Domain Adaptation for Graph Classification

Zhangjie Cao, Hongyu Ren, Lantao Yu

Computer Science Department, Stanford University

Basic Setup

We use a joint data distribution $p(x, y)$ to define a domain, with which we also define both the marginals and the conditionals. Let $p_s(x_s, y_s)$ denote the underlying joint data distribution of the data instance x_s and the corresponding label y_s for source domain, and let $p_s(x_s)$ denote the marginal distribution of x_s. $p_t(x_t, y_t)$ and $p_t(x_t)$ are defined analogously for target domain.

In feature-based unsupervised domain adaptation, our objective is to train a classifier $f_{\phi,\theta} = h_\phi \circ g_\theta$ which can perform well on target domain. Specifically, we parametrize two modules:

1. Feature extractor $g_\theta : \mathcal{X} \to \mathcal{Z}$: a projection function from data space $\mathcal{X}$ to a common latent feature space $\mathcal{Z}$ shared by two domains.
2. Classification function $h_\phi : \mathcal{Z} \to \mathcal{P}(\mathcal{Y})$: a projection function from the latent feature space to the set of probability distributions over the label set $\mathcal{Y}$.

Adversarial Domain Adaptation

To address the covariate shift problem, many domain adaption methods are proposed to minimize the following objective:

$$\mathbb{E}_{(x_s,y_s)\sim p_s}\mathcal{L}_c(f_{\phi,\theta}(x_s), y_s) + \lambda d(q_s(z_s;\theta), q_t(z_t;\theta))$$

Here z_s and z_t are latent representations for source and target domain; q_s and q_t are the marginal distributions of z_s and z_t, which is implicitly defined by the marginals $p_s(x_s), p_t(x_t)$ and the deterministic mapping g_θ; $\mathcal{L}_c$ is the cross entropy loss for training a classifier; $d(\cdot,\cdot)$ is some divergence or distance measure between two distributions and λ is the weighting factor.

Typically, the Jensen-Shannon divergence between q_s and q_t is minimized within an adversarial learning framework:

$$\min_\theta \max_\omega \mathbb{E}_{x_s\sim p_s} \log D_\omega(g_\theta(x_s)) + \mathbb{E}_{x_t\sim p_t} \log(1 - D_\omega(g_\theta(x_t)))$$

where D_ω is a domain classifier on the representation space. At optimality, the marginal distributions of latent representations will be matched and the learned representations will be *domain-invariant*.

Graph Classification

We use graph neural networks to do graph classification. Given an input graph, we aim to output the label of the graph, e.g. whether a chemical is toxic or not.

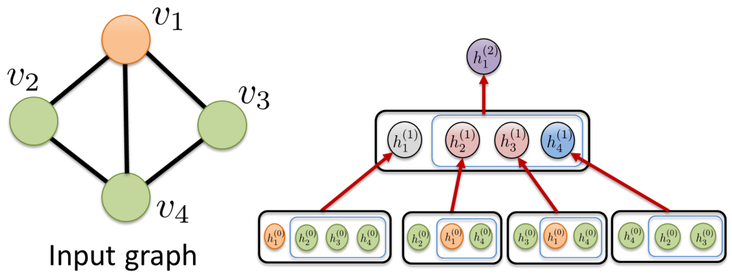
Graph Neural Networks

1. Iteratively aggregate information from neighbours to learn/update node representation.

$$a_v^{(k)} = \text{AGGREGATE}^{(k)}(\{h_u^{(k-1)} : u \in \mathcal{N}(v)\})$$
$$h_v^{(k)} = \text{COMBINE}^{(k)}(h_v^{(k-1)}, a_v^{(k)})$$

2. Globally pool node representation

$$h_G = \text{READOUT}(\{h_v^{(K)} | v \in G\})$$



Input graph

Examples of AGGREGATION

- GraphSAGE

$$h_v^{(k)} = W^{(k)} \cdot \text{CONCAT}(\text{MAX}(\{\text{ReLU}(W \cdot h_u^{(k-1)}), \forall u \in \mathcal{N}(v)), h_v^{(k-1)})$$

- GCN

$$h_v^{(k)} = \text{ReLU}(W \cdot \text{MEAN}(\{h_u^{(k-1)}, \forall u \in \mathcal{N}(v) \cup \{v\}\}))$$

- GIN

$$h_v^{(k)} = \text{MLP}^{(k)}((1 + \epsilon^{(k)}) \cdot h_v^{(k-1)} + \sum_{u\in\mathcal{N}(v)} h_u^{(k-1)})$$

Examples of READOUT Literature use MEAN, SUM or Attention modules to globally pool the node representation to achieve the graph representation.

Main Idea of This Work

To enable graph domain adaptation, we adopt a new adaptation structure for graph. In the graph neural network, we exploit both the node and graph features. We adapt the source and the target graphs on both node and graph levels with adversarial domain adaptation. We use two adversarial networks, one for the node feature and the other for the graph feature. The node adversarial network can draw the distribution of node features closer, which serves as a basic for the graph feature distribution alignment. The node-level adaptation aims to match the nodes in graphs between the source and target domains while the graph-level adaptation aims to match the graph structure of the source and target domains.

Experimental Settings

Dataset

We validate our model on 8 binary classification datasets in MoleculeNet, a recently curated benchmark for molecular property prediction. Please find the statistics in Table 1.

Dataset	BBBP	Tox21	ToxCast	SIDER	ClinTox	MUV	HIV	BACE
# Molecules	2039	7831	8575	1427	1478	93087	41127	1513
# Prediction tasks	1	12	617	27	2	17	1	1

Table: Statistics of the moleculenet.

In order to create data from two domains, instead of random splitting the dataset, we use **scaffold split** in chemical graph classification experiments, which clusters the molecules according to the scaffold (molecular substructure). Then we randomly select several clusters to be training domain and the remaining to be test domain. It also simulates the real-world use case, since different labs may focus on different types of scaffold, but all labs have same graph classification tasks, for example they aim to predict whether a molecule graph is toxic or not.

Evaluation Metric We use ROC-AUC score and average precision as our evaluation metric. We compare the results on both the source and the target domains.

Results

Method	ROC-AUC		AP	
	Source	Target	Source	Target
Source Only	0.9890	0.7818	0.9296	0.4103
Adversarial	0.8972	0.7678	0.5657	0.3766
Entropy	0.9892	0.7808	0.9266	0.4152
Ours	0.9892	0.7828	0.9422	0.4252

Table: ROC-AUC and Average Precision (AP) of different methods for source and target data

We can observe that naively applying adversarial domain adaptation to the graph (Adversarial) performs even worse than directly using the model trained on the source domain (Source Only). Also, a widely-used technique, entropy minimization, does not improve the model trained on the source much. The above results demonstrate that naively combining domain adaptation methods with graph neural network cannot address graph domain adaptation. Our node-level and graph-level co-adaptation model performs better than all the baselines, which proves that the importance of a newly designed adaptation structure for graph.

Mail: {caozj18,hyren,lantaoyu}@stanford.edu