Recommendation system: Overlap user Augmentation

Yuwen Lai
National Yang Ming
Chiao Tung University
Email: yuwen.mg11@nycu.edu.tw

Tzu an Cheng
National Yang Ming
Chiao Tung University
Email: cjorm69@gmail.com

Abstract

Recommendation System (RS), as a powerful information filtering tool, has been used for finding the best option from massive possible ones according to users' preference on items. Unfortunately, traditional RS are generally faced with data sparsity and cold start problems. As a promising solution to tackle these issues, cross-domain recommendation system (CDRS) has gained increasing attention recently. Using knowledge transfer to communicate useful information between multiple auxiliary domains, cross-domain recommendation effectively deals with data sparsity on single domain and improves the accuracy in target domain. In this paper, we aim to propose a novel framework, which makes the model keep the advantages of CDRS without facing the problem of data sparsity. To do this, we introduce the knowledge distillation architecture and data augmentation method to the model.The main contributions of this paper are Using DeepCF model to find potentially useful embeddings in advance, and use these embeddings to do data augmentation to increase the training data set and alleviate the data sparsity problem.

1. Introduction

In recent decades, Recommendation System (RS), as a powerful information filtering tool, has been used for finding the best option from massive possible ones according to users' preference on items. In general, recommendation systems are classified into the following two categories based on the forms of recommendations: Collaborative recommendations and Content-based recommendations. The former type recommends the target user by searching for those who have similar preference with target user, and the latter type makes users recommended based on users' historical record(mostly applied on websites or streaming services like Google and Netflix). Unfortunately, both ways are generally faced with data sparsity and cold start problems. As a promising solution to tackle these issues, cross-domain recommendation (CDR) has gained increasing attention recently. Using knowledge transfer to communicate useful information between multiple auxiliary domains, cross-domain recommendation effectively deals with data sparsity on single domain and improves the accuracy in target domain.

In this paper, we aim to propose a novel framework, which makes the model keep the advantages of CDRS without facing the problem of data sparsity. To do this, we introduce the knowledge distillation architecture and data augmentation method to the model. Knowledge distillation is a process of transferring knowledge from a large model to a smaller one. Small models(students) learn and use the knowledge from large ones(teachers) . Recently, knowledge distillation has achieved outstanding results in the computer vision field, there are some studies that apply KD to RS as well: 1)Maintain performance while reducing model size [1] [2] , and 2)solving the bias problems [3].

Having discussed how we want to improve CDRS, now we propose the more concrete framework as follows. First, using the DeepCF model as the teacher model, we can obtain the pretrained user and item embeddings in the two domains (taking two domains as an example), so that the initial input of the model has already been trained once. Second, we use the user embeddings obtained from the DeepCF model to train a Neural Network. The function of the Neural Network is to determine whether the user input from the two domains is the same. Then, we apply the idea of data augmentation. Based on one user domain, we match up with the most similar user from another domain and view this pair as augmented overlapping user data. Last, using the BiTGCF model as the student model, we input the pretrain data and augmented data we got previously to the BiTGCF model. Current research on the SOTA paper(BiTGCF) is to perform recommendation tasks more accurately by using feature propagation and feature transfer to transfer user embeddings between domains. In this paper, moreover, we improve the model by replacing the original input with the pretrained and augmented input. By doing this, we hope to increase the accuracy of forecasting user's behavior patterns.

The main contributions of this paper are summarized as follows:

- We use DeepCF model to find potentially useful embeddings in advance, and use these embeddings to do data augmentation to increase the training data set and alleviate the data sparsity problem.
- The initial embeddings of BiTGCF model are chosen

randomly, and the pretrain method is proposed to improve the initial input quality and make it valuable.

2. Related work

2.1. cross-domain recommendation

Cross domain recommendation(CDR) is an effective technique to deal with alleviating data sparsity issues by using auxiliary information in other domains. Existing CDR methods can be classified into two groups [4]:(1) content-based transfer approaches, and (2) embedding-based transfer approaches.

