

Cross-domain Recommendation via Deep Domain Adaptation ^{*} ^{**}

Heishiro Kanagawa¹ (✉), Hayato Kobayashi^{2,4},
Nobuyuki Shimizu², Yukihiro Tagami², and Taiji Suzuki^{3,4}

¹ Gatsby Unit, UCL, London, United Kingdom
heishiro.kanagawa@gmail.com

² Yahoo Japan Corporation, Tokyo, Japan
{hakobaya, nobushim, yutagami}@yahoo-corp.jp

³ The University of Tokyo, Tokyo, Japan

⁴ RIKEN Center for Advanced Intelligence Project, Tokyo, Japan
taiji@mist.i.u-tokyo.ac.jp

Abstract. The behavior of users in certain services indicates their preferences, which may be used to make recommendations for other services they have never used. However, the cross-domain relation between items and user preferences is not simple, especially when there are few or no common users and items across domains. We propose a content-based cross-domain recommendation method for cold-start users that does not require user- or item-overlap. We formulate recommendations as an extreme classification problem, and the problem is treated as an instance of unsupervised domain adaptation. We assess the performance of the approach in experiments on large datasets collected from Yahoo! JAPAN video and news services and find that it outperforms several baseline methods including a cross-domain collaborative filtering method.

Keywords: Cross-domain recommendation · Deep domain adaptation

1 Introduction

Conventional recommender systems are known to be ineffective in cold-start scenarios, e.g., when the user is new to the service, or the goal is to recommend items from a service that the user has not used, because they require knowledge of the user’s past interactions [32]. On the other hand, as the variety of Web services has increased, information about cold-start users can often be obtained from their activities in other services. Therefore, cross-domain recommender systems (CDRSs), which utilize such information from other domains, have gained research attention as a promising solution to the cold-start problem [4].

This paper addresses a problem of cross-domain recommendation in which we cannot expect users or items to be shared across domains. Our task is to recommend items in a domain (service) to users who have not used it before but have a history in another domain (see Sec. 5 for details). This situation is common in practice. For instance, when one service is not known (e.g., when it is

^{*} This work was conducted during the first author’s intenship at Yahoo! JAPAN.

^{**} TS was partially supported by MEXT Kakenhi (18H03201) and JST-CREST.

newly built) to users of another more popular service, there are few overlapping users, which was actually observed in the dataset of this study. Another probable instance is when user identities are anonymized to preserve privacy [17]; there are no shared users in this case because we cannot know if two users are identical across domains.

The case in which domains have a user- or item-overlap has been extensively studied because an overlap helps us learn relations between users and items of different domains [24, 25, 27, 36]. The problem, however, is harder when there is little or no overlap as learning of user-item relations becomes more challenging. Previous studies dealt with this challenge by using specific forms of auxiliary information that are not necessarily available including user search queries on Bing for recommendations in other Microsoft services [8], tags on items given by users such as in a photo sharing service, e.g., Flickr [1, 2], and external knowledge repositories such as Wikipedia [9, 19, 20, 23]. Methods that use the content information of items like this are called content-based (CB) methods. Collaborative filtering (CF) methods, which do not require auxiliary information, have also been proposed [12, 17, 43]. Despite their broader applicability, a drawback of CF methods compared to CB ones is that they suffer from data sparsity and require a substantial amount of user interactions.

Our main contribution is to fill in the gap between CB and CF approaches. We propose a CB method that (a) only uses content information generally available to content service providers and (b) does not require shared users or items. Our approach is the application of unsupervised domain adaptation (DA). Although unsupervised DA has shown fruitful results in machine learning [3, 11, 33], its connection with CDRSs has been largely unexplored. Our formulation by extreme multi-class classification [6, 30], where labels (items) corresponding to a user are predicted, allows us the use of this technique. We use a neural network (NN) architecture for unsupervised DA, the domain separation network (DSN) [3], because NNs can learn efficient features, as shown in the successful applications to recommender systems [6, 8, 14, 28, 38–41, 44]. We provide a practical case study through experiments on large datasets collected from existing commercial services of Yahoo! JAPAN. We compare our method with several baselines including the state-of-the-art CF method [17]. We show that (a) the CF method does not actually perform well in this real application, and (b) our approach shows the highest performance in DCG.

