Applied Network Science

RESEARCH Open Access

Framework for role discovery using transfer learning

Shumpei Kikuta[1*], Fujio Toriumi[1], Mao Nishiguchi[1], Shu Liu[1], Tomoki Fukuma[1,2], Takanori Nishida[3] and Shohei Usui[3]

*Correspondence:
lskabc11ciel30@gmail.com
[1] University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, 113-8654 Tokyo, Japan
Full list of author information is available at the end of the article

Abstract

Discovering the node roles in a network helps to solve diverse social problems. Role discovery attempts to predict the node roles from a network structure, and this method has been extensively studied in various fields. Role discovery using transfer learning has many advantages, but methods using this approach face two kinds of problems: domain-shift problems and model selection. To address these problems, we propose a general framework that includes network representation learning, domain adversarial learning for suppressing domain-shift problems, and model selection without using target labels. As a result of computational experiments, we show on publicly available datasets that the proposed model outperforms conventional methods, the proposed model selection method performs well without using target labels, and the proposed method can be used in real-world datasets. Furthermore, we found that our framework suppressed domain-shift problems, worked well even with differences between networks, and could handle imbalanced classes.

Keywords: Role discovery, Transfer learning, Domain adversarial learning, Model selection

Introduction

Each individual node in a network can have different functions, which are called “roles" (e.g. central nodes, peripheral nodes, clique members). Node roles can be discovered from the structure of their networks, such as the locations of nodes within the entire network, or from the structure of neighboring nodes. Role discovery attempts to take a graph and pick out sets of structurally similar nodes (Rossi and Ahmed 2015). Role discovery has become an active field within network research, and it is expected to be useful for solving diverse social problems. For instance, in a social network, identifying and focusing on star-centers for advertisement could increase profit. Furthermore, observing chronological structural changes in individual nodes can lead to the identification of abnormal nodes (Rossi et al. 2013).

Among several conventional methods, unsupervised learning has become mainstream (Rossi and Ahmed 2015). Studies using this approach have predicted node roles by generating the features of nodes, clustering them, and assigning a role to each cluster. Although unsupervised learning can be effective in mining unknown information with relatively low complexity, confirming the results and comparing methods are difficult tasks due to the absence of any supervisor.

It would be advantageous to use supervised learning to overcome the difficulties of unsupervised learning and improve performance as well. However, when there are not sufficient labels on a given network, annotating appropriate roles to some nodes requires deep domain knowledge, which causes a high cost.

Transfer learning, where a predictor learned on a source network is applied to a target network, can reduce labeling difficulty because it allows the source network to be used repeatedly and applied to various networks once such a source network has been created. Furthermore, data and pretrained models from different domains help to create source networks easily, which facilitates transfer learning. Therefore, we focused on transfer learning and proposed a high-performance method for role discovery.

A few studies (Ribeiro et al. 2017; Henderson et al. 2012) have investigated transfer learning for role discovery, but their methods are likely to encounter domain-shift (Pan et al. 2010) problems, where feature representations (i.e. domains) are slightly different between the source and target domains. In such cases, when a model trained on a source domain is applied to a target domain, the model experiences a significant drop in performance. Our approach addresses domain-shift problems by applying a domain adaptation technique, which attempts to transfer knowledge from the source domain to the target domain undergoing a domain-shift. In particular, we use domain adversarial learning, which makes the two node representations both similar and discriminative for role discovery.

Another difficulty in transfer learning is model selection. Unlike supervised learning settings, we have source and target data, neither of which can be considered independent and identically distributed (IID). For this reason, model selection tailored for transfer learning is necessary. Conventional studies (Zhong et al. 2010; Sugiyama et al. 2007) have used the label or its ratio on the target data, which is not available in unsupervised learning, requiring enormous efforts in making annotations or inferences. Therefore, we propose a new model selection method without using labels on the target network.

In this paper, we attempt to more accurately infer the node roles on a target network using a source network. We propose a general framework for role discovery that includes node representation learning, domain adaptation, and model selection. Node representation learning is used for generating node representations, where structurally similar nodes have representations that are as similar as possible regardless of which network nodes belong to. Then, we tailor domain adversarial learning (Ganin et al. 2016) for role discovery to make node representations more domain-invariant and discriminative. Furthermore, we introduce a validation network and propose a new model selection method without using target labels.

A computational experiment on a barbell graph and a toy network shows that our proposed method outperforms the conventional methods and that domain-invariant representations leads to high accuracy. In addition, we discovered through an experiment on the toy network and on Zachary's Karate Club (Zachary 1977) that our proposed model selection using a validation network has comparable performance to a model tuned on the target network. Finally, we further explored whether the proposed method could achieve high accuracy in comparing real-world air-traffic networks, and we show an empirical way of selecting the appropriate validation network in a practical role discovery problem.

This paper extends a previous study (Kikuta et al. 2019) by proposing a general framework, presenting a new model selection method, and conducting experiments on different datasets.

Related work

Role discovery

Role discovery attempts to divide nodes into sets of structurally similar nodes (Rossi and Ahmed 2015). Methods for doing this can be categorized into unsupervised learning and transfer learning.

Unsupervised learning

The first approach to unsupervised learning uses graph-based methods, which discover roles directly from graph representations (e.g. adjacency matrices). Blockmodel (White et al. 1976) creates a compact representation called a role-interaction graph and then groups nodes according to their structural similarities. Stochastic Blockmodel (Nowicki and Snijders 2001) uses a less strict definition of a role, which makes applying it to real-world datasets easier. Burt (1976) computed a similarity matrix among nodes by calculating the correlations between the rows of the adjacency matrix and then clustering the nodes based on the similarity matrix. Brandes and Lerner (Brandes and Lerner 2010) computed the eigenvectors of the adjacency matrix for discovering specific structural patterns (star-centers, bridges, etc.). Although graph-based methods use a strict notion of a role and lead to easier interpretation of the results, they tend to require a lot of computation, which makes applying them to today's large-scale networks difficult.

Another approach to unsupervised learning uses feature-based methods. Methods using this approach involve the steps of embedding nodes into low-dimensional space, clustering them, and assigning a role to each cluster. RolX (Henderson et al. 2012) extracts the structural features, applies non-negative matrix factorization to a feature matrix, and then assigns roles. struc2vec (Ribeiro et al. 2017) computes the structural distance between nodes heuristically, grasps the context by a random walk, and brings structurally similar nodes closer using Skip-gram (Mikolov et al. 2013). DRNE (Tu et al. 2018) uses a recursive strategy to obtain node representations based on regular equivalence (Rossi and Ahmed 2015).

Feature-based methods are flexible enough to determine complex roles on large-scale networks and are widely used these days, but choosing the most effective features to use is difficult.