Content-based transfer approaches are mainly used in the case of dual/multiple domains with same content or metadata features, and tend to utilize identical content information to connect different domains. Berkovsky et al. [5] targeting the data sparsity problem of the ratings in the users-items interaction matrix by applying cross-domain mediation of collaborative user models. [6] Later on, Wang et al. [7] propose a TagCDCTR model, which utilizes common tags as bridges to exploit the relations between domains through an extended collaborative topic modeling framework.

Embedding-based transfer approaches are mainly used in the case of common users with different levels of items, or common items with different users in the dual/multiple domains. Different from content-based transfer approaches, embedding-based transfer approaches tend to apply machine learning techniques. This group of approaches first acquire user and item embedding through different CF-based models, these embeddings are then transferred through common or similar users/items across domains. By using multi-task learning strategy, 1.Multi-task learning (2008 start) Singh et al. [8] utilize a collective matrix factorization model to simultaneously consider several relation matrix(e.g. user x movie ratings relation matrix), and sharing parameters between factors while an entity takes part in multiple relations.

2.Transfer learning(2009 start) Then, by using transfer learning technique, Pan et al. [9] exploits coordinate system transfer(CST) to integrate both user and item knowledge in the auxiliary data matrix, and transfer them through a principled matrix-based transfer learning framework to the target domain that considers the data heterogeneity. Zhang et al. [10] proposed a cross-domain recommender system based on kernel-induced knowledge transfer(KerKT) to adjust the feature spaces of overlapping entities, and knowledge can effectively transfer through overlapping entities between domains.

3.DNN(2017 start) Nowadays, deep learning techniques have revivaled, many deep learning based models are proposed to support knowledge transfer. Man et al. [11] proposed an Embedding and Mapping framework(EMCDR), which used a multilayer perceptron to capture the nonlinear mapping function between domains. Later on, Zhu et al. [12] proposed a deep framework(DCDCSR), utilizing the Matrix Factorization(MF) models to generate user and item latent factors and then employing the Deep Neural Network(DNN) to map the latent factors across domains or systems.

The above CDR approaches are mostly used in the single-target domain, but it also can extend to dual-target or multi-target CDR to improve the performance of recommendation systems.

2.2. Knowledge Distillation

Knowledge Distillation (KD) refers to the process of transferring knowledge from cumbersome large model to lightly small one. Fundamentally, KD is a form of model compression, the similar idea of Model Compression [**?**]was first proposed by Bucila.Deep neural network models face the challenge of limited memory and computational capacity. To tackle the problem, the word 'knowledge distillation' and its concept was formally proposed by Hinton [14]. Hinton et al. proposed to pass the Softened probabilities, known as Dark knowledge, from cumbersome teachers to compact students, which enable students to perform as well as teachers.

KD's algorithms classify into a few types, such as multi-teacher KD, cross model KD, graph based KD, adversarial KD, et cetera, distillating knowledge through different frameworks.

Multi-teacher KD methods have recently been proposed to improve the origin idea of Knowledge Distillation which lets students learn from a single teacher and neglects the potential that a student can learn from multiple teachers. (You et al. 2017 [15] ; Chen et al. 2019 [16]; Wu et al. 2019 [17]; Fei Yuan 2021 [18]; Liu et al. 2021 [19]; Fukuda et al.2017 [20]).Specifically, Wu et al. (2019) teach the student with the comprehensive knowledge by integrating multiple teachers' knowledge to improve the accuracy after compression. Fei Yuan et al. (2021) investigate how to assign appropriate weights to different teacher models on various training samples. They are the first to treat teacher models deferentially at instance level in knowledge distillation.