2 Related Work

Approaches to CDRS can be categorized into two types[4]. The first type aggregates knowledge by combining datasets from multiple domains in a common format, e.g., a common rating matrix [24]. As a result, user-item overlaps or a specific data format are typically assumed [1, 2, 8, 25–27, 35]. In particular, the deep-learning approach [15] uses, as in our method, unstructured texts to learn user and item representations but assumes a relatively large number of common users. The second type links domains by transferring learned knowledge. This

line of research has been limited to matrix-factorization based CF methods because sharing latent factors across domains allows knowledge transfer [12, 17, 43]. Our method is a CB method and solves the aforementioned data sparsity problem of CF methods. Also, DA provides an alternative method of knowledge transfer to shared latent factors.

There are works labeled as cross-domain recommendation that solve a different task. Studies including [7, 22, 29, 42] have dealt with sparsity reduction. The task is to improve recommendation quality within a target domain and is not cross-selling from a different domain. This is performed, for example, by conducting CF on a single domain having sparse data with the help of other domains, such as using a shared latent factor learned in dense domains [7, 22]. Although the work [42] uses DA, it cannot solve the same problem as ours.

Finally, online reinforcement learning can be used to obtain feedback from cold-start users [31]. However, it is difficult to deploy it in commercial applications because the performance is harder to evaluate compared to offline methods such as ours since the model constantly changes over time.

3 Problem Setting

Let X be the input feature space, such as \mathbb{R}^d with a positive integer d . Let $Y = \{1, \dots, L\}$ be a collection of items that we wish to recommend. We are given two datasets from two different domains. The first is a labeled dataset $\mathcal{D}_S = \{(\mathbf{x}_i^S, y_i^S)\}_{i=1}^{N_S}$. Each element denotes a user-item pair with $y_i^S \in Y$ being a label representing the item, and $\mathbf{x}_i^S \in X$ being a feature vector that represents the previous history of a user and is formed by the content information of the items in the history. This domain, from which the labeled dataset comes, is called the *source domain*. The second dataset⁵ is $\mathcal{D}_T^X = \{\mathbf{x}_i^T\}_{i=1}^{N_T}$, where $\mathbf{x}_i^T \in X$ denotes the feature vector of a user’s history consisting of items in the other domain. We call this data domain the *target domain*. The goal is to recommend items in the source domain to users in the target domain. Our approach is to construct a classifier $\eta : X \rightarrow Y$ such that $\eta(\mathbf{x}^T)$ gives the most probable item in Y for a new user history \mathbf{x}^T from the target domain.

4 Unsupervised Domain Adaptation

We define the problem of unsupervised DA [3, 11]. We are given two datasets $\mathcal{D}_S \stackrel{\text{i.i.d.}}{\sim} P_S$ and $\mathcal{D}_T^X \stackrel{\text{i.i.d.}}{\sim} P_T^X$ as in the previous section. Here, P_S and P_T are probability distributions over $X \times Y$, which are assumed to be similar but different. P_T^X is the marginal distribution over X in domain T . The goal of unsupervised DA is to obtain a classifier η with a low risk defined by $R_T(\eta) = \Pr_{(\mathbf{x}, y) \sim P_T}(\eta(\mathbf{x}) \neq y)$ only using the datasets \mathcal{D}_S and \mathcal{D}_T^X .

⁵ Superscript X denotes that the data is missing labels associated to their input vectors.