Transfer learning

Transfer learning is widely used in various fields, such as link prediction (Cao et al. 2010) and image classification (Duan et al. 2012). Transfer learning is also used to improve the accuracy of role discovery (Rossi and Ahmed 2015). However, few studies have used transfer learning in role discovery. RolX is used for transfer learning by extracting features from both networks, training the classifier on the source network, and applying it to the target network. However, ReFeX (Henderson et al. 2011), used for extracting features in RolX, is not guaranteed to perform well for transfer learning, especially when the difference in size between the two networks is large. struc2vec can also be used in transfer learning in the same way because it attempts to map structurally similar nodes closer to each other, even if they belong to different networks. In struc2vec, the distance $f_k(u, v)$ between two nodes u and v can be computed as follows:

$$f_k(u, v) = f_{k-1}(u, v) + g(s(R_k(u)), s(R_k(v))),\\ k \geq 0, |R_k(u)|, |R_k(v)| > 0, \tag{1}$$

where k represents the number of hops from one node to another node and $R_k(u)$ represents the degree lists of k-hop nodes from node u. $s(.)$ represents the sorting function, and g computes the distance between two lists. However, if there is a big difference in size between graph G_1 and graph G_2, the degree distributions of their results naturally become quite different. Thus, the term $g(s(R_k(u)), s(R_k(v)))$, where $u \in G_1$ and $v \in G_2$, becomes smaller despite their structural similarities. Hence, $f_k(u, v)$ becomes smaller and the node representations of u and v become more distant even if u and v play the same roles. In other words, two domains, generated by struc2vec, create each subspace (i.e. domain-shift). To address this problem, we use domain adaptation.

Domain adaptation

The objective of domain adaptation is to transfer knowledge from one field (source data) to another (target data), where both feature representations are slightly different (i.e. a domain-shift problem Pan et al. (2010)). Many studies (Mirza and Osindero 2014; Long et al. 2015; Xie et al. 2018) have examined domain adaptation, and domain-invariant feature learning with adversarial learning analogous to generative adversarial networks (Goodfellow et al. 2014) has performed well (Ganin et al. 2016; Hoffman et al. 2017; Bousmalis et al. 2017).

Domain adversarial learning, proposed by Ganin et al. (2016) for computer vision tasks, attempts to not only solve a classification task in a source domain but also to make both domains similar. Specifically, a discriminator is trained to predict which domain a sample comes from, and a label-predictor attempts to minimize the task loss as well as fool the discriminator by making the two domains similar. Both models are updated alternately as each model tries to increase the loss of the other model. Feature representations after convergence are expected to be domain-invariant and discriminative for the task.

FRAGE (Gong et al. 2018) applies domain adversarial learning to the field of natural language processing. In natural language processing, word representations derived from Skip-gram (Mikolov et al. 2013) can be divided by their frequency of appearance in a corpus (Mu et al. 2017). Words can be classified into two types (i.e. common words and rare words), and FRAGE attempts to transform both word representations so that they are similar to each other as well as to solve a task (e.g. machine translation). After convergence, word representations are expected to be frequency-agnostic and discriminative for the task.

To the best of our knowledge, conventional studies in role discovery have never used domain adversarial learning. We tailor domain adversarial learning for role discovery to improve performance.

Model selection

In supervised learning, we assume the data to be IID. However, in transfer learning, the data consist of the source data with labels and the target data without labels, which makes it impossible to select models in the same manner as in supervised settings (e.g. hold-out and k-fold cross validation Kohavi et al. (1995)).

Choosing a model using a few target labels has been used by both Long et al. (2016); Russo et al. (2018), and this allows the model to be unbiased and of similar variance to the target domain. However, using the target label is not preferable because, in unsupervised settings, the real-world datasets do not include labels. IWCV (Sugiyama et al. 2007) searches for better hyperparameters given the ratios of the target labels. It has been theoretically guaranteed that the model is unbiased, but if the density ratio is not available, its estimation requires much computation. TrCV (Zhong et al. 2010) requires target labels in the approximation of the conditional distribution, which causes the same problem as the model selected with the target labels.

We propose a model selection method with a labeled validation network, which allows us to select models without the labeled target data.

Proposed framework

In this section, we introduce our framework for transferring node roles from a source network to a target network. Our framework consists of three components: pretrained node representations, domain adversarial learning, and model selection. Pretrained node representations, where structurally similar nodes have closer distances, are needed to make discriminative representations in each domain, which facilitates transfer learning. However, node representations might create individual subspaces depending on the network, so we use domain adversarial learning to make the two node representations similar. Furthermore, since node representation learning and domain adversarial learning have various hyperparameters that need to be tuned, we present a model selection method without using target labels. Our proposed framework is so general that one component can be replaced with another. For example, a different network representation learning method from ours could be used. Furthermore, the proposed model can deal with imbalanced class problems by operating the source network as shown in the experiment.

Notation

Let V denote all nodes in graphs. We define $\theta^E \in R^{d\times|V|}$ as node representations, where d is the number of dimensions of node representations and $|V|$ is the number of V. Let V_S denote all nodes in a source network and V_T all nodes in a target network. θ^E can be divided into θ^E_S and θ^E_T, indicating node representations of source nodes (i.e. the source domain) and target nodes (i.e. the target domain), respectively. θ^E_v denotes the node representation of node v. We define f_{θ^D} as the discriminator with parameter θ^D and f_{θ^R} as the role-model with parameter θ^R. Both models are composed of feed-forward neural networks.

Pretrained node representation

First, the source domain and the target domain must be initialized in a way that two nodes having similar structures also have similar representations in each domain, and two structurally similar nodes have representations that are as close as possible in the other domains. In order to meet the above criteria, we use struc2vec (Ribeiro et al. 2017) for node representation learning because of its state-of-the-art performance, and as mentioned in "Related work" section, it can also be used for transfer learning. The details can be found in the original paper (Ribeiro et al. 2017).

Domain adversarial learning

We assume that both domains generated by struc2vec are similar, but the two are slightly different across domains, which lowers the performance of transfer learning. To address this problem, we use domain adversarial learning. The overview of the method is shown in Fig. 1. Our proposed domain adversarial learning has two neural networks: a discriminator and a role model. We explain each model in the following sections.