Adversarial KD is used to tackle the problem of teacher model have difficulty learning from the true data distribution perfectly. Adversarial learning received a great deal of attention due to its great success in generative networks. s, i.e., generative adversarial networks or GANs [21]. Specifically, the discriminator tries to tell the fake data produced by the generator from the real one, while the generator tries to fool the discriminator using generated data samples. Adversarial KD Joint GAN and KD ,generating the valuable data for improving the KD performance and overcoming the limitations of unusable and inaccessible. Many adversarial knowledge distillation methods have been proposed recently.(Yu et al. 2021 [22];Chen et al. 2019 [23];Ye et al. 2020 [24]; Zhiqiang Shen et al. 2019 [25])

In the last few years, a variety of knowledge distillation methods have been widely used for model compression in different visual recognition applications. But we also observed that few knowledge distillation methods have been used for dealing with some problems in the Recommendation system [1] [26] [27]. Based on the observation, we also

apply the Adversarial Distillation framework on the cross-domain Recommendation system for the sake of improving the performance of it.

3. Preliminaries

3.1. Deep collaborative filtering(representation learning)

Representation learning-based CF methods try to map users and items into a common representation space. In this case, the higher similarity between a user and an item in that space implies they match better.

First, representation learning starts from distilling data from a database. IDs, historical records and other auxiliary data help to build an initial representation of user u and item i, which are denoted by V_u and V_i respectively.

The matrix Y is taken as input, i.e., user u is represented by the corresponding row y_{u*} in Y and item i is represented by the corresponding column y_{i*} in Y. Then, we apply MLP to learn latent representation for users and items, the representation learning part for users can be defined as:

$$\begin{aligned} a_0 &= W_0^T y_{u*} \\ a_1 &= r(W_1^T a_0 + b_1) \\ &..... \\ p_u &= a_X = r(W_X^T a_{X-1} + b_X) \end{aligned} \tag{1}$$

where W_x, b_x and a_x denote the weight matrix, bias vector and activation for x-th layer's perceptron respectively. r(·) is the activation function and we use ReLU function in this paper. The latent representation q_i for item i is calculated in the same manner. Different from the existing representation learning-based CF methods, the matching function part is defined as:

$$y_{u_i} = \sigma(W_{out}^T(p_u \cdot q_i)) \tag{2}$$

where W_{out} and $\sigma(\cdot)$ denote the weight matrix and the sigmoid function respectively, and y_{u_i} is our final matching score.

3.2. Bi-directional Transfer Graph Collaborative Filtering Networks

As depicted in Figure 2, the proposed model BiTGCF mainly includes three modules: (1) an embedding layer that provide the initialization of user and item embeddings; (2) a feature propagation and transfer module with multiple layers that distill the initial embeddings of user and item by feature propagation in single domain and feature transfer cross domains; (3) a prediction layer which aggregate the refined embeddings from different propagation layers and outputs the affinity score of a user-item pair.

In this module, we feed $e_u^{A(0)}$, $e_i^{A(0)}$, $e_u^{B(0)}$, $e_i^{B(0)}$ through L graph convolution layers to refine the embeddings of users and items. The module consists of feature of both users and items propagation within each domain and feature of only user transfer cross domains. . We leverage the user-item interaction graphs to propagate and transfer embeddings as follows,

$$\begin{aligned} e_i^{A(k+1)} &= f_P^A(e_i^{A(k)}) \\ e_i^{B(k+1)} &= f_P^B(e_i^{B(k)}) \\ e_u^{A(k+1)} &= f_T^A(f_P^A(e_u^{A(k)}), f_P^B(e_u^{B(k)}) \\ e_u^{B(k+1)} &= f_T^B(f_P^A(e_u^{A(k)}), f_P^B(e_u^{B(k)}) \end{aligned} \tag{3}$$

where $e_u^{A(k)}$ and $e_i^{A(k)}$ respectively denote the refined embeddings of u and i after k layers propagation in D_A, $e_u^{B(k)}$ and $e_i^{B(k)}$ respectively denote the refined embeddings of u and i after k layers propagation in D_B, $f_P^A(\cdot)$ and $f_P^B(\cdot)$ respectively denote the feature propagation function in D_A and D_B, $f_T^A(\cdot\ ,\ \cdot)$ and $f_T^B(\cdot\ ,\ \cdot)$ respectively denote the feature transfer function in D_A and D_B.