Domain Separation Networks: We introduce the neural network architecture, domain separation networks (DSNs) [3]. DSN, separately from domain-specific features, learns predictive domain-invariant features, which gives a shared representation of inputs from different domains. Figure 1 shows the architecture of a DSN model, which consists of the following components: a shared encoder $E_c(\mathbf{x}; \theta_c)$, private encoders $E_p^S(\mathbf{x}; \theta_p^S)$, $E_p^T(\mathbf{x}; \theta_p^T)$, a shared decoder $D(\mathbf{h}; \theta_d)$, and a classifier $G(\mathbf{h}; \theta_g)$, where θ_c , θ_p^S , θ_p^T , θ_d , and θ_g denote parameters. Given an input vector \mathbf{x} , which comes from the source domain or the target domain, a shared encoder function E_c maps it to a hidden representation \mathbf{h}_c , which represents features shared across domains. For the source (target) domain, a DSN has a private encoder E_p^S (E_p^T) that maps an input vector to a hidden representation \mathbf{h}_p^S (\mathbf{h}_p^T), which serves as a feature vector specific to the domain. A common decoder D reconstructs an input vector \mathbf{x} of the source (or target) domain from the sum of shared and private hidden representations: \mathbf{h}_c and \mathbf{h}_p^S (\mathbf{h}_p^T). The classifier G takes a shared hidden representation \mathbf{h}_c as input and predicts the corresponding label.

Training is performed by minimizing the following objective function L_{DSN} with respect to parameters θ_c , θ_p^S , θ_p^T , θ_d , and θ_g : $L_{\text{DSN}} = L_{\text{task}} + \alpha L_{\text{recon}} + \beta L_{\text{diff}} + \gamma L_{\text{sim}}$, where α , β , and γ are parameters

that control the effect of the associated terms. L_{task} is a classification loss (cross entropy loss). L_{recon} is a reconstruction loss (squared Euclidean distance). The loss L_{diff} encourages the shared and private encoders to extract different types of features. As in [3], we impose a soft subspace orthogonality loss. The similarity loss L_{sim} encourages the shared encoder to produce representations that are hardly distinguishable. We use the domain adversarial similarity loss [10, 11].

For the output layer of the classifier, we use the softmax activation. We suggest a list of items in descending order of the output probabilities.

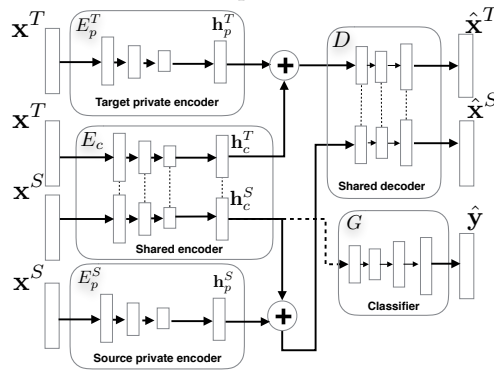


Fig. 1: Architecture of DSN

5 Experiments

We conduct performance evaluation on a pair of two real-world datasets in collaboration with Yahoo! JAPAN. The datasets consist of the browsing logs of a video on demand service (VIDEO) and a news aggregator (NEWS). The task is to recommend videos to NEWS users who have not used the VIDEO service.

Dataset Description: For VIDEO, we create a labeled dataset where a label is a video watched by a particular user, and the input is the list of videos previously watched by the user. Similarly, for NEWS, we build an unlabeled dataset that consists of histories of news articles representing users. The number of instances

for each dataset is roughly 11 million. For evaluation, we form a test dataset of size 38,250 from users who used both services, where each instance is a pair of a news history (input) and a video (label).

Items in both services have text attributes. For VIDEO, we use the following attributes: title, category, short description, and cast information. For NEWS, article title and category are used. We treat a history as a document comprised of attribute words and represent it with the TF-IDF scheme. We form a vocabulary set for each dataset according to the TF-IDF value. Combining the two sets gives a common vocabulary set of 50,000 words.