Training the discriminator

The objective of the discriminator is to learn to tell whether a node comes from a source network or a target network. The discriminator inputs are node representations, and the outputs are confidence scores between 0 and 1; the score approaches 1 if the discriminator predicts that the input representations are derived from the target network, and 0 if derived from the source network. Then let $L_D\left(V;\theta^D,\theta^E\right)$ denote the discriminator loss using negative log likelihood:

$$L_D\left(V;\theta^D,\theta^E\right) = \frac{1}{|V_S|}\sum_{v\in V_S}\log f_{\theta^D}\left(\theta_v^E\right) + \frac{1}{|V_T|}\sum_{v\in V_T} log\left(1 - f_{\theta^D}\left(\theta_v^E\right)\right). \quad (2)$$

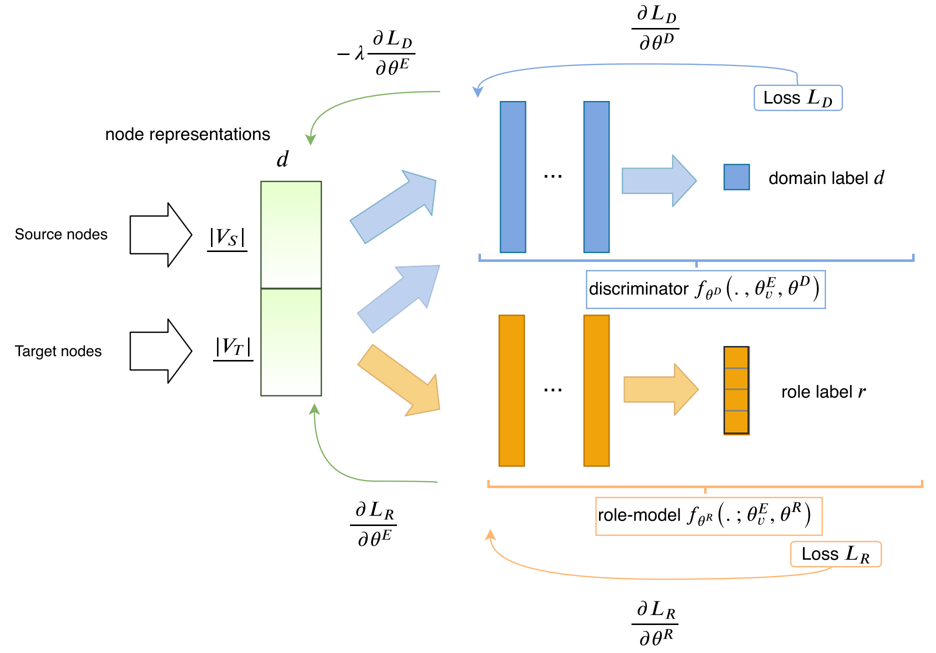
$\frac{\partial L_D}{\partial \theta^D}$
$-\lambda\frac{\partial L_D}{\partial \theta^E}$
Loss L_D
node representations
d
...
domain label d
Source nodes
$|V_S|$
discriminator $f_{\theta^D}(.\,,\theta_v^E,\theta^D)$
Target nodes
$|V_T|$
...
role label r
$\frac{\partial L_R}{\partial \theta^E}$
role-model $f_{\theta^R}(.\,;\theta_v^E,\theta^R)$
Loss L_R
$\frac{\partial L_R}{\partial \theta^R}$

Fig. 1 Proposed domain adversarial learning. This includes a discriminator (blue) and a role model (orange). Thick arrows indicate forward propagation and narrow arrows indicate backward propagation

When θ^R and θ^E are fixed, the discriminator will learn by solving the optimization equation below:

$$\max_{\theta^D} -\lambda L_D\left(V;\theta^D,\theta^E\right), \tag{3}$$

where λ is a hyperparameter to balance the loss.

Training the role model

The goals of the role model are to predict the role labels of nodes and to fool the discriminator. These goals are thought to be attainable by making both representations similar, which allows the role model trained on the source domain to be applicable to the target domain. The inputs of the role model are node representations, and the outputs are the probabilities of the corresponding role labels. Given that role discovery can be considered a classification task, let $L_R\left(S;\theta^R,\theta^E\right)$ denote the role-model loss below when using the cross entropy loss $J(.,.)$:

$$L_R(S;\theta^R,\theta^E) = -\frac{1}{|S|}\sum_{(v,r)\in S} J\left(f_{\theta^R}\left(\theta^E_v\right),r\right), \tag{4}$$

where S is a source dataset including node v and the corresponding role label r. When θ^D is fixed, the role model will be trained to solve the classification task and to fool the discriminator as well.

$$\min_{\theta^R,\theta^E} L_R\left(S;\theta^R,\theta^E\right) - \lambda L_D\left(V;\theta^D,\theta^E\right). \tag{5}$$

Note that this procedure updates not only θ^R but also θ^E to transform node representations to be more domain-invariant and discriminative.

Objective function

Based on the adversarial learning framework (Goodfellow et al. 2014), we train the role model and the discriminator by following the min-max strategy:

$$\min_{\theta^R,\theta^E}\max_{\theta^D} L_R\left(S;\theta^R,\theta^E\right) - \lambda L_D\left(V;\theta^D,\theta^E\right). \tag{6}$$

θ^D, θ^E, and θ^R should be optimized by an iterative method. Algorithm 1 shows how to train the model and obtain the node representations using domain adversarial learning. We use a gradient descent algorithm for the sake of simplicity.

In the testing phase, θ^E_T is input to the role model, and the corresponding role labels are predicted.

Model selection

Pretrained node representation learning and domain adversarial learning have many parameters that need to be tuned. In order to select a better model, we prepared a labeled validation network, which is structurally similar to the source and target networks. An overview of the method is shown in Fig. 2. First, node representations are generated among three networks together, and we obtain the source domain S, the validation domain V, and the target domain T. Next, we treat both T and V as a new target domain $T'(= T \cup V)$; applying domain adversarial learning between S and T' makes both domains domain-invariant, which results in similarity among S, V and T. We select the model in which performance is maximized in the validation network because it is expected to

Algorithm 1 Domain Adversarial Learning for Role Discovery

```
Require:
  Dataset $S = \{(v_i, r_i)\}_{i=1}^{|V_S|}$, $T = \{(v_i\}_{i=1}^{|V_T|}$, All nodes $V$, Hidden layer size: $H$
  $\theta^E \leftarrow struc2vec()$.
  $\theta^R, \theta^D \leftarrow random\_init(H)$.
  repeat
      # Update $\theta^D$ by gradient ascent according to Eq. (3)
      $L_D(V;\theta^D,\theta^E) = \frac{1}{|V_S|}\sum_{v \in V_S} \log f_{\theta^D}(\theta^E_v) + \frac{1}{|V_T|}\sum_{v \in V_T} log(1 - f_{\theta^D}(\theta^E_v))$.
      $\theta^D \leftarrow \theta^D - \mu_D \lambda \frac{\partial L_D}{\partial \theta^D}$.

      #Update $\theta^R, \theta^E$ by gradient descent according to Eq. (5) with $S$.
      $L_R(S;\theta^R,\theta^E) = -\frac{1}{|S|}\sum_{(v,r)\in S} J(f_{\theta^R}(\theta^E_v), r)$,
      $\theta^R \leftarrow \theta^R - \mu_R \frac{\partial L_R}{\partial \theta^R}$.
      $\theta^E \leftarrow \theta^E - \mu_R(\frac{\partial L_R}{\partial \theta^E} - \lambda \frac{\partial L_D}{\partial \theta^E})$.
  until Converge
Ensure: $\theta^R, \theta^E, \theta^D$
```

have high performance in the target domain as long as the three networks are structurally similar.