After feature propagation and transfer, BiTGCF will evaluate the accuracy of recommendation

$$HR, NDCG = F(e_u^{A(k)}, e_i^{A(k)}, e_u^{B(k)}, e_i^{B(k)} | \Theta) \tag{4}$$

We use the commonly used Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) to evaluate the ranking performance. The HR measures whether the test item is present on the top-10 list, and the NDCG measures the ranking quality by assigning higher scores to hits at top ranks.

4. Proposed method

We first introduce the problem formulation of CDRS in Section 4-1, the overall structure of our model in Section 4-2, and then detail the problem formulation of three main modules in Section 4-3, …..

4.1. Problem formulation

Cross-domain recommendation system solves the sparsity problem by transferring rating knowledge from auxiliary domains. Without loss of generality, we assume to work with two domains: A and B. We introduce the following notation:

TABLE 1. SYMBOL TABLE.

D^A, D^B	Domains
R_a, R_b	user-rating matrices
U^A, U^B	sets of users (user's feature matrix)
U_n^A, U_n^B	Non-overlap user in A, B domain
U_o^A, U_o^B	overlap user in A, B domain
U^{AB}	set of users who have ratings in both domains
U_{aug}^{AB}	set of augmented overlap users
I^A, I^B	sets of items (item's feature matrix)
e_u^A, e_u^B	user's feature (embedding)
e_i^A, e_i^B	item"s feature (embedding)

