Learning Privacy-aware Transferable Representations for Cross-Domain Recommendation

Anonymous Author(s)

ABSTRACT

Transfer learning is an effective technique to improve a target recommender system with the knowledge from a related auxiliary data source in the cross-domain recommendation. Although the data provider in the source domain only releases part of their data (i.e., public) to the target domain, the party in the target domain can still have a high chance (e.g. compared with a random guess) to recover the remaining part of their data (i.e., private). As a result, the private information in the source domain (e.g., user age & gender) can be eavesdropped by a malicious attacker. In this paper, we aim to learn a privacy-aware transferable representation such that this representation is useful for improving the recommendation performance in the target domain while it is helpful to secure private information in the source domain. The privacy of the transferable representation is measured by the ability of an attacker to recover accurately the private information from it. We adopt the adversarial learning technique to modify the training objective of a vanilla cross-domain model and show that the proposed defense model (PrNet) achieves a good tradeoff between the utility and privacy of the transferable representations.

1 INTRODUCTION

In this paper, we investigate the scenario of attacking the transferred representations during the knowledge transfer from the source domain to the target domain in cross-domain recommender systems. This scenario exists when traditional transfer learning methods have access to the hidden representations transferred from a source domain, or when the neural representations are communicated and shared among several terminal devices [24]. A malicious attacker can eavesdrop these hidden representations and recovers private information of the source domain from them.

Consider a typical example of deep transfer learning for cross-domain recommendation as follows (see Fig. 2). A neural network (source network) is trained from source domain examples and the intermediate hidden representations are transferred as knowledge to the neural network in the target domain (target network). Such representations may contain information about the samples of the source domain since they are trained on the source domain. There is a study [36] showing that user demographics (e.g., age) can be accurately predicted from a user's records such as her online behavior. As a result, an attacker can still have a high chance (e.g. better than a random guess) to recover the private part of the source data, though the data providers only release the public part of their data to the target domain. Even worse, it may be illegal to share the public part of the data under the General Data Protection Regulation (GDPR) [42] or California Consumer Privacy Act (CCPA) [1].

The private information can be *explicit* when sensitive attributes (e.g., user demographics) are used to train models [11, 18] since these demographic features can be used to improve the model performance or to alleviate the data sparsity issue [9, 47]. Under this case, there is a trade-off between the utility of the sensitive information and their privacy since the private information correlates with the label directly and will be easily learned by the model. Sometimes, it may sacrifice a bit performance to protect privacy.

The private information can also be *implicit* where sensitive information (e.g., occupational class) is inferred accurately from the input but it is not present in the input [33, 36]. Under this case, there are still privacy concerns since the model may incidentally learn something about the sensitive information when it predicts the label. For example, the transferred representations may incidentally learn representations which are useful to infer the sensitive information of the source domain.

We present an attack scenario meant at characterizing the privacy of neural representations for cross-domain recommendation and propose a defense model (PrNet) to improve the recommendation performance as well as the privacy of transferred representations. Traditional transfer learning models aim at improving recommendation performance only and have risks on privacy leaks. In this paper, we aim to learn a privacy-aware transferable representation such that this representation is useful for improving recommendation performance in the target domain while it is difficult to be used to recover private information in the source domain.

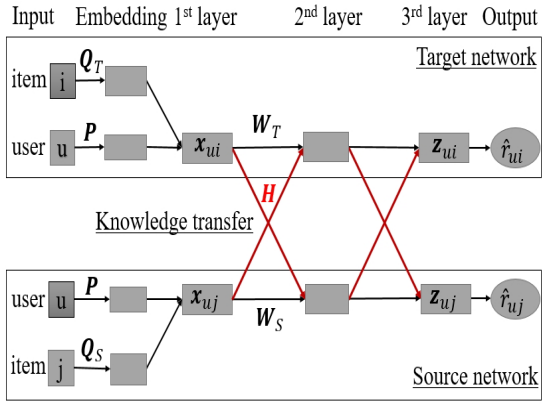
As a running example, we consider a user's gender as the private variable of the source domain to be protected when we transfer the source domain's user-item interactions to the target domain. As a result, we focus on the *implicit* private information protection, that is, the private information is not involved for improving recommendation. Other private information (e.g., age and residence) can also be considered as labels to protect in the same way.

Under the attack scenario, the cross-domain recommender systems can work as follows. The source neural network does not provide its input (public or private) to the target neural network, but only its vector representation. Furthermore, the vector representation is not much useful to recover private information of the source domain from it even if it is eavesdropped by a malicious attacker.