Note that the key to having better model selection using our proposed method is to obtain a validation network that is structurally similar to the source and target networks. We empirically show how to select the validation network in the "Experiment" section.

Experiment

In this section, we report on four experiments we conducted to validate our framework. First, we used a barbell graph, which had many structurally identical nodes, to verify that domain-adversarial learning improved accuracy. Next, in a toy network, we evaluated our proposed model selection approach and showed how imbalanced datasets can be handled. Then, we applied our framework to a benchmark dataset, Zachary's Karate

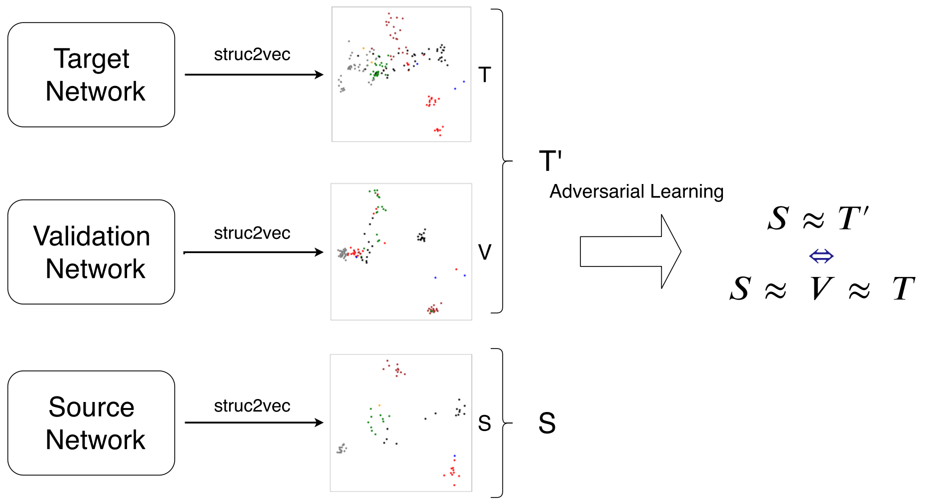

Fig. 2 Model selection using validation network

Club (Zachary 1977), to show that our framework gave us a reasonable interpretation. Finally, we used an air-traffic network to show how we selected the validation network and applied our method to a practical setting.

Baseline model

We used struc2vec (Ribeiro et al. 2017) and RolX (Henderson et al. 2012), which learn node representations based on structural similarity, as baseline models. In struc2vec, we generated node representations in both networks at the same time and passed them into a three-layered neural network. We set the window size to 5, the walk length to 80, the number of walks per node to 20, and the number of dimensions to 64. In RolX, we extracted feature representations in both networks together, assigned optimal roles for RolX in an unsupervised way, and then put them into a three-layered neural network. We used ReFeX for extracting node features and Minimum Description Length (Rissanen 1978) for determining the optimal number of roles.

Note that the classifier was trained based on node representations and the corresponding labels on the source data, and that we evaluated the predicted target labels for fair comparison in both baseline methods.

Parameter settings

We used struc2vec for the initialization of node representations in the same setting as above. Both the discriminator and the role model were three-layered neural networks. The number of hidden nodes in both models was ($x \rightarrow x/2 \rightarrow x/4 \rightarrow y$), where x represents the number of input dimensions and y represents the number of roles. On the second and third layers, both models used Dropout (Srivastava et al. 2014) to prevent overfitting.

Hyperparameters were tuned using the grid-search strategy, where the epoch number was [500, 1000, 5000], the dropout rate was [0.25, 0.5], each learning rate of the role model and the discriminator was [0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001], and λ was [0.01, 0.1, 1, 10, 100]. Note that Ganin et al. (2016) attempted to remove noise by changing λ gradually from 0 to 1 during the training phase, but since λ can be considered one of the parameters, we tuned it using the grid-search strategy. We used Adam (Kingma and Ba 2014) as an iterative optimization method.

Barbell graph

Dataset

The barbell graph $B(h,k)$ is composed of two complete graphs K_1 and K_2 with h nodes in each and one path graph P with a length of k that links both complete graphs. $\{p_1, p_2, ...p_k\}$ denotes nodes of P, and p_i is connected to $p_{(i+1)}(i = 1, 2, ..., k-1)$. p_1 connects b_1 from K_1, and p_2 connects b_2 from K_2. Nodes $\{K_1/\{b_1\}\} \cup \{K_2/\{b_2\}\}$, (b_1, b_2) and $(p_i, p_{(k-i)})(i = 1, 2, ..., k-1)$ are all structurally identical.

In our experiment, we used $B^a(5,5)$ as the target network and $B^b(n,n)$ $(n = 5, 10, 20, 30, ..., 100)$ as the source network. We assumed that RolX and struc2vec had domain-shift problems when the source network size was bigger, so we showed how the proposed method deals with domain-shift problems. As shown in Fig. 3a, we roughly divided nodes into two types: nodes in the complete graph playing role r_1 and nodes in the path playing role r_2. Since the barbell graph has two complete graphs, the ratio of

