Modern Topics on Statistical Learning Theory 2023 Spring

Lecture 9 — Domain Adaptation

Prof. Qi Lei *Scribe: Ying Wang, Akash Gupta, Yunlei Lu*

1 Introduction

In this lecture, we will focus on concepts and methodologies for domain adaptation. Domain adaptation is a field in machine learning where the model is trained on the source dataset and then applied to the target dataset different from the source dataset.

From the notations in the previous lecture, let $D^S = \{(x_i^S, y_i^S)_{i=1}^{n_S}\}$ denote the source dataset, where (x_i^S, y_i^S) are drawn from the distribution $\mathbb{P}_{X,Y}^S$. The model is trained where the goal is to minimize the loss on the source dataset, i.e., find θ^* such that the model outputs $f_\theta(x_i^S)$ minimizes the loss

$$\theta^* = \arg\min_\theta \frac{1}{n_S} \sum_{i=1}^{n_S} l(f_\theta(x_i^S), y_i^S). \tag{1}$$

The question is that whether the trained model performs well on the target dataset, i.e., the loss $\mathcal{L}_T(\theta)$ on the target dataset D^T is small, where

$$\mathcal{L}_T(\theta) = \mathbb{E}_{(x^T, y^T) \sim \mathbb{P}_{X,Y}^T} \left[l(f_\theta(x_i^S), y_i^S) \right]. \tag{2}$$

Based on the information of the target dataset D^T, we have the following different types of domain adaptation:

- Unsupervised domain adaptation: The target dataset is unlabeled, $D^T = \{(x_i^T)_{i=1}^{n_T}\}$.
- Semi-supervised domain adaptation: The target dataset contains both unlabeled data and a limited set of labeled data, $D^T = \{(x_i^T)_{i=1}^{n_T}\} \cup \{(\tilde{x}_i, \tilde{y}_i)_{i=1}^{\tilde{n}_T}\}$.
- Supervised domain adaptation: All data in the target dataset is labeled.
- Domain/OOD Generalization: No data from the target domain. Learn distributional-robust model (no adaption anymore) when we have multiple source tasks.

This is in contrast to traditional supervised learning, where the data in the source and target dataset are sampled from the same distribution, i.e., $\mathbb{P}^S = \mathbb{P}^T$.

2 Domain Adaptation

Domain adaptation refers to the problem of adapting a predictive model learned from one domain to make accurate predictions on a different, but related, domain. Let's assume we have a source

domain $\mathcal{D}_s$ and a target domain $\mathcal{D}_t$ with corresponding input spaces $\mathcal{X}_s$ and $\mathcal{X}_t$ and output spaces $\mathcal{Y}_s$ and $\mathcal{Y}_t$. We have a labeled training set $\mathcal{D}s = (x_i, y_i)_{i=1}^{n_s}$ from the source domain and an unlabeled test set $\mathcal{D}_t = (x_j)_{j=1}^{n_t}$ from the target domain. The goal is to learn a predictor $f : \mathcal{X}_t \to \mathcal{Y}_t$ that performs well on the target domain.

One approach to domain adaptation is to use a regularized objective function that encourages the model to learn features that are invariant to changes in the input distribution. A popular example is the Maximum Mean Discrepancy (MMD) regularizer:

$$min_f \frac{1}{n_t} \sum_{j=1}^{n_t} l(f(x_j), y_j) + \lambda \, MMD(p_t, p_s)$$

where ℓ is a loss function, λ is a regularization parameter, p_s and p_t are the probability distributions of the source and target domains respectively, and $\text{MMD}(p_t, p_s)$ is the maximum mean discrepancy between the two distributions. We define other such examples in Section 4.

Another approach to domain adaptation is transfer learning, which involves leveraging knowledge learned from a source domain to improve performance on a target domain. One popular technique is to use pre-trained models, such as convolutional neural networks (CNNs), to extract features from the input data and fine-tune them on the target domain. This can be done by adding new layers to the pre-trained model and training them on the target domain.

Below we define the problem of domain adaptation more formally,

Goal: from $D^S + D^T$: learn a f_θ s.t. $\mathbb{E}\, l(f_\theta(x), y)$ is small.

Note that the problem is ill-defined/impossible when no assumption connecting P^S and P^T.

Setting 1: covariance shift $P^S_{Y|X} = P^T_{Y|X}$, but $P^S_X \neq P^T_x$

Setting 2: model/concept shift $P^S_{Y|X} \neq P^T_{Y|X}$ but they are "close" (Example: $\min_{\theta\in\Theta} \mathbb{E}_{P^S}\, l(f_\theta(x), y) + \mathbb{E}_{P^T}(f_\theta(x), y) =: \lambda^*_\Theta$ is small, or essentially $P_T(y|x) \approx P_S(y|x) \forall x, y$

Setting 3: label shift $P^S_Y \neq P^T_Y$ (or $P^S_{Y|X} \neq P^T_{Y|X}$) but $P^S_{X|Y} = P^T_{X|Y}$

3 Domain generalization

Another parallel problem similar to domain adaptation or DA is domain generalization.

Domain generalization is a related problem to domain adaptation, but with the goal of learning a model that can generalize to new, unseen domains. Let's assume we have k different domains, $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$, each with its own input space $\mathcal{X}_i$ and output space $\mathcal{Y}_i$. We have labeled training sets $\mathcal{D}_i = (x_{i,j}, y_{i,j})_{j=1}^{n_i}$ from each domain, and we want to learn a predictor $f : \mathcal{X} \to \mathcal{Y}$ that performs well on all domains.