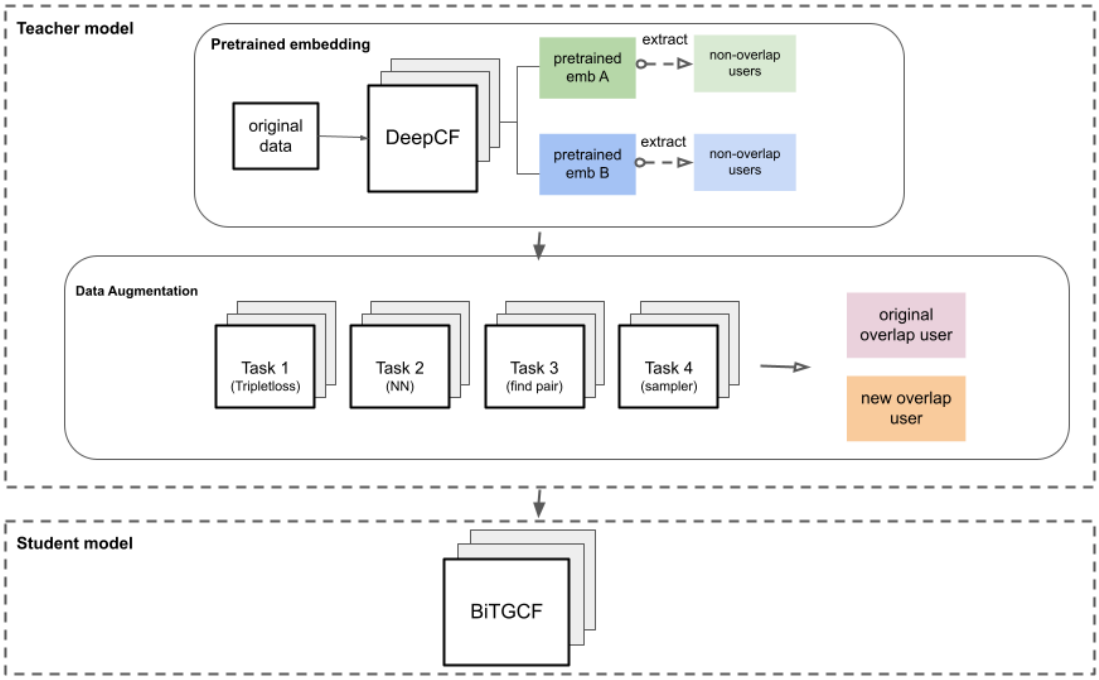

Figure 1. System Overview

Users in each domain are classified into overlap(U_o) and nonoverlap(U_n) users. U_o represent the user purchase items in both domainsI^A, I^B, U_n otherwise. We have the 3-dimensions array composed of UsersID, itemsID, and rating. When rating R R (m,n) states 1 represents users do buy the item, 0 otherwise. Feeding this array into DeepCF model outputs the 64-dimensions embeddings $(e_u^A, e_u^B, e_i^A, e_i^B)$, which is the features of user or item in single domain respectively. Based on these embeddings, we are able to transfer the knowledge including the similarity of buying habits between users, the more similar two users are, the more likely we recommend right on the mark. In the end, BiTGCF will output the scores which represent if we recommend the user precisely and effectively.

4.2. System overview

As depicted in figure 1, the proposed model mainly included three modules.First, we have rating data that represent whether each user purchases any item. Then, By passing the DeepCF model, the features(embedding) of each user and item are outputted by the representation function(network) .

Second, the degree of overlap among domains strongly influences cross-domain recommendations. To deal with the sparsity of common users in CDR, we implement multiple tasks, take task A as an example, we feed the user's embedding through neural networks and obtain the coordinate in the n-dimension. Then we calculate the euclidean distance with the U^A and U^B's coordinate and output the number standing for the preference similarity between users in two domains. Ultimately, we regard $U_i^A (U_i^A \in \mathrm{U}^A)$ and its most similar $U_j^B (U_j^B \in \mathrm{U}^B)$ as the augmented overlap users (U_{aug}^{AB}) and replace all the information of U_j^B with that of U_i^A.

After obtaining augmented U^{AB}, we can then train bothU^{AB} and U_{aug}^{AB} using the BiTGCF model. BiTGCF is mainly composed of three modules, including(1) an embedding layer maps user ID and item ID into embedding vector e^u, e^i, which is used to initialize the user embedding and item embedding in two domains; (2) a feature propagation and transfer module (with multiple layers) is used to refine the initial embedding of user and item through L graph convolution layers; and (3) a prediction layer which concatenates multiple embedding vectors output from different layers, and adopts a dot product to estimate the probability of interaction of user and target item.

4.3. Data Augmentation

We apply four tasks to achieve data augmentation, they are all with the goal of finding new overlap users to solve the problem of data sparsity issue in CDRS while using overlap users as bridge to transfer knowledge.

Task1 and Task2 simply go through the different neural network. Task2 and Task3 using a same neural network to compress the data but the way they choosing pair is different. The former finds all the U^A the most similar U^B as U_{aug}^{AB}, the latter constructs a matching score table of U^A and U^B and choose the top-N users as U_{aug}^{AB}.

TABLE 2. EXPERIMENT RESULT

Dataset	Metrics	BiTGCF	Augmented 10%			Augmented 30%			Augmented 50%		
			Task1	Task2	Task3	Task1	Task2	Task3	Task1	Task2	Task3
elec	HR	0.6036	0.5320	0.4565	0.4830	0.5512	0.5163	0.6382	0.5269	0.5481	0.6554
	NDCG	0.3767	0.3314	0.3003	0.3266	0.3334	0.3243	0.5014	0.3341	0.3416	0.4636
cell	HR	0.6571	0.5563	0.5621	0.5520	0.6338	0.6412	0.6113	0.6742	0.6891	0.6859
	NDCG	0.4210	0.3917	0.3923	0.3927	0.4843	0.5026	0.4225	0.5346	0.5707	0.5710
sport	HR	0.5442	0.4438	0.4611	0.4938	0.4674	0.4788	0.5814	0.5382	0.5397	0.6309
	NDCG	0.3374	0.2915	0.2967	0.3338	0.3031	0.3084	0.4516	0.3821	0.3723	0.4513
cell	HR	0.5654	0.5566	0.5567	0.5578	0.6243	0.6231	0.6265	0.6731	0.6747	0.6083
	NDCG	0.3709	0.4007	0.3859	0.3787	0.4497	0.4761	0.4879	0.5613	0.5613	0.4536