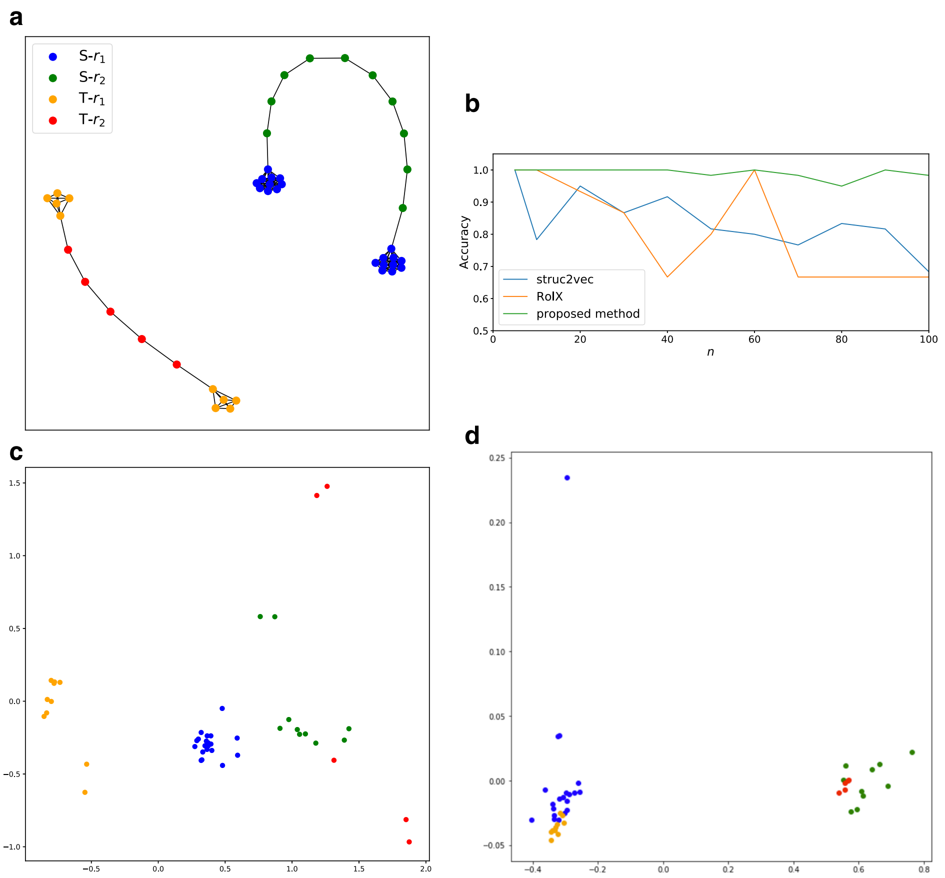

Fig. 3 Barbell graphs. **a** $B^a(5,5)$(lower left) and $B^b(10,10)$(upper right), where color indicates different roles in the same domain and S and T represent the source and the target networks, respectively. **b** Transition of the accuracy when we changed the source network size n. Nodes are mapped by (**c**) struc2vec and (**d**) proposed method. Visualization was performed by PCA

$r_1 : r_2$ equals $2 : 1$. In this experiment, we selected models by using the target labels for all models, since the goal of this experiment was to show whether domain adversarial learning improved performance. We repeated the experiment five times with different random seeds and made our evaluation using the average accuracy.

Results

Figure 3b shows the transition curve of the accuracy when we gradually changed the source network size n. struc2vec classifies correctly in the case of $n = 5$, which indicates that struc2vec can be used when the target and the source network are the same size. However, it results in low accuracy as the network size gets larger. Even in the case of $n = 100$, the accuracy is close to 0.667, which is the same accuracy as random prediction. RolX worked well until the size of the source network was two times bigger than that of the target network ($n = 10$). However, the performance when using RolX decreased when the difference in sizes increased (except for $n = 60$). Therefore, both conventional methods failed in predicting role labels correctly if the sizes were quite different.

By contrast, our proposed method continued to predict the roles with high accuracy. This indicates that our proposed method is robust to differences in network size and may discover roles more accurately in many situations.

Visualization

Next, we visualized node representations learned by our proposed model and struc2vec to analyze how the proposed method can achieve high accuracy. Figure 3c shows the visualization results of node representations generated by struc2vec. In the same domain, nodes playing roles r_1 and r_2 had similar representations, which is ideal for transfer learning. However, in different domains, nodes playing $r_1$and r_2 had different representations. In other words, node representations from different domains were farther apart than they should be and created each subspace (i.e. a domain-shift problem), which led to low accuracy. As shown in Eq. (1), struc2vec calculates the distance between two nodes based on their degrees. The difference in scale between two networks leads to a difference in degree and the difficulty of judging structurally similar nodes as different across the domains.

By contrast, as shown in Fig. 3d, the proposed method successfully brought nodes playing the same role closer regardless of the networks these nodes belonged to. Adversarial learning allowed our proposed method to make the node representations domain-invariant in practice and more discriminative, which made applying the model trained on the source network to the target network easier and led to higher accuracy.

Toy network

Dataset

In a toy network, we evaluated our proposed model selection method and showed how to handle imbalanced datasets. We created the source network by combining four basic topology networks (mesh, ring, star, and tree) as shown in Fig. 4. All node role labels were set as "mesh" in the mesh network and as "ring" in the ring network. In the star network, the central node role label was set as "star center", while the surrounding nodes were set as "peripheral". In the tree network, the root node role label was set as "root of tree", the leaf nodes as "leaf of tree", and the internal nodes as "internal of tree".

We prepared the validation network (Fig. 5) and the target network (Fig. 6) in a way that made the distribution of the validation network similar to that of the target network. Roles in the target network and the validation network followed the rules of the source network, but since the connected nodes changed their roles, we removed them after node representations were generated.

We used Macro-F1 and Micro-F1 as evaluation metrics due to the imbalance. We repeated the experiments five times, and we used the average of the results.

Results

The classification results of the toy network are shown in Table 1. Among the methods that were target tuned, our proposed method had the best Micro-F1, while struc2vec performed slightly better than the proposed method in Macro-F1. In our method, domain adversarial learning attempted to maximize accuracy with cross-entropy loss, which prioritizes the class in which ratios are higher and prevents the model from predicting the class in which ratios are lower. We address this issue in the next section.

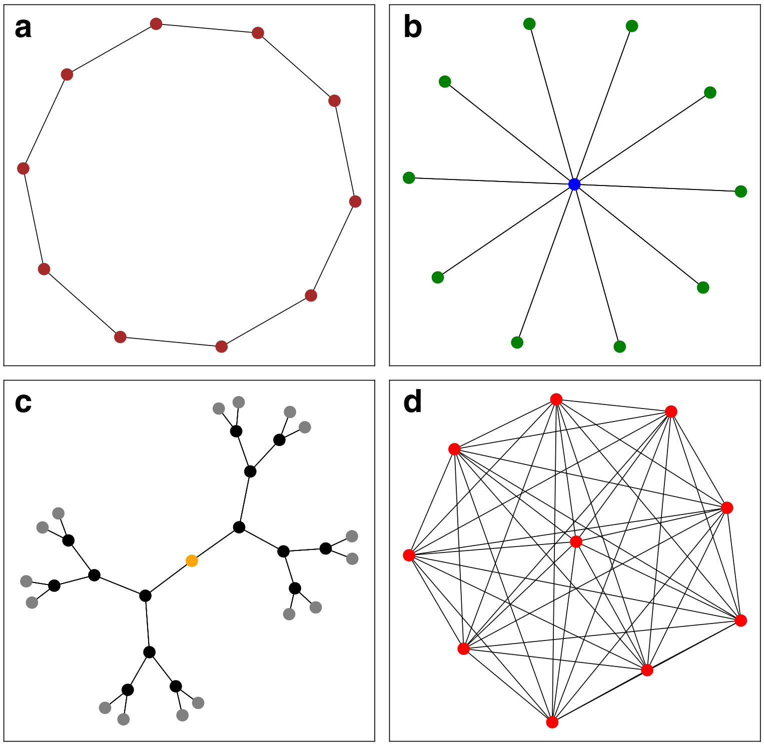

Fig. 4 Four network topologies. We used (**a**) a ring graph with ten nodes inside, (**b**) a star graph with ten peripheral nodes, (**c**) a complete binary tree graph with a height of four, and (**d**) a complete graph as a mesh graph. Each node role is colored based on its structural similarity