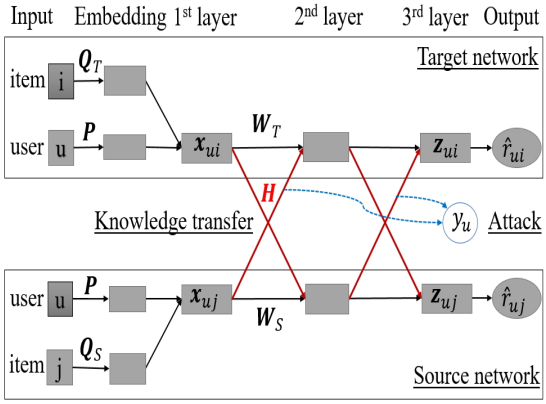
The paper is organized as follows. We firstly introduce the problem scenario in Section 2, and compare traditional transfer learning with the attack during representation transfer. In Section 3, we present a defense model (PrNet) to learn a privacy-aware transferable representation by following a two-agent evaluation methodology. In Section 4, we experimentally show the effectiveness of the defense model on both recommendation performance (Section 4.3) and privacy protection (Section 4.4). We review related works in Section 5 and conclude the whole paper in Section 6.

2 PROBLEM SCENARIO

We firstly describe a typical cross-domain recommendation scenario. Then we point out the possibility of an attacker to eavesdrop the private information during the knowledge transfer between the



(a) Traditional transfer learning



(b) Attack during transfer

Figure 1: Left: Traditional transfer learning for cross-domain recommendation [20] where the red solid arrows labelled with $\boldsymbol{H}$ are to enable knowledge transfer. Right: An attacker eavesdropping representations during knowledge transfer to recover private information where the blue dotted curves labelled with y_u are to attack on representations.

source domain and the target domain. We give the notations before introducing the problem scenario.

2.1 Notation

We have two domains, a source domain $\mathcal{S}$ and a target domain $\mathcal{T}$. The sets of users in two domains are shared, denoted by $\mathcal{U}$ (of size $m = |\mathcal{U}|$) and act as a bridge for knowledge transfer. Denote the sets of items in $\mathcal{S}$ and $\mathcal{T}$ by $\mathcal{I}_S$ and $\mathcal{I}_T$ (of size $n_S = |\mathcal{I}_S|$ and $n_T = |\mathcal{I}_T|$), respectively. For the target domain, a binary matrix $\boldsymbol{R}_T \in \mathbb{R}^{m \times n_T}$ describes the user-item interactions, where the entry $r^T_{ui} \in \{0, 1\}$ equals 1 if user u has an interaction with item i and 0 otherwise. Similarly, for the source domain, a binary matrix $\boldsymbol{R}_S \in \mathbb{R}^{m \times n_S}$ describes the user-item interactions, where the entry $r^S_{uj} \in \{0, 1\}$ equals 1 if user u has an interaction with item j and 0 otherwise.

For item recommendation, we recommend top-K items for each user and they are ranked by their predicted scores:

$$\hat{r}_{ui} \triangleq P(r_{ui}|u, i; \Theta) = f(u, i|\Theta), \tag{1}$$

where f is the interaction function and Θ are model parameters. For neural collaborative filtering approaches, the function f is learned via a neural network:

$$f(u, i|\boldsymbol{P}, \boldsymbol{Q}, \theta_f) = \phi_o(\phi_L(...(\phi_1(\boldsymbol{x}_{ui}))...)), \tag{2}$$

where $\boldsymbol{x}_{ui}$ is the concatenated embeddings of the user and the item projected by embedding matrices $\boldsymbol{P} \in \mathbb{R}^{m \times d}$ and $\boldsymbol{Q} \in \mathbb{R}^{n \times d}$. The output and hidden layers are computed by ϕ_o and $\{\phi_l\}_{l=1}^{L}$ in a multilayer neural network parameterized by θ_f.

2.2 Traditional Transfer Learning

We follow the work of collaborative cross networks (CoNet) [20] to describe the cross-domain recommendation scenario. CoNet is a recently proposed deep transfer learning method for cross-domain recommendation.

CoNet couples two basic neural networks via the cross-connection units as shown in Fig. 1a. Each basic neural network is an implementation for the function described in Eq. (2). The cross-connection units (the solid red arrows labelled with $\boldsymbol{H}$ in Fig. 1a) enable the knowledge transfer between the source network and the target network. In detail, given two activation maps $\boldsymbol{a}_S$ and $\boldsymbol{a}_T$ of some hidden layer ℓ in the two basic networks, CoNet learns a combination of two input activations and feed these combinations as input to the successive layer's filter:

$$\tilde{\boldsymbol{a}}_S = \boldsymbol{W}_S \boldsymbol{a}_S + \boldsymbol{H} \boldsymbol{a}_T, \tag{3a}$$

$$\tilde{\boldsymbol{a}}_T = \boldsymbol{W}_T \boldsymbol{a}_T + \boldsymbol{H} \boldsymbol{a}_S, \tag{3b}$$

where $\boldsymbol{W}_S$ and $\boldsymbol{W}_T$ are connection weight matrices, and $\boldsymbol{H}$ is the transfer matrix which enables the knowledge transfer.