4.3.1. Task 1(Tripletloss).

We input a three-tuple (anchor,positive,negative) each of them is a 64-dimension array, and expect to output a three-tuple transformed into three 4-dimension arrays which store the data's feature relatively.Then, we input (U_o^A, U_o^B, U_n^B) as (anchor, positive, negative) to train the network. By Optimizing Loss(L), We can calculate the preference similarity of U_n^A, U_n^B.

$$L = max(d(a,p) - d(a,n) + margin, 0) \tag{5}$$

where a: anchor; p: positive; n: negative, d(a,p) and d(a,n) are distance between anchor and positive and distance between anchor and negative respectively, margin is a positive constant.

The neural network outputs the relative coordinate of each user in 4-dimension space, which represents the user's preference. Then we calculate euclidean distance to know whether the buying habits of different users in two domains are similar. If U_n^A find the person U_n^B who has the most similar buying preference, we seen the two users as the same and replace all the information of U_n^B with that of U_n^A.

4.3.2. Task 2(Neural Network).

We train a network by using overlap user pairs with the label 1, and non-overlap user pairs with the label 0 as samples, allowing the network to have the ability to distinguish if any two users(i.e. U_i^A and U_j^B) are the same. Specifically,

$$l = f(U_i^A, U_j^B|\Theta) \tag{6}$$

where f is the distinguish function, Θ represents all learnable parameters, and l is the predicted label(between 0 to 1).

Then, using this network to find a person $\hat{u}$ in U_n^B to fit every non-overlap user we sample in U_n^A, and seem them as same user. Finally, change the user id $\hat{u}$ to be the same as D^A. For example, as finding the most similar user in D^B with the user U_{20}^A in D^A , we find a person U_{50}^B, then we change the user id as U_{20}^B.

4.3.3. Task 3.

This task uses the same neural network as task 2, but this task focuses on finding top few non-overlap user pairs as augmented user û. And renumber these user, make sure their user id is in order.

$$\hat{u} = max(f(u_i^A, u_j^B|\Theta)) \tag{7}$$

û is the augmented overlap user pair we find, and other symbols in (7) have the same meaning as (6). After finding augmented overlap user pairs, we also renumbering the user id to make sure they are in order.

5. Experiment

We first explain our experiment setup in section 5-1, and then compare original BiTGCF with pretrained one in section 5-2,

5.1. Experiment setup

5.1.1. Dataset. We evaluate our proposed model on real-world datasets from Amazon dataset, including four datasets(two couple datasets), Electronics (Elec for short) Cell Phones (Cell for short), and Accessories, Sports and Outdoors (Sport for short) & Clothing Shoes and Jewelry (Cloth for short). Furthermore, to prove our proposed model remain a good transfer capability in domains with low similarity, we add Elec & Cloth and Sport & Cell as the third and fourth couple datasets.For the data in these four couple datasets, we first transform them into implicit data, where each entry is marked as 0 or 1, indicating whether the user has rated the item. Then, we filter the datasets to retain users with number of ratings greater than 5 and items with number of ratings greater than 10, and extract the overlapping users in both domains. Table 1 summarizes the detailed statistics of the four couple datasets.

5.2. Performance Comparison

Table 2 shows the summarized result of our experiment on the two-couple datasets of added augmented 10%, 30% and 50% users respectively with HR@10 and NDCG@10. From our experiment, the situation of adding more overlap users get better performance. For example, in sport&cell dataset, when augmented 30% augmented users, task 3 can get better performance then the original BiTGCF. And after adding 50% augmented users, B domain's performance get better than original BiTGCF.

References

[1] J. Tang and K. Wang, "Ranking distillation: Learning compact ranking models with high performance for recommender system," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 2289–2298.

[2] J.-w. Lee, M. Choi, J. Lee, and H. Shim, "Collaborative distillation for top-n recommendation," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 369–378.

[3] D. Liu, P. Cheng, Z. Dong, X. He, W. Pan, and Z. Ming, "A general knowledge distillation framework for counterfactual recommendation via uniform data," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 831–840.