Experimental Settings: We construct a DSN consisting of fully-connected layers⁶. The exponential linear unit (ELU) [5] is used as the activation function for all layers. The weight and bias parameters for the fully-connected layers are initialized following [13]. We apply dropout [34] to all fully-connected layers with the rates 0.25 for all the encoders and 0.5 for the decoder and the classifier. We also impose an L2 penalty on the weight parameters with the regularization parameter chosen from $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. The parameters α , β , and γ in L_{DSN} are set to 10^{-3} , 10^{-2} , and 100, respectively. The parameter γ is chosen by the similarity of the marginals⁷ and agrees with the setting in [3]. The ADAM optimizer [21] is used, and the initial learning rate is set to 10^{-3} .

We use discounted cumulative gain (DCG) [18] for evaluation. DCG gives a higher score when a correct item appears earlier in a suggested item list and thus measures ranking quality. This is defined by $\text{DCG}@M = \sum_{m=1}^M I(\hat{y}_m = y) / \log(m + 1)$, where the function $I(\hat{y}_m = y)$ returns 1 when the m -th suggested item \hat{y}_m is y and otherwise 0. The use of metrics such as precision is not appropriate since negative feedback is not well-defined in implicit feedback; unobserved items may express users' distaste or just have been unseen, whereas we can treat viewed items as true positive.

We evaluate the method for five different training and test dataset pairs, which are randomly sampled from the datasets. We sample 80% of both the whole training and test datasets. For each pair, we compute DCG averaged over the test set. We use a validation set that consists of logs of common users on the same dates as the training data for model selection to investigate attainable performance of domain adaptation. Note that this setting still keeps our target situation where there are not enough overlapping users for training.

Baseline Models: The following baseline methods are compared.

- **Most Popular Items (POP).** This suggests the most popular items in the training data. The comparison with this shows how well the proposed approach achieves personalization.
- **Cross-domain Matrix Factorization (CdMF) [17].** This is the state-of-the-art collaborative filtering method for CDRSs requiring no user- or item-overlap. To alleviate sparsity, we eliminate users with history logs fewer than

⁶ The unit size of each hidden layer is as follows: $E_c = E_p^S = E_p^T = (256 - 128 - 128 - 64)$, $D = (128 - 128 - 256)$, and $G = (256 - 256 - 256 - 64)$. Left is the input.

⁷ We train a classifier that detects the domain of the input represented by the shared encoder and test the classification accuracy. This is due to the adversarial training.

Table 1: Performance comparison of DSN, NN, CdMF, POP. (DCG) of DSN and NN denotes models that are chosen according to the value of DCG@100. Similarly, (CEL) denotes the cross entropy loss. Each entry is Mean \pm Std.

Method	DCG@1	DCG@10	DCG@50	DCG@100
DSN (DCG)	0.0618 \pm 0.0212	0.2133 \pm 0.0154	0.2873 \pm 0.0151	0.2945 \pm 0.0153
DSN (CEL)	0.0406 \pm 0.0212	0.1668 \pm 0.0384	0.2583 \pm 0.0230	0.2655 \pm 0.0229
NN (DCG)	0.0415 \pm 0.0211	0.1938 \pm 0.0131	0.2735 \pm 0.0102	0.2797 \pm 0.0107
NN (CEL)	0.0282 \pm 0.0301	0.1616 \pm 0.0279	0.2473 \pm 0.0247	0.2556 \pm 0.0238
CdMF	0.0005 \pm 0.0000	0.0040 \pm 0.0000	0.0135 \pm 0.0000	0.0644 \pm 0.0004
POP	0.0398 \pm 0.0007	0.2099 \pm 0.0012	0.2790 \pm 0.0016	0.2871 \pm 0.0010

5 for the video data and 20 for the news data from the training data. We construct a user-item matrix for each domain, with the value of observed entries 1 and of unobserved entries 0. The unobserved entries are randomly subsampled as the size of the whole unseen entries is too large to be processed. As with our method, we use 80 percent of the instances for training and the remaining 20 percent for validation. The hyper-parameters α , β_0 , and ν_0 are set at the same values as in [17]. The method requires inference of latent variables by Gibbs sampling; each sampling step involves computing matrix inverses. Due to the computational complexity, we were unable to optimize these hyper-parameters, and therefore they are fixed.