To select representations to transfer, a variant of CoNet imposes an ℓ_1-norm penalty on entries of $\boldsymbol{H}$ to induce its sparsity.

2.3 Attack During Transfer

As shown in Eq. (3b), the party in the target network can access to the representations (i.e., $\boldsymbol{a}_S$) from the source network. These representations may contain private information of the source domain that the source data provider is not willing to reveal; or even worse, it is illegal under GDPR [42]. For example, a previous study [36] shows that user demographics (e.g., age) can be accurately predicted from a user's records such as online behavior.

We consider an attack scenario on the transferred representation as shown in Fig. (1b). An attacker eavesdrops representations during knowledge transfer and recovers the private information of the source domain from it. The private information is denoted by y_u in Fig. (1b). The attacker predicts user u's private information y_u based on her transferred representation $trep(\boldsymbol{a}_S^{(u)})$:

$$\hat{y}_u \triangleq P\Big(y_u|trep(\boldsymbol{a}_S^{(u)}); \theta\Big) = g\Big(trep(\boldsymbol{a}_S^{(u)})|\theta\Big), \tag{4}$$

where g is the privacy prediction function parameterized by θ (e.g., a logistic regression model or a neural network). We will discuss the details of the transferred representation $trep(\cdot)$ and privacy prediction function $g(\cdot)$ in the proposed defense model next.

We have two goals in the attack setting: One is to improve the performance of the target recommender and one is to protect the

privacy of the source data provider. That is, we need to learn a privacy-aware transferable representation.

3 PRNET

We propose a defense model to learn a privacy-aware transferable representation, aimed at achieving a good trade-off between the utility and privacy.

3.1 A Two-agent Evaluation Methodology

We firstly introduce a methodology to evaluate the utility and privacy of a model. We measure the privacy of a transferred representation by the ability of an attacker to recover the private information from it, i.e., to predict accurately the user gender from it. We design a two-agent evaluation methodology. We train a recommender agent and an attacker agent.

3.1.1 Recommender agent. The recommender agent is a cross-domain recommender system, which can be a privacy-agnostic model (*vanilla recommender agent*) such as CoNet, or be a privacy-aware model (adversarial recommender agent) such as the one to be proposed. Following the description in Section 2.2, the recommender agent of CoNet is trained to minimize the negative log-likelihood (the cross-entropy loss) over user-item interaction examples:

$$\mathcal{L}_{vanilla}(\Theta) = \sum_{c_T} -\log P(r^T_{ui}|u,i;\Theta_T) + \sum_{c_S} -\log P(r^S_{uj}|u,j;\Theta_S),$$

where the two indices are $c_T \triangleq (u,i) \in \boldsymbol{R}^+_T \cup \boldsymbol{R}^-_T$ and $c_S \triangleq (u,j) \in \boldsymbol{R}^+_S \cup \boldsymbol{R}^-_S$, respectively. $\boldsymbol{R}^+_*$ and $\boldsymbol{R}^-_*$ with $* \in \{S,T\}$ are the observed interactions and randomly sampled negative examples [30].

3.1.2 Attacker agent. The *attacker agent* is an adversarial discriminator, which learns to recover the private information in the source domain from the transferred representations. Since the data example has a form of the user-item interaction, Eq. (4) is formulated as:

$$\hat{y}_{uj} = P\Big(y_{uj}|trep(\boldsymbol{a}^{(uj)}_S);\theta\Big) = g\Big(trep(\boldsymbol{a}^{(uj)}_S)|\theta\Big), \quad (5)$$

where $y_{uj} \equiv y_u$ since a user's private information is determined by the user itself only. We use a simple logistic regression model to implement the privacy prediction function $g(\boldsymbol{x}) = \sigma(\boldsymbol{w}^T\boldsymbol{x} + b)$. In reality, the malicious attacker can user any machine learning model to implement the privacy prediction function, e.g., a neural network.