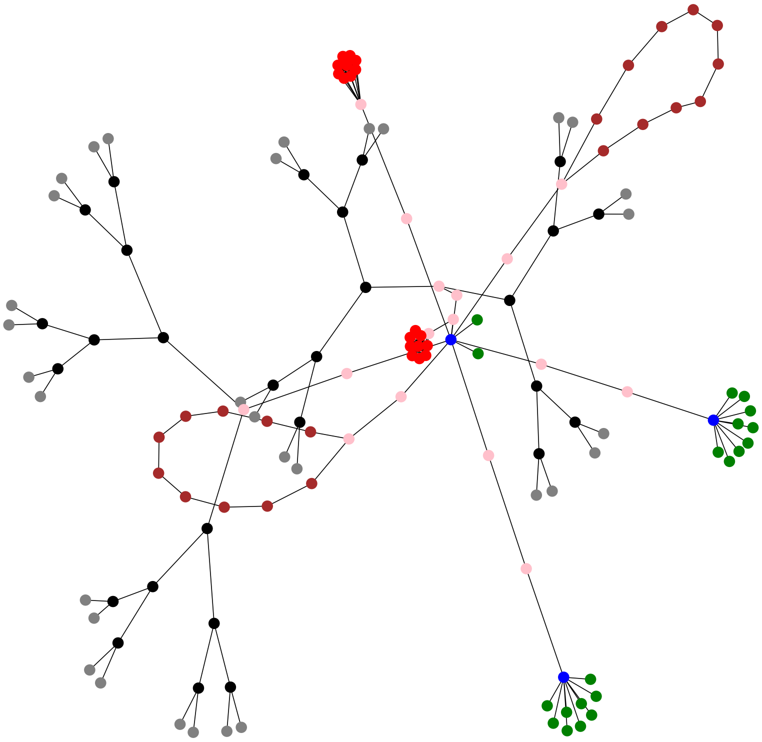

Fig. 5 Validation network. The star graph is placed at the center, and four network topologies are connected with the edge of the central star graph

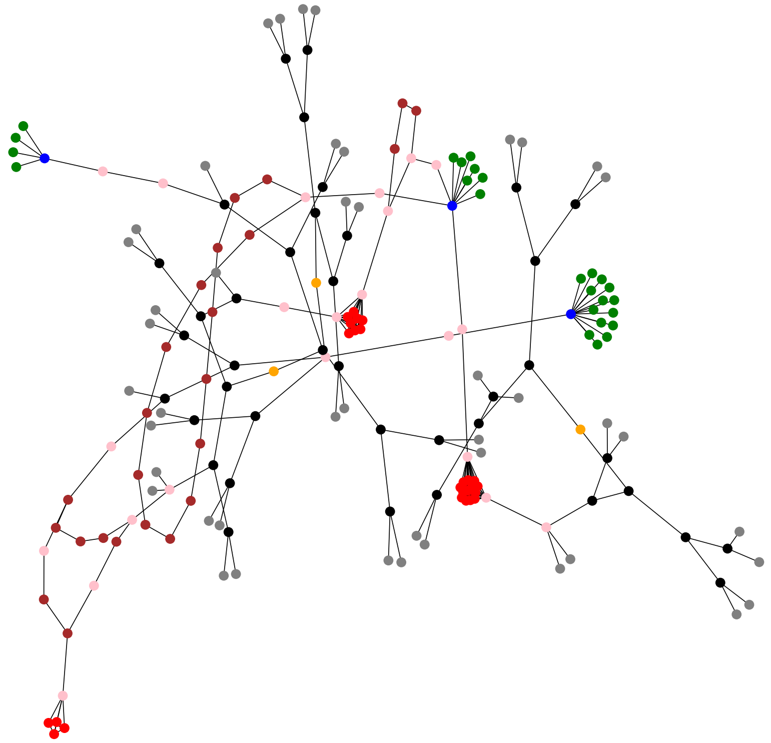

Fig. 6 Target network. Three sets of the four network topologies are randomly connected. Aside from tree graphs, the sizes of each graph are 5, 10, and 15

Moreover, our model selection method had nearly the same performance as the target tuned method. Therefore, our proposed method helped to select a better model on this dataset. In fact, the correlation coefficients in performance between the validation data and the target data were 0.970 (Macro-F1) and 0.965 (Micro-F1).

Operation of source network

We attempted to improve the Macro-F1 of the proposed method using the following strategy:

1 Create tens of copies of each network topology and generate node representations using struc2vec.
2 Abstract 70 node representations in such a way that the number of each role is 10.
3 Treat the abstracted node representations as the source network for domain adversarial learning.

Table 1 Classification results on a toy network

	Micro-F1	Macro-F1
RolX (Target tuned)	0.757	0.537
struc2vec (Target tuned)	0.865	0.731
Proposed method (Target tuned)	0.894	0.730
Proposed method (Model selection)	0.880	0.699

Target tuned means the model selected on the target labels, which is the upper bound performance each model can achieve, and model selection means the model selected with the validation network

This process allowed us to balance the classes on the source network without using the labels on the target network.

The classification results are shown in Table 2. Compared with the target-tuned models, our proposed method had the best performance in both metrics. Balancing the class ratio facilitated domain adversarial learning to improve performance even more. Furthermore, comparing Macro-F1 between Tables 1 and 2, our proposed method had higher performance. This indicates that the equal class ratio made the classification boundary unbiased, and the model classified more accurately those classes in which ratios were lower.

Zachary's karate club

Dataset

Zachary's Karate Club consists of 34 nodes and 78 links, where nodes represent members and links represent the connections between members outside the club, usually interpreted as friendship. The network is an unweighted and undirected network. Nodes 0 (the administrator "John A") and 33 (the instructor "Mr. Hi") are two leaders who led to a split of the club into two groups.

The source and validation networks were the same as those used in the last section, and we set the target network as Zachary's Karate Club. We selected the model that had the best accuracy on the validation network. Note that when the probability of the roles assigned on nodes was less than 40%, the nodes were considered to have no role in this experiment.

Results

The results of role discovery are shown in Fig. 7, and the pairs of roles and sets of nodes are given in Table 3.

First, for nodes assigned star-center, {0, 33} were the leaders of each group, and {1, 2, 3} and {23, 32} were located close to the leaders and played important roles in connecting many members. This indicates that the assignment of star-center roles was consistent with the background.

Next, for nodes assigned peripheral, 11 did not have links aside from the leader, and {14, 15, 17, 18, 20, 22} only connected the star-center nodes. The limitations on connectivity for the non-leaders indicate that these nodes are peripheral.

As mesh nodes, {7, 8, 13, 19, 27, 28, 30, 31} had many relationships regardless of groups and thus played many roles in bridging between the two groups. The high mediation these nodes exhibited showed that they worked as a mesh. By contrast, 9 did not seem to be correctly predicted because it connected the leader and another member. This mistaken prediction might be corrected when using networks more similar to the target.