- **Neural Network (NN).** To investigate the effect of DA, the same network without DA is evaluated. This is obtained by minimizing the same loss as DSN except that β , γ in L_{DSN} are set to 0. It can be thought of as a strong content-based single-domain method as more robust features are learned with the reconstruction loss L_{recon} than only with the task loss L_{task} (as in [39], with the roles of item and user swapped).

Experimental Results: We report the results in Table 1. CdMF underperformed other methods. As mentioned in [16], this is likely because CdMF cannot process the implicit feedback or accurately capture the popularity structure in the dataset. The DSN model chosen by the cross-entropy loss (DSN (CEL)) also had a worse performance than POP. We also chose the model based on the values of DCG@100 on the validation set (denoted by DSN (DCG)), which showed the best performance in all cut-off number settings. This result can be interpreted as follows. As predicting the most probable item is hard, the cross-entropy loss does not give a useful signal for model selection. On the other hand, DCG provides the quality of ranking, and therefore a classifier which captures the joint distribution of items and users well is more likely to be chosen. The improved performance of DSN (DCG) over NN (DCG) supports this claim since DA improves the learning of the joint distribution. This implies that replacing the loss with a ranking loss such as [37] could improve ranking quality.

Conclusion and Future Work: Our evaluation demonstrated the potential of the proposed DA-based approach. However, extreme classification is still challenging and should be addressed as a future work. One possible direction would be incorporation of item information as in [39], as this would make items more distinguishable and enable predictions of unobserved items.

References

1. Abel, F., Araújo, S., Gao, Q., Houben, G.J.: Analyzing Cross-System User Modeling on the Social Web, pp. 28–43. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22233-7_3
2. Abel, F., Herder, E., Houben, G.J., Henze, N., Krause, D.: Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction* **23**(2), 169–209 (2013). <https://doi.org/10.1007/s11257-012-9131-2>
3. Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., Erhan, D.: Domain separation networks. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) *Advances in Neural Information Processing Systems 29*, pp. 343–351. Curran Associates, Inc. (2016), <http://papers.nips.cc/paper/6254-domain-separation-networks.pdf>
4. Cantador, I., Fernández-Tobías, I., Berkovsky, S., Cremonesi, P.: Cross-Domain Recommender Systems, pp. 919–959. Springer US, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7637-6_27
5. Clevert, D., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs). *CoRR* **abs/1511.07289** (2015), <http://arxiv.org/abs/1511.07289>
6. Covington, P., Adams, J., Sargin, E.: Deep neural networks for youtube recommendations. In: *Proceedings of the 10th ACM Conference on Recommender Systems*. pp. 191–198. RecSys '16, ACM, New York, NY, USA (2016). <https://doi.org/10.1145/2959100.2959190>
7. Cremonesi, P., Quadrana, M.: Cross-domain recommendations without overlapping data: Myth or reality? In: *Proceedings of the 8th ACM Conference on Recommender Systems*. pp. 297–300. RecSys '14, ACM, New York, NY, USA (2014). <https://doi.org/10.1145/2645710.2645769>
8. Elkahky, A.M., Song, Y., He, X.: A multi-view deep learning approach for cross domain user modeling in recommendation systems. In: *Proceedings of the 24th International Conference on World Wide Web*. pp. 278–288. WWW '15, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2015). <https://doi.org/10.1145/2736277.2741667>
9. Fernández-Tobías, I., Cantador, I., Kaminskis, M., Ricci, F.: A generic semantic-based framework for cross-domain recommendation. In: *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*. pp. 25–32. HetRec '11, ACM, New York, NY, USA (2011). <https://doi.org/10.1145/2039320.2039324>
10. Ganin, Y., Lempitsky, V.S.: Unsupervised domain adaptation by backpropagation. In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. pp. 1180–1189 (2015), <http://jmlr.org/proceedings/papers/v37/ganin15.html>
11. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. *Journal of Machine Learning Research* **17**(59), 1–35 (2016), <http://jmlr.org/papers/v17/15-239.html>
12. Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., Guo, J.: Cross-domain recommendation via cluster-level latent factor model. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) *Machine Learning and Knowledge Discovery in Databases*. pp. 161–176. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)

13. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. CoRR **abs/1502.01852** (2015), <http://arxiv.org/abs/1502.01852>
14. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web. pp. 173–182. WWW '17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2017). <https://doi.org/10.1145/3038912.3052569>
15. Hu, G., Zhang, Y., Yang, Q.: MTNet: A neural approach for cross-domain recommendation with unstructured text. In: KDD18 Deep Learning Day (2018), https://www.kdd.org/kdd2018/files/deep-learning-day/DLDay18_paper_5.pdf
16. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining. pp. 263–272. ICDM '08, IEEE Computer Society, Washington, DC, USA (2008). <https://doi.org/10.1109/ICDM.2008.22>
17. Iwata, T., Takeuchi, K.: Cross-domain recommendation without shared users or items by sharing latent vector distributions. In: Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS). pp. 379–387. AISTATS'15, USA (2015)
18. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.* **20**(4), 422–446 (Oct 2002). <https://doi.org/10.1145/582415.582418>
19. Kaminskas, M., Fernández-Tobías, I., Cantador, I., Ricci, F.: Ontology-Based Identification of Music for Places, pp. 436–447. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36309-2_37
20. Kaminskas, M., Fernández-Tobías, I., Ricci, F., Cantador, I.: Knowledge-based identification of music suited for places of interest. *Information Technology & Tourism* **14**(1), 73–95 (2014). <https://doi.org/10.1007/s40558-014-0004-x>
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014), <http://arxiv.org/abs/1412.6980>
22. Li, B., Yang, Q., Xue, X.: Can movies and books collaborate?: Cross-domain collaborative filtering for sparsity reduction. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence. pp. 2052–2057. IJCAI'09, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2009)
23. Loizou, A.: How to recommend music to film buffs: enabling the provision of recommendations from multiple domains (May 2009), <https://eprints.soton.ac.uk/66281/>
24. Loni, B., Shi, Y., Larson, M., Hanjalic, A.: Cross-domain collaborative filtering with factorization machines. In: Proceedings of the 36th European Conference on IR Research on Advances in Information Retrieval - Volume 8416. pp. 656–661. ECIR 2014, Springer-Verlag, Berlin, Heidelberg (2014). https://doi.org/10.1007/978-3-319-06028-6_72
25. Low, Y., Agarwal, D., Smola, A.J.: Multiple domain user personalization. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 123–131. KDD '11, ACM, New York, NY, USA (2011). <https://doi.org/10.1145/2020408.2020434>
26. Man, T., Shen, H., Jin, X., Cheng, X.: Cross-domain recommendation: An embedding and mapping approach. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. pp. 2464–2470 (2017). <https://doi.org/10.24963/ijcai.2017/343>

