for collaborative filtering via a rating-matrix generative model. In *The 26th annual international conference on machine learning* (pp. 617–624).
Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, *21*(2), 498–516.
Liu, L., Feng, J., Pei, Q., Chen, C., Ming, Y., Shang, B., et al. (2020). Blockchain-enabled secure data sharing scheme in mobile edge computing: An asynchronous advantage actor-critic learning approach. *IEEE Internet of Things Journal*, http://dx.doi.org/10.1109/JIOT.2020.3048345.
Liu, Z., Feng, X., Wang, Y., & Zuo, W. (2021). Self-paced learning enhanced neural matrix factorization for noise-aware recommendation. *Knowledge-Based Systems*, http://dx.doi.org/10.1016/j.knosys.2020.106660.
Liu, G., Liu, Y., Zheng, K., Liu, A., Li, Z., Wang, Y., et al. (2018). Mcs-gpm: Multi-constrained simulation based graph pattern matching in contextual social graphs. *IEEE Transactions on Knowledge and Data Engineering*, *30*(6), 1050–1064.
Liu, G., Zheng, K., Wang, Y., Orgun, M. A., Liu, A., Zhao, L., et al. (2015). Multi-constrained graph pattern matching in large-scale contextual social graphs. In *The international conference on data engineering*. http://dx.doi.org/10.1109/ICDE.2015.7113297.
Loni, B., Shi, Y., Larson, M., & Hanjalic, A. (2014). Cross-domain collaborative filtering with factorization machines. In *the European Conference on Information Retrieval* (pp. 656–661).
Mollah, M. B., Zhao, J., Niyato, D., Guan, Y., Yuen, C., Sun, S., et al. (2021). Blockchain for the internet of vehicles towards intelligent transportation systems: A survey. *IEEE Internet Things*, *8*(6), 4157–4185.
Oppermann, M., Kincaid, R., & Munzner, T. (2021). Vizcommender: Computing text-based similarity in visualization repositories for content-based recommendations. *IEEE Transactions on Visualization and Computer Graphics*, *27*(2), 495–505.
Pan, W., Xiang, E., Liu, N., & Yang, Q. (2010). Transfer learning in collaborative filtering for sparsity reduction. In *The AAAI conference on artificial intelligence* (pp. 230–235).
Qian, C., Yu, Y., & Zhou, Z.-H. (2015). Pareto ensemble pruning. In *The AAAI conference on artificial intelligence* (pp. 2935–2941).
Rendle, S. (2010). Factorization machines. In *The 2010 IEEE international conference on data mining* (pp. 14–17).
Singh, A. P., & Gordon, G. J. (2008). Relational learning via collective matrix factorization. In *The 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 650–658).
Van Den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In *The neural information processing systems conference* (pp. 2643–2651).
Xi, L., Wang, R., Yao, Z., & Zhang, F. (2021). Multi-source neighborhood immune detector adaptive model for anomaly detection. *IEEE Transactions on Evolutionary Computation*, *25*(3), 582–594.
Xu, L., Yu, X., & Gulliver, T. A. (2021). Intelligent outage probability prediction for mobile iot networks based on an igwo-elman neural network. *IEEE Transactions on Vehicular Technology*, http://dx.doi.org/10.1109/TVT.2021.3051966.
Xu, L., Zhou, X., Tao, Y., Liu, L., Yu, X., & Kumar, N. (2021). Intelligent security performance prediction for iot-enabled healthcare networks using improved cnn. *IEEE Transactions on Industrial Informatics*, http://dx.doi.org/10.1109/TII.2021.3082907.
Yu, X., Chu, Y., Jiang, F., Guo, Y., & Gong, D. (2018). Svms classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features. *Knowledge-Based Systems*, *141*, 80–91.
Yu, X., Hu, Q., Li, H., Du, J., Gao, J., & Sun, L. (2021). Cross-domain recommendation based on latent factor alignment. *Neural Computing and Applications*, http://dx.doi.org/10.1007/s00521-021-05737-w.
Yu, X., Jiang, F., Du, J., & Gong, D. (2019). A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains. *Pattern Recognition*, *94*, 96–109.
Yu, M., Quan, T., Peng, Q., Yu, X., & Liu, L. (2021). A model-based collaborate filtering algorithm based on stacked autoencoder. *Neural Computing and Applications*, http://dx.doi.org/10.1007/s00521-021-05933-8.
Yu, Y., Yao, X., & Zhou, Z.-H. (2012). On the approximation ability of evolutionary optimization with application to minimum set cover. *Artificial Intelligence*, *180*(181), 20–33.
Yu, X., Zhan, D., Liu, L., Lv, H., Xu, L., & Du, J. (2021). A privacy-preserving cross-domain healthcare wearables recommendation algorithm based on domain-dependent and domain-independent feature fusion. *IEEE Journal of Biomedical and Health Informatics*, http://dx.doi.org/10.1109/JBHI.2021.3069629.
Zhang, Q., Wu, D., Lu, J., Liu, F., & Zhang, G. (2017). A cross-domain recommender system with consistent information transfer. *Decision Support Systems*, *104*, 49–63.
Zhou, Z.-H., Wu, J., & Tang, W. (2002). Ensembling neural networks: many could be better than all. *Artificial Intelligence*, *137*(1–2), 239–263.