Table 2 Classification results after operation

	Micro-F1	Macro-F1
RolX (Target tuned)	0.812	0.721
struc2vec (Target tuned)	0.773	0.703
Proposed method (Target tuned)	0.845	0.791
Proposed method (Model selection)	0.810	0.748

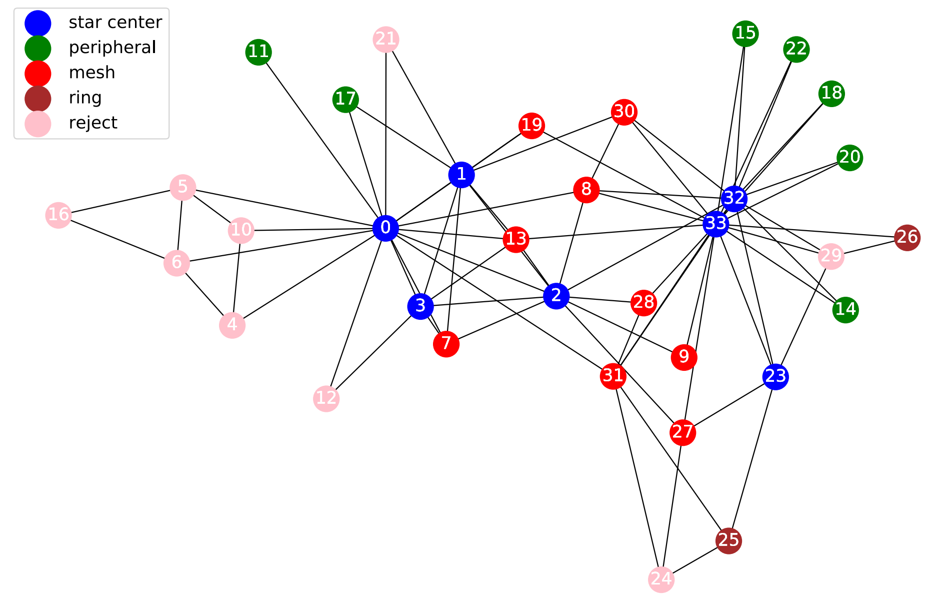

Fig. 7 Classification Results on Zachary's Karate Club(Zachary 1977)

Nodes assigned as ring nodes played roles of connecting nodes and did not have many links. Ring nodes simply connect two neighboring nodes, so this result was consistent with the actual function.

There were some nodes that did not exhibit any role in our proposed method. The two biggest probabilities of these nodes were almost the same, which implies that these nodes had two roles. For example, {4, 5, 6, 10} had star-center and inner nodes, which is consistent with the fact that they were close to the leader and formed a hierarchical structure. In addition, {12, 21} were assigned peripheral and ring roles. This seems reasonable in that they connected the leader to a node that was close to the leader.

To summarize, the experimental results of Zachary's Karate Club show that our proposed method can provide a meaningful interpretation.

Air-traffic network

We applied the proposed method to an air-traffic network to show an example of applying the proposed method to a real-world data network. Furthermore, we showed that choosing the validation network within the same field can be a good solution to the selection of the validation network.

Table 3 Classification results on Zachary's Karate Club (Zachary 1977)

Role	Sets of nodes
Star center	{0, 1, 2, 3, 23, 32, 33}
Peripheral	{11, 14, 15, 17, 18, 20, 22}
Mesh	{7, 8, 9, 13, 19, 27, 28, 30, 31}
Ring	{25, 26}
Reject	{4, 5, 6, 10, 12, 16, 21, 24, 29}

Table 4 Statistics of air-traffic networks

Dataset	American (US)	European (EU)	Brazilian (BR)
Number of nodes	1190	399	131
Number of links	13599	5995	1074
Average degree	22.93	30.05	16.40
Clustering coefficient	0.50	0.54	0.64
Average shortest path length	3.07	2.32	2.19

Dataset

We used the air traffic networks in the United States, Brazil, and the European Union, which are unweighted, undirected graphs. The nodes represent airports, and the links indicate that there is a commercial flight between two airports. Each node is labeled from (0,1,2,3) according to the frequency of airport usage. The airport usage of each network is determined as follows.

- Airport usage of Brazilian air-traffic network: sum of the departure and arrival flights from Jan. 2016 to Dec. 2016 collected by The National Civil Aviation Agency[1].
- Airport usage of American air-traffic network: number of users from Jan. 2016 to Oct. 2016 gathered by The Bureau of Transportation Statics[2].
- Airport usage of European air-traffic network: number of departure flights from Jan. 2016 to Nov. 2016 estimated by The Statistical Office of the European Union[3].

Although the airport usages are calculated using different criteria, the three network structures are comparable by smoothing the proportion of the nodes' labels to make each label in an individual network to account for 25% of the total. The statistics of the networks are shown in Table 4.

Note that the network size and density are fairly different, which increases the difficulty of transfer learning.

We assigned these three networks as the source network, the target network, and the validation network, so there are six combinations in total. We used the source network's label knowledge to predict the target network's labels by using the accuracy of the validation network to select the best-performing model.

Result

The results are shown in Table 5. First, compared with struc2vec (Target Tuned), RolX (Target Tuned) performed worse in all cases. Especially those cases when the Brazilian air-traffic network is the source network or the target network, the accuracy results were close to random. After confirming the result, the predicted labels were all the same, which proved that the RolX model fails when the distribution of the two networks are different.

Second, the proposed method (Target Tuned) performed better than struc2vec and RolX in all cases, which showed the advantage of the domain adversarial learning applied in the proposed method. Furthermore, for one particular target network, the proposed method had the smallest of the results' difference between different source networks, which indicates that the proposed method has the advantage of smaller reliance on the

[1] http://www.anac.gov.br/
[2] https://transtats.bts.gov/
[3] http://ec.europa.eu/

Table 5 Results of air-traffic networks

Method	T	BR		EU		US	
	S	EU	US	BR	US	BR	EU
RolX (Target tuned)		0.267	0.267	0.256	0.496	0.251	0.459
struc2vec (Target tuned)		0.588	0.710	0.551	0.521	0.582	0.508
Proposed method (Target tuned)		0.634	0.733	0.559	0.546	0.598	0.535
Proposed method (Model selection)		0.588	0.664	0.459	0.451	0.434	0.516
Target-only (Target tuned)		0.741		0.622		0.683	

S means the source network, and T means the target network; US, EU, and BR indicate the United States, European Union, and Brazil. The target-only method, which is the result of the model trained by the target network data, showed the difficulty of these tasks

source network and better versatility. In particular, for the case of US as the source network and BR as the target network, the result of the proposed method (Target Tuned) is fairly close to the target-only (Target Tuned)'s result, which indicates the high usefulness of the proposed method.

Finally, the proposed method(Model Selection) performed fairly close to struc2vec(Target Tuned), which indicates that even for the cases where the labels of the target network are not given, the proposed method demonstrates its effectiveness in practical uses.