27. Nakatsuji, M., Fujiwara, Y., Tanaka, A., Uchiyama, T., Ishida, T.: Recommendations over domain specific user graphs. In: Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence. pp. 607–612. IOS Press, Amsterdam, The Netherlands, The Netherlands (2010)
28. Oord, A.v.d., Dieleman, S., Schrauwen, B.: Deep content-based music recommendation. In: Proceedings of the 26th International Conference on Neural Information Processing Systems. pp. 2643–2651. NIPS’13, Curran Associates Inc., USA (2013)
29. Pan, W., Xiang, E.W., Liu, N.N., Yang, Q.: Transfer learning in collaborative filtering for sparsity reduction. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence. pp. 230–235. AAAI’10, AAAI Press (2010)
30. Partalas, I., Kosmopoulos, A., Baskiotis, N., Artières, T., Paliouras, G., Gaussier, É., Androutsopoulos, I., Amini, M., Gallinari, P.: LSHTC: A benchmark for large-scale text classification. CoRR **abs/1503.08581** (2015), <http://arxiv.org/abs/1503.08581>
31. Rubens, N., Elahi, M., Sugiyama, M., Kaplan, D.: Active Learning in Recommender Systems, pp. 809–846. Springer US, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7637-6_24
32. Sahebi, S., Brusilovsky, P.: Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation. In: Carberry, S., Weibelzahl, S., Micarelli, A., Semeraro, G. (eds.) User Modeling, Adaptation, and Personalization. pp. 289–295. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
33. Sener, O., Song, H.O., Saxena, A., Savarese, S.: Learning transferrable representations for unsupervised domain adaptation. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 2110–2118. Curran Associates, Inc. (2016), <http://papers.nips.cc/paper/6360-learning-transferrable-representations-for-unsupervised-domain-adaptation.pdf>
34. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* **15**, 1929–1958 (2014), <http://jmlr.org/papers/v15/srivastava14a.html>
35. Sun, M., Li, F., Zhang, J.: A multi-modality deep network for cold-start recommendation. *Big Data and Cognitive Computing* **2**(1) (2018). <https://doi.org/10.3390/bdcc2010007>, <http://www.mdpi.com/2504-2289/2/1/7>
36. Tang, J., Wu, S., Sun, J., Su, H.: Cross-domain collaboration recommendation. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1285–1293. KDD ’12, ACM, New York, NY, USA (2012). <https://doi.org/10.1145/2339530.2339730>
37. Valizadegan, H., Jin, R., Zhang, R., Mao, J.: Learning to rank by optimizing ndcg measure. In: Bengio, Y., Schuurmans, D., Lafferty, J.D., Williams, C.K.I., Culotta, A. (eds.) Advances in Neural Information Processing Systems 22, pp. 1883–1891. Curran Associates, Inc. (2009), <http://papers.nips.cc/paper/3758-learning-to-rank-by-optimizing-ndcg-measure.pdf>
38. Wang, H., SHI, X., Yeung, D.Y.: Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 415–423. Curran Associates, Inc. (2016)
39. Wang, H., Wang, N., Yeung, D.Y.: Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD International Conference on

- Knowledge Discovery and Data Mining. pp. 1235–1244. KDD '15, ACM, New York, NY, USA (2015). <https://doi.org/10.1145/2783258.2783273>
40. Wu, C.Y., Ahmed, A., Beutel, A., Smola, A.J., Jing, H.: Recurrent recommender networks. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. pp. 495–503. WSDM '17, ACM, New York, NY, USA (2017). <https://doi.org/10.1145/3018661.3018689>
 41. Wu, Y., DuBois, C., Zheng, A.X., Ester, M.: Collaborative denoising auto-encoders for top-n recommender systems. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. pp. 153–162. WSDM '16, ACM, New York, NY, USA (2016). <https://doi.org/10.1145/2835776.2835837>
 42. Zhang, Q., Wu, D., Lu, J., Liu, F., Zhang, G.: A cross-domain recommender system with consistent information transfer. *Decision Support Systems* **104**, 49 – 63 (2017). <https://doi.org/https://doi.org/10.1016/j.dss.2017.10.002>, <http://www.sciencedirect.com/science/article/pii/S016792361730180X>
 43. Zhang, Y., Cao, B., Yeung, D.Y.: Multi-domain collaborative filtering. In: Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence. pp. 725–732. UAI'10, AUAI Press, Arlington, Virginia, United States (2010)
 44. Zheng, Y., Tang, B., Ding, W., Zhou, H.: A neural autoregressive approach to collaborative filtering. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. pp. 764–773. ICML'16, JMLR.org (2016)