For the transferred representation $trep(\boldsymbol{a}^{(uj)}_S)$, since there are typically multiple hidden layers in a neural network (see Fig. 2), we use a simple concatenation strategy to merge the hidden representation $\{\boldsymbol{a}^{(uj)}_S|\ell\}$ from each hidden layer ℓ:

$$trep(\boldsymbol{a}^{(uj)}_S) = \text{concat}(\{\boldsymbol{a}^{(uj)}_S|\ell\}),\ \ell = 1,...,L-1. \quad (6)$$

The attacker agent is trained to minimize the negative log-likelihood (the cross-entropy loss) over user gender labels:

$$\mathcal{L}_{attacker}(\theta) = \sum_{c_S} -\log P\Big(y_{uj}|trep(\boldsymbol{a}^{(uj)}_S);\theta\Big). \quad (7)$$

3.2 PrNet: The Proposed Defense Model

We present a defense model against the attack on representations described in Section 2.3. Since we have two rival objectives (i.e., utility and privacy), we adopt the adversarial learning technique [16] to learn a privacy-aware representation by modifying the training objectives. Following the workflow of the two-agent evaluation methodology described in Section 3.1, we modify the objective of a vanilla recommender agent, i.e., CoNet, and propose an adversarial recommender agent. We dub the privacy-aware neural model as PrNet.

The basic idea is to simulate the attack on representations during training. We integrate an *attacker simulator* into the vanilla recommender agent, leading to an *adversarial recommender agent*. That is, we duplicate the attacker agent as the attacker simulator. Note that, although the duplicated attacker simulator is the same with the attacker agent after the training of the recommender agent, they are conceptually quite different: the attacker simulator is only used to simulate the attack on representations during the training in order to take privacy into account, while the attacker agent is the malicious third party and is used to evaluate the privacy of the PrNet.

PrNet is trained to optimize:

$$\mathcal{L}_{adversarial}(\Theta) = \sum_{c_T} -\log P(r^T_{ui}|u,i;\Theta_T) + \sum_{c_S} -\log P(r^S_{uj}|u,j;\Theta_S) + \lambda \sum_{c_S} -\log P\Big(\neg y_{uj}|trep(\boldsymbol{a}^{(uj)}_S);\theta\Big),$$

where on the right hand, the first two terms are to optimize losses over user-item interactions in the target and source domains, respectively. The last term is trained to deceive the adversary such that the transferred representations are not useful for the attacker to recover the private information of the source domain. If y_{uj} has C choices, then $\neg y_{uj}$ randomly chooses a class from the other $C-1$ choices. The hyperparameter $\lambda \in \mathbb{R}^+$ is to control the influence from the attacker simulator. PrNet can reduce to CoNet when $\lambda = 0$.

4 EXPERIMENTS

In this section, we conduct experiments to evaluate the performance of both recommendation (Section 4.3) and privacy protection (Section 4.4) on real-world datasets to show the effectiveness of the proposed defense model.

4.1 Dataset

We firstly introduce the datasets. We evaluate the methods on three datasets extracted from a large real-world data. Yelp is a public dataset[1]. It is a knowledge-sharing website and contains information on businesses, users, and user's stars/reviews on businesses.

We preprocess the raw data of Yelp Round 11 and get the following three datasets in four steps.

Firstly, we divide the raw data by the locations of the businesses and reserve the top three locations, which are *Arizona, Nevada,* and *Ontario*.

Secondly, for each of the three datasets, we split it by the categories of the businesses. We reserve the top four categories of Yelp

[1] https://www.yelp.com/dataset/challenge

Dataset	Domain	Field	Statistics
Arizona	Shared	#Users	9,366
	Target	#Items of CategoryI	2,078
		#Interactions	45,079
		Density	0.231%
	Source	#Items of CategoryII	1,762
		#Interactions	39,138
		Density	0.237%
		#Males	2,305
		#Females	2,305
Nevada	Shared	#Users	6,388
	Target	#Items of CategoryI	1,441
		#Interactions	38,322
		Density	0.416%
	Source	#Items of CategoryII	969
		#Interactions	21,044
		Density	0.339%
		#Males	1,574
		#Females	1,574
Ontario	Shared	#Users	2,280
	Target	#Items of CategoryI	2,404
		#Interactions	21,654
		Density	0.395%
	Source	#Items of CategoryII	1,148
		#Interactions	8,272
		Density	0.316%
		#Males	645
		#Females	645

Table 1: Datasets and statistics. CategoryI = {Food, Nightlife}, CategoryII = {Bars, Sandwiches}.

Restaurants, which are *Food, Nightlife, Bars,* and *Sandwiches*. We group the first two categories as CategoryI and the last two categories as CategoryII.

Thirdly, we predict the gender based on the username [39]. We label a user's gender as Male if the corresponding username is predicted as *Male* or *MostlyMale*. Likewise, we label a user's gender as Female if the corresponding username is predicted as *Female* or *MostlyFemale*. We discard the situations where the predicted results are Androgyny or Unknown. Furthermore, we balance the two classes of gender by enforcing the number of users in each gender class to be the same. The balanced user gender dataset is used to train and test the privacy-related models.

Lastly, the four and five stars of user reviews are reserved as positive interactions. We filter users who have less than three interactions since we need to have at least one example for training, validation, and test data respectively for each user.