[4] A. K. Sahu and P. Dwivedi, "Knowledge transfer by domain-independent user latent factor for cross-domain recommender systems," *Future Generation Computer Systems*, vol. 108, pp. 320–333, 2020.

[5] S. Berkovsky, T. Kuflik, and F. Ricci, "Cross-domain mediation in collaborative filtering," in *International Conference on User Modeling*. Springer, 2007, pp. 355–359.

[6] P. Winoto and T. Tang, "If you like the devil wears prada the book, will you also enjoy the devil wears prada the movie? a study of cross-domain recommendations," *New Generation Computing*, vol. 26, no. 3, pp. 209–225, 2008.

[7] J. Wang and J. Lv, "Tag-informed collaborative topic modeling for cross domain recommendations," *Knowledge-Based Systems*, vol. 203, p. 106119, 2020.

[8] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 650–658.

[9] W. Pan, E. Xiang, N. Liu, and Q. Yang, "Transfer learning in collaborative filtering for sparsity reduction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 24, no. 1, 2010, pp. 230–235.

[10] Q. Zhang, J. Lu, D. Wu, and G. Zhang, "A cross-domain recommender system with kernel-induced knowledge transfer for overlapping entities," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 7, pp. 1998–2012, 2018.

[11] T. Man, H. Shen, X. Jin, and X. Cheng, "Cross-domain recommendation: An embedding and mapping approach." in *IJCAI*, vol. 17, 2017, pp. 2464–2470.

[12] F. Zhu, Y. Wang, C. Chen, G. Liu, M. Orgun, and J. Wu, "A deep framework for cross-domain and cross-system recommendations," *arXiv preprint arXiv:2009.06215*, 2020.

[13] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 535–541.

[14] G. Hinton, O. Vinyals, J. Dean *et al.*, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[15] S. You, C. Xu, C. Xu, and D. Tao, "Learning from multiple teacher networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1285–1294.

[16] X. Chen, J. Su, and J. Zhang, "A two-teacher framework for knowledge distillation," in *International Symposium on Neural Networks*. Springer, 2019, pp. 58–66.

[17] M.-C. Wu, C.-T. Chiu, and K.-H. Wu, "Multi-teacher knowledge distillation for compressed video action recognition on deep neural networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2202–2206.

[18] F. Yuan, L. Shou, J. Pei, W. Lin, M. Gong, Y. Fu, and D. Jiang, "Reinforced multi-teacher selection for knowledge distillation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 16, 2021, pp. 14 284–14 291.

[19] Y. Liu, W. Zhang, and J. Wang, "Adaptive multi-teacher multi-level knowledge distillation," *Neurocomputing*, vol. 415, pp. 106–113, 2020.

[20] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, "Efficient knowledge distillation from an ensemble of teachers." in *Interspeech*, 2017, pp. 3697–3701.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[22] Y. Jin, Z. Qiu, G. Xie, J. Cai, C. Li, and L. Shen, "Data-free knowledge distillation via adversarial," in *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)*. IEEE, 2021, pp. 447–451.

[23] H. Chen, Y. Wang, C. Xu, Z. Yang, C. Liu, B. Shi, C. Xu, C. Xu, and Q. Tian, "Data-free learning of student networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3514–3522.

[24] J. Ye, Y. Ji, X. Wang, X. Gao, and M. Song, "Data-free knowledge amalgamation via group-stack dual-gan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 516–12 525.

[25] Z. Shen, Z. He, and X. Xue, "Meal: Multi-model ensemble via adversarial learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 4886–4893.

[26] C. Xu, Q. Li, J. Ge, J. Gao, X. Yang, C. Pei, H. Sun, and W. Ou, "Privileged features distillation for e-commerce recommendations," 2019.

[27] Y. Zhang, X. Xu, H. Zhou, and Y. Zhang, "Distilling structured knowledge into embeddings for explainable and accurate recommendation," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 735–743.