Conclusion

In this paper, we proposed a general framework using transfer learning for inferring node roles more accurately. In this framework, we used node representation learning to make structurally similar nodes as close as possible in a pretrained manner. To overcome domain-shift problems, where node representations are slightly different between domains, we tailored domain adversarial learning for role discovery. Furthermore, we presented a validation network that led to a simple model selection without using labels on a target network.

A computational experiment on a barbell graph and a toy network showed that domain adversarial learning improved the accuracy of role discovery, that our proposed method outperformed conventional methods, and that domain-invariant representations led to high performance. In addition, we discovered with the experiment on the toy network and on Zachary's Karate Club that a model tuned on our proposed model selection with a validation network had comparable performance to a model tuned on a target network. Finally, we further explored whether our proposed method could achieve high accuracy in a comparison of real-world air-traffic networks. Consequently, we showed an empirical way of choosing the validation network in a practical role discovery problem.

As future work, we will pursue the possibilities of our framework by using different components. For example, we may use various node-embedding methods other than struc2vec. A comparison of our model selection method with conventional methods is also needed.

Acknowledgements
Not applicable.

Authors' contributions
SK organized research, built the framework, conducted experiments, and analyzed the results. FT and MN designed the work and analyzed the results, and SL summarized the results and helped draft the paper. TF provided the basic

conception and several references related to this work. TN and SU interpreted the data and partially designed the work. All authors read and approved the final manuscript.

Funding
The authors received no specific funding for this work.

Availability of data and materials
The datasets in this work can be found in the [Networkx] (https://networkx.github.io/documentation/stable/index.html) package in Python. The implementation codes are available in [Github] (https://github.com/Shumpei-Kikuta/Domain-invariant-Latent-Representation-Discovers-Roles).

Competing interests
The authors declare that they have no competing interests.

Author details
[1]University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, 113-8654 Tokyo, Japan. [2]TDAI Lab Co., Ltd., Tokyo, Japan. [3]Sansan Inc., Tokyo, Japan.

Received: 28 February 2020 Accepted: 7 July 2020
Published online: 04 August 2020

References

Bousmalis K, Silberman N, Dohan D, Erhan D, Krishnan D (2017) Unsupervised pixel-level domain adaptation with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 3722–3731. https://doi.org/10.1109/cvpr.2017.18
Brandes U, Lerner J (2010) Structural similarity: Spectral methods for relaxed blockmodeling. J Classif 27(3):279–306
Burt RS (1976) Positions in networks. Soc Forces 55(1):93–122
Cao B, Liu NN, Yang Q (2010) Transfer learning for collective link prediction in multiple heterogenous domains. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10). Citeseer. pp 159–166
Duan L, Xu D, Tsang I (2012) Learning with augmented features for heterogeneous domain adaptation. arXiv preprint arXiv:1206.4660
Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V (2016) Domain-adversarial training of neural networks. J Mach Learn Res 17(1):2096–2030
Gong C, He D, Tan X, Qin T, Wang L, Liu T-Y (2018) Frage: Frequency-agnostic word representation. In: Advances in Neural Information Processing Systems. pp 1334–1345
Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp 2672–2680
Henderson K, Gallagher B, Eliassi-Rad T, Tong H, Basu S, Akoglu L, Koutra D, Faloutsos C, Li L (2012) Rolx: structural role extraction & mining in large graphs. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM. pp 1231–1239
Henderson K, Gallagher B, Li L, Akoglu L, Eliassi-Rad T, Tong H, Faloutsos C (2011) It's who you know: graph mining using recursive structural features. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM. pp 663–671. https://doi.org/10.1145/2020408.2020512
Hoffman J, Tzeng E, Park T, Zhu J-Y, Isola P, Saenko K, Efros AA, Darrell T (2017) Cycada: Cycle-consistent adversarial domain adaptation. arXiv preprint arXiv:1711.03213
Kikuta S, Toriumi F, Nishiguchi M, Fukuma T, Nishida T, Usui S (2019) Domain-invariant latent representation discovers roles. In: International Conference on Complex Networks and Their Applications. Springer. pp 834–844
Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
Kohavi R, et al (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Ijcai, vol. 14, Montreal. pp 1137–1145
Long M, Cao Y, Wang J, Jordan MI (2015) Learning transferable features with deep adaptation networks. arXiv preprint arXiv:1502.02791
Long M, Zhu H, Wang J, Jordan MI (2016) Unsupervised domain adaptation with residual transfer networks. In: Advances in Neural Information Processing Systems. pp 136–144
Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp 3111–3119
Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784
Mu J, Bhat S, Viswanath P (2017) All-but-the-top: Simple and effective postprocessing for word representations. arXiv preprint arXiv:1702.01417
Nowicki K, Snijders TAB (2001) Estimation and prediction for stochastic blockstructures. J Am Stat Assoc 96(455):1077–1087
Pan SJ, Tsang IW, Kwok JT, Yang Q (2010) Domain adaptation via transfer component analysis. IEEE Trans Neural Netw 22(2):199–210
Ribeiro LF, Saverese PH, Figueiredo DR (2017) struc2vec: Learning node representations from structural identity. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM. pp 385–394. https://doi.org/10.1145/3097983.3098061
Rissanen J (1978) Modeling by shortest data description. Automatica 14(5):465–471
Rossi RA, Ahmed NK (2015) Role discovery in networks. IEEE Trans Knowl Data Eng 27(4):1112–1131
Rossi RA, Gallagher B, Neville J, Henderson K (2013) Modeling dynamic behavior in large evolving graphs. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining. ACM. pp 667–676. https://doi.org/10.1145/2433396.2433479

Russo P, Carlucci FM, Tommasi T, Caputo B (2018) From source to target and back: symmetric bi-directional adaptive gan. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 8099–8108. https://doi.org/10.1109/cvpr.2018.00845
Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
Sugiyama M, Krauledat M, MÃžller K-R (2007) Covariate shift adaptation by importance weighted cross validation. J Mach Learn Res 8(May):985–1005
Tu K, Cui P, Wang X, Yu PS, Zhu W (2018) Deep recursive network embedding with regular equivalence. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM. pp 2357–2366. https://doi.org/10.1145/3219819.3220068
White HC, Boorman SA, Breiger RL (1976) Social structure from multiple networks. i. blockmodels of roles and positions. Am J Sociol 81(4):730–780
Xie S, Zheng Z, Chen L, Chen C (2018) Learning semantic representations for unsupervised domain adaptation. In: International Conference on Machine Learning. pp 5419–5428
Zachary WW (1977) An information flow model for conflict and fission in small groups. J Anthropol Res 33(4):452–473
Zhong E, Fan W, Yang Q, Verscheure O, Ren J (2010) Cross validation framework to choose amongst models and datasets for transfer learning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer. pp 547–562. https://doi.org/10.1007/978-3-642-15939-8_35

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.