The statistics are summarized in Table 1 and we can see that all of the datasets have more than 99% sparsity. It is expected that the transfer learning technique is helpful to alleviate the data sparsity issues in these real-world recommendation services.

4.2 Protocol

We introduce the evaluation metrics, recommendation baselines, and implementation details in this section.

4.2.1 Metric. We follow the protocol in [17, 20] to evaluate the item recommendation task by the leave-one-out (LOO) methodology. That is, we reserve one interaction as the test item for each user. We reserve another interaction per user as the development set. We follow the strategy which randomly samples 99 negative items that are not interacted by the user and then evaluate how well the recommender can rank the test item against these negative ones.

For top-K item recommendation, the typical evaluation metrics are hit ratio (HR), normalized discounted cumulative gain (NDCG), and mean reciprocal rank (MRR) [14, 19]. We cut off the ranked list at $K = 10$ [10, 21]. HR measures whether the reserved test item is present in the top-K list and is defined as:

$$HR = \frac{1}{|\mathcal{U}|} \sum\nolimits_{u \in \mathcal{U}} \delta(p_u \leq K),$$

where p_u is the hit position for the test item of user u, and $\delta(\cdot)$ is the indicator function. NDCG and MRR account for the rank of the hit position. They are defined as:

$$NDCG = \frac{1}{|\mathcal{U}|} \sum\nolimits_{u \in \mathcal{U}} \frac{\log 2}{\log(p_u + 1)} \text{ and } MRR = \frac{1}{|\mathcal{U}|} \sum\nolimits_{u \in \mathcal{U}} \frac{1}{p_u}.$$

A higher value indicates better recommendation performance.

The evaluation metric for privacy-related models is accuracy and is introduced in later Section 4.4.

4.2.2 Baseline. We compare with different kinds of recommender systems.

POP: The most popular model predicts a user's preference on an item by the popularity of this item.

BPRMF: Bayesian personalized ranking [35] is a latent factors approach which learns user and item factors via matrix factorization. It is a shallow model and learns on the target domain only.

MLP: Multilayer perceptron [17] is a typical neural collaborative filtering approach which learns the user-item interaction function using neural networks. MLP corresponds to the basic network as described in Section 2.2. It is a deep model and learns on the target domain only.

CSN: The cross-stitch network [28] is a deep transfer learning model which couples the two basic networks via a linear combination of activation maps using a scalar rather than a matrix (see Eq. (3)).

CoNet: A deep transfer learning method for cross-domain recommendation. Compared with CSN, it learns linear combination of activation maps using a matrix rather than a scalar as described in Section 2.2.

PrNet is a deep transfer learning method for cross-domain recommender system. It is an adversarial version of CoNet.

4.2.3 Implementation. For BPRMF, we use LightFM's implementation [34] which is a popular CF library[2]. For MLP, we use the code released by its authors[3]. For CoNet, we use the code released by its authors[4]. Our method is implemented using TensorFlow [2]. Parameters are randomly initialized from Gaussian $\mathcal{N}(0, 0.01^2)$. The optimizer is Adam [22] with initial learning rate in {1e-4, 5e-4, 1e-3}. The size of mini-batch is in {64,128}. The ratio of negative

[2]https://github.com/lyst/lightfm
[3]https://github.com/hexiangnan/neural_collaborative_filtering
[4]http://home.cse.ust.hk/~ghuac/

Dataset	Metric	POP	BPRMF	MLP	CSN	CoNet	PrNet
Arizona	HR	47.13	45.44	47.29	52.26	53.88	49.56*
	NDCG	26.02	26.49	25.94	28.21	29.92	27.40*
	MRR	19.62	20.68	24.59	25.72	26.86	25.42*
Nevada	HR	49.57	42.65	49.89	50.10	50.42	**50.41***
	NDCG	26.59	24.76	26.56	27.05	26.81	**26.70***
	MRR	19.63	19.29	24.30	24.94	24.66	24.26
Ontario	HR	46.37	41.13	47.16	48.59	48.17	47.64*
	NDCG	26.31	24.00	26.42	27.22	26.79	26.64*
	MRR	20.21	18.71	25.19	25.47	25.31	**25.33***

Table 2: Comparison results of different methods on recommendation performance ($\times$ 100%). The star marks indicate that PrNet is better than all three single-domain models. The bold face indicates that PrNet is competitive with the two cross-domain models within $\pm 0.5\%$ relative performance.

sampling is 1. The configuration of hidden layers in the basic network is $[64] * 4$. This is also the network configuration of MLP and CoNet. λ is in $\{0.1, 1, 5\}$.

4.3 Results on Recommendation

There are two goals in privacy-aware recommender systems, i.e., improving recommendation performance in the target domain and protecting private information in the source domain. In this section, we evaluate the recommendation performance of different methods. We expect that PrNet can exploit the knowledge from the source domain to improve the recommendation performance in the target domain, though it needs to protect the privacy of the source domain during knowledge transfer.

We report results on recommendation in Table 2. We have the following observations. Firstly, we can see that PrNet is better than all of single-domain models (POP, BPRMF, and MLP) on all settings except for the MRR metric on the Nevada dataset (but it is almost the same: 24.26 vs. 24.30). This demonstrates that PrNet is an effective transfer learning approach to exploit the source domain knowledge to improve the target recommender systems. In more details, PrNet improves relative 5.59% performance over MLP in terms of NDCG on the Arizona dataset.

Secondly, PrNet is almost comparable with cross-domain models (CSN and CoNet). PrNet even slightly outperforms CSN for the HR metric on the Nevada dataset and outperforms CoNet for the MRR metric on the Ontario dataset. It is shown that PrNet learns an effective privacy-aware representation such that it is also useful to improve the target recommender systems.

In summary, PrNet is better than single-domain models in almost all cases and competitive with cross-domain models in several settings. PrNet learns an effective privacy-aware transferable representation for improving target recommendation.

4.4 Results on Privacy Defense

In this section, we measure the privacy of neural representations by training a classifier to predict private information from it. The accuracy metric is defined as:

$$Accuracy = \frac{\text{num of test examples predicted correctly}}{\text{num of test examples}}.$$

Dataset	Lower bound (MFB)	Upper bound (Trained)	Defense (PrNet)
Arizona	50.00	52.13 $\pm$0.5267	51.65 $\pm$0.6165
Nevada	50.00	54.94 $\pm$0.7377	52.24 $\pm$0.4905
Ontario	50.00	55.37 $\pm$1.0402	54.34 $\pm$0.2198

Table 3: Comparison results (mean$\pm$std) between baselines and the defense method. The metric is accuracy ($\times$ 100%).

Intuitively, the transferred representations may imply private information about the source domain and can be eavesdropped to recover it with a reasonable accuracy. Therefore, it is more difficult to exploit the transferred representations to recover the user's gender when the accuracy metric is lower. As a result, a lower accuracy is better for privacy protection.

We consider two baselines. *As a lower bound baseline*, the most frequent baseline (MFB) is a reasonable classifier which determines the labels of private information as the most frequent class in the training data. It is equivalent to random guess since the user gender's dataset is balanced as described in Section 4.1.

As an upper bound baseline, we train a classifier from the transferred representations of CoNet (Trained). Thereafter, the transferred representations are not protected and not defended from a malicious attacker. This is the scenario of attack on representations as described in Section 2.3.

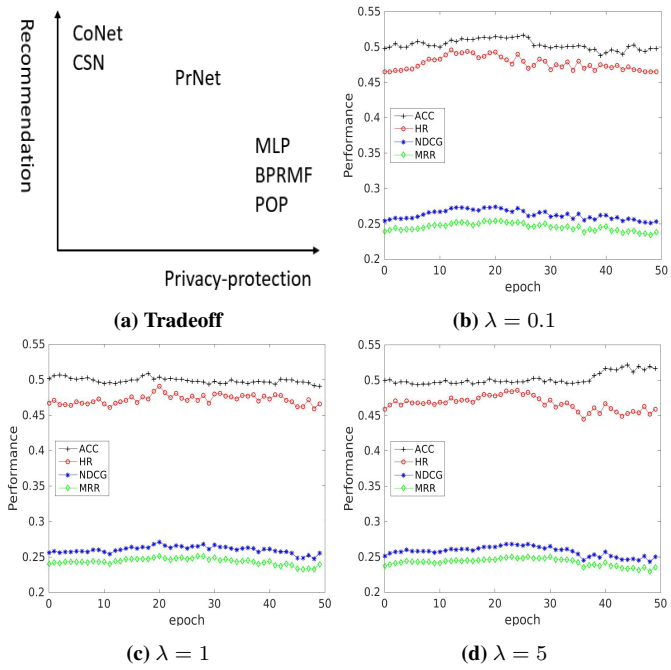
We report results on privacy defense in Table 3. We have the following observations. Firstly, we can see that both Trained CoNet and PrNet models are more accurate than random guess on three datasets. It is demonstrated that the transferred representations indeed learn private information in the source domain implicitly, even though the private information (i.e., user gender) is not present in the input explicitly. As a result, they can be exploited by an attacker to recover private information from it with a reasonable accuracy (i.e., better than random guess). Secondly, we can see that it is more difficult for an attacker to predict accurately the user's gender by using the transferred representations learned from PrNet than CoNet. In summary, PrNet learns an effective privacy-aware representation such that it is not much useful to recover the private information.

4.5 Analysis

We analyze the tradeoff of utility and privacy, and the effect of λ.

Tradeoff. Summarizing the results on both recommendation and privacy defense, we plot an illustration for model performance as shown in Fig. 2a. Firstly, the single-domain recommendation models (POP, BPRMF, and MLP) naturally protect the privacy in the source domain since they do not use any information from it. The data sparsity issue, however, limits their performance in the target domain. Secondly, the cross-domain models (CSN, CoNet, and PrNet) naturally get better recommendation performance since they transfer the knowledge from the source domain to alleviate the data sparsity issue in the target domain. Thirdly, PrNet achieves a good trade-off between recommendation performance and privacy protection since it learns a privacy-aware transferable representation.

Effect of λ. We show the optimization performance of the recommender and the attacker varying with λ . Results are shown on the Arizona dataset only due to space limit and the trends are similar on



(a) Tradeoff **(b)** $\lambda = 0.1$ **(c)** $\lambda = 1$ **(d)** $\lambda = 5$

Figure 2: (a): Tradeoff on utility and privacy. (b)-(d): Performance varying with λ.

Nevada and Ontario. As shown in Figures 2b-d, the performance of the recommender (HR, NDCG, and MRR) gradually improves until 30 iterations for all three values of λ. The accuracy of the attacker (ACC) begins to degrade after 25 epochs when $\lambda = 0.1$ and 20 epochs when $\lambda = 1$. This means that the attacker has difficulty in recovering the private information. For large value of $\lambda = 5$, the recommendation performance degrades while the attacker improves after 35 epochs. So, we recommend to set $\lambda \in \{0.1, 1\}$.

5 RELATED WORKS

We review related works on transfer learning for cross-domain recommendation and on privacy protection, attack, and defense.

Cross-domain recommendation [6] is an effective technique to alleviate the data sparsity issue in one domain by exploiting the knowledge from other domains. Typical methods apply matrix factorization [32, 37, 41] and representation learning [14, 25, 26, 40, 45] on each domain and share the user (item) factors, or learn a cluster level rating pattern [23, 44]. Transfer learning is to improve the target performance by exploiting knowledge from a source domain [8, 13, 15, 31, 46].

One transfer strategy (two-stage) is to initialize a target network with transferred representations from a pre-trained source network [29, 43]. Another transfer strategy (end-to-end) is to transfer knowledge in a mutual way such that the source and target networks benefit from each other during the training, with examples including the cross-stitch networks [28] and collaborative cross networks [20]. These cross-domain methods have access to the input or representations from source domain. Therefore, it raises a concern on privacy leaks and provides an attack possibility during knowledge transfer.

User privacy is an increasing interest and real-world recommendation systems must confront this issue under laws such as the General Data Protection Regulation (GDPR) [38]. Typical private information includes user's age, gender, location, and occupational class. User's demographic information has been used in product recommender system[47] and video recommendation [9]. They showed that demographic features can be used to improve the model performance or to alleviate the data sparsity issue. There is a trade-off between the utility and privacy of the sensitive variables. Sometimes, it may sacrifice performance in order to protect privacy.

Privacy-preserving recommender systems are studied from cryptographic and security [4, 5]. Differential privacy quantifies and bounds privacy loss based on the principle that the published results of a computation should be independent of whether any individual record is present in or absent from the computation's input [12]. It has been used in recommender systems to trade-off the accuracy and privacy using the noise-adding trick, as practically evaluated on the Netflix Prize contest dataset [27]. It provides a privacy guarantee for the releasing of information without compromising confidential data, with noise added in the released information. Differential privacy is applicable to the case of explicit private information where sensitive attributes are part of the input. In this case, noise can be added to the input. Differential privacy, however, is not directly applicable to the case of implicit private information where sensitive attributes are not in the input. Because of in this case, we do not exploit the private variables for the recommendation and hence there is no chance of adding noise to them for publishing the results.

Recently, federated machine learning [42] is used in recommender systems for protecting user privacy and meeting laws [3]. Federated matrix factorization [7] uploads the gradient information instead of raw data to a server. Furthermore, homomorphic encryption is used since the attacker can still recover the private information by eavesdropping the gradient information. The attack scenario investigated in this article is applicable to the case without homomorphic encryption, that is, attacking on the gradient information. Moreover, we consider the cross-domain recommendation and attack on representations during knowledge transfer, instead of single-domain recommendation. That is, we have access to the raw data in the target domain and we need to protect the privacy in the source domain.

6 CONCLUSION

We presented an attack scenario on the representations during knowledge transfer in the cross-domain recommendation and proposed a defense model PrNet to learn a privacy-aware transferable representation. The effectiveness of this privacy-aware transferable representation is evaluated on both recommendation performance and privacy protection. We showed that PrNet achieves a good trade-off between the utility and privacy of hidden representations. PrNet learns an effective privacy-aware transferable representation both for improving target recommendation and for making the attacker more difficult to recover the private information of source domain from it.

Besides measuring the privacy of representations by the ability of an attacker to predict it accurately, we hope to develop an alternative metric which is independent of a specific classifier in future works.

REFERENCES

[1] California Consumer Privacy Act (AB-375). https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375. 2018.

[2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation*, pages 265–283, 2016.

[3] Muhammad Ammad-ud din, Elena Ivannikova, Suleiman A Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. Federated collaborative filtering for privacy-preserving personalized recommendation system. *arXiv preprint arXiv:1901.09888*, 2019.

[4] John Canny. Collaborative filtering with privacy. In *Proceedings 2002 IEEE Symposium on Security and Privacy*, pages 45–57, 2002.

[5] John Canny and John Canny. Collaborative filtering with privacy via factor analysis. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 238–245, 2002.

[6] Iván Cantador, Ignacio Fernández-Tobías, Shlomo Berkovsky, and Paolo Cremonesi. Cross-domain recommender systems. In *Recommender systems handbook*, pages 919–959. 2015.

[7] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. Secure federated matrix factorization. *arXiv preprint arXiv:1906.05108*, 2019.

[8] Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. An efficient adaptive transfer neural network for social-aware recommendation. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 225–234, 2019.

[9] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.

[10] Evangelia Christakopoulou and George Karypis. Local latent space models for top-n recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1235–1243, 2018.

[11] Maximin Coavoux, Shashi Narayan, and Shay B. Cohen. Privacy-preserving neural representations of text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1–10, 2018.

[12] Cynthia Dwork. Differential privacy. *Encyclopedia of Cryptography and Security*, pages 338–340, 2011.

[13] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 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*, 2015.

[14] Chen Gao, Xiangning Chen, Fuli Feng, Kai Zhao, Xiangnan He, Yong Li, and Depeng Jin. Cross-domain recommendation without sharing user-relevant data. In *The World Wide Web Conference*, pages 491–502, 2019.

[15] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, and Depeng Jin. Neural multi-task recommendation from multi-behavior data. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1554–1557, 2019.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[18] Dirk Hovy. Demographic factors improve classification performance. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 752–762, 2015.

[19] Binbin Hu, Yuan Fang, and Chuan Shi. Adversarial learning on heterogeneous information networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 120–129, 2019.

[20] Guangneng Hu, Yu Zhang, and Qiang Yang. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 667–676, 2018.

[21] Wang-Cheng Kang and Julian McAuley. Candidate generation with binary codes for large-scale top-n recommendation. In *Proceedings of The 28th ACM International Conference on Information and Knowledge Management*, 2019.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *The 3rd International Conference for Learning Representations*, 2015.

[23] Bin Li, Qiang Yang, and Xiangyang Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[24] Meng Li, Liangzhen Lai, Naveen Suda, Vikas Chandra, and David Z Pan. Privynet: A flexible framework for privacy-preserving deep neural network training. *arXiv preprint arXiv:1709.06161*, 2017.

[25] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. Pi-net: A parallel information-sharing network for shared-account cross-domain sequential recommendations. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 685–694, 2019.

[26] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. Cross-domain recommendation: an embedding and mapping approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2464–2470, 2017.

[27] Frank McSherry and Ilya Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 627–636, 2009.

[28] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[29] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

[30] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *2008 Eighth IEEE International Conference on Data Mining*, pages 502–511, 2008.

[31] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[32] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. Transfer learning in collaborative filtering for sparsity reduction. In *Twenty-fourth AAAI conference on artificial intelligence*, 2010.

[33] Daniel Preoţiuc-Pietro, Vasileios Lampos, and Nikolaos Aletras. An analysis of the user occupational class through twitter content. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1754–1764, 2015.

[34] Steffen Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.

[35] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461, 2009.

[36] Sara Rosenthal and Kathleen McKeown. Age prediction in blogs: A study of style, content, and online behavior in pre-and post-social media generations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 763–772, 2011.

[37] Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658, 2008.

[38] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017.

[39] Kamil Wais. Gender prediction methods based on first names with genderizer. *The R Journal*, 8(1):17–37, 2016.

[40] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1245–1254, 2017.

[41] Chunfeng Yang, Huan Yan, Donghan Yu, Yong Li, and Dah Ming Chiu. Multi-site user behavior modeling and its application in video recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 175–184, 2017.

[42] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):12, 2019.

[43] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[44] Feng Yuan, Lina Yao, and Boualem Benatallah. Darec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4227–4233, 2019.

[45] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362, 2016.

[46] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

[47] Xin Wayne Zhao, Yanwei Guo, Yulan He, Han Jiang, Yuexin Wu, and Xiaoming Li. We know what you want to buy: a demographic-based system for product recommendation on microblogs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1935–1944, 2014.