One approach to domain generalization is to use a regularized objective function that encourages the model to learn features that are invariant to changes in the input distribution across domains. One example is the Group Distribution Discrepancy (GDD) regularizer:

$$min_f \frac{1}{k} \sum_{i=1}^{k} \frac{1}{n_t} \sum_{j=1}^{n_t} l(f(x_{i,j}), y_j) + \lambda \, GDD(\{p_i\}_{i=1}^k)$$

where ℓ is a loss function, λ is a regularization parameter, p_i is the probability distribution of the ith domain, and $\text{GDD}(\{p_i\}_{i=1}^k)$ is the group distribution discrepancy between the k domains.

Another approach to domain generalization is to use a model that can learn shared representations across domains, such as deep neural networks. One popular technique is to use a Siamese network architecture, which consists of two or more identical subnetworks that share weights. The input data from each domain is passed through its own subnetwork, and the outputs are then concatenated and passed through a final output layer.

To evaluate the performance of a domain generalization model, we can use the notion of the generalization error, which measures the expected loss of the model on new, unseen data from the same domain as the training data. We can define the generalization error as:

$$GenError(f) = \mathbb{E}_{(x,y)\ p(x,y)}[l(f(x)), y]$$

where $p(x, y)$ is the joint distribution of the input-output pairs.

To analyze the generalization error of a model, we can use techniques such as the Rademacher complexity and the VC dimension. The Rademacher complexity measures the ability of the model to fit random noise in the data, while the VC dimension measures the complexity of the model class and its ability to fit different functions.

4 Related Work

• [Ben-David 2006] [1]

$$L_T(\theta) : \mathop{\mathbb{E}}_{(x,y)\sim P^T} l(f_\theta(x), y) \leq L_S(\theta) + d(P^T_{X,Y}, P^S_{X,Y}) + \lambda^*_\Theta \tag{3}$$

• [Kifer et al. 2004] [3] or [Redko et al. 2017] [4]: W_1 distance

We can further replace x with $h(x) \Rightarrow P^S_{h(X),Y} \approx P^T_{h(X),Y}$.

Potentially, one can reduce the distance of $d\left(\mathbb{P}^T_{h(X),Y}, \mathbb{P}^S_{h(X),Y}\right)$, i.e., find an invariant representation.

$$\min_{f_\theta := w \circ h} L_S(f_\theta) + \lambda d(P^T_{h(X)}, P^T_{h(X)}) \tag{4}$$

• [Ganin & Lempitsky] [2].

Make the distribution of $h(x), x \sim P^S_X$, and $h(x), x \sim P^T_X$ to be indistinguishable. We hope the best classifier performs badly on the domain classification task, $(h(x_i)^S, 1)\&(h(x_j)^T, 0)$.

$$\min_{w,h} L_S(f_\theta(f_\theta) - \lambda L(optimal\ domain\ classifier\ on\ h(x)) \tag{5}$$

If $P_S(h(x)) \approx P_T(h(x))$, the best model will be similar to random guess.

MMD-based distance (maximum mean discrepancy) $h(X) =: \tilde{X}$.

$$d_{MMD}(P^S_{\tilde{X}}, P^T_{\tilde{X}}) = ||\frac{1}{n_S}\sum_{i=1}^{n_S}\phi(\tilde{x_i}^S) - \frac{1}{n_S}\sum_{i=1}^{n_T}\phi(\tilde{x_i}^T)|| \tag{6}$$

Example: $\phi(\tilde{x_i}) = \tilde{x_i}\tilde{x_i}^T$

• [Tzeng etal 2014] [5].

$$\min_{w,h} L_S(w \circ h) + \lambda d^2_{MMD}(h(X^T), h(X^S)) \tag{7}$$

Wasserstein distance (Kantorovich duality)

$$min_{w,h} L_S(w \circ h) + \lambda W_1(P^S_{h(x)}, P^T_{h(x)})$$
$$where\ W_1(P^S_{h(x)}, P^T_{h(x)}) = max \mathbb{E}_{D:1-lipX^S \sim \mathbb{P}^S_X}[D(h(x^S))] - \mathbb{E}_{X^T \sim \mathbb{P}^T_X}[D(h(x^T))]$$

• [Zhu etal 2017] [6]: Cycle GAN

$$L_{GAN,S\to T}(X^S, X^T, G_{S\to T}, D^T) = \mathop{\mathbb{E}}_{X^T}[logD^T(X)] + \mathop{\mathbb{E}}_{X^S}[log(1 - D^T(G_{S\to T}(X^S))]$$
$$L_{GAN,T\to S}(X^S, X^T, G_{T\to S}, D^S) = \mathop{\mathbb{E}}_{X^S}[logD^S(X)] + \mathop{\mathbb{E}}_{X^T}[log(1 - D^S(G_{T\to S}(X^T))]$$
$$L_{cycle}(X^S, X^T, G_{S\to T}, G_{T\to S}) = \mathbb{E}(||G_{T\to S}(G_{S\to T}(X^S)) - X^S||) + \mathbb{E}(||G_{S\to T}(G_{T\to S}(X^T)) - X^T||)$$

Then, the objective is

$$\min_{f_\theta} L_S(f_\theta) + \lambda(L_{GAN,S\to T} + L_{GAN,T\to S} + L_{cycle})$$

and the prediction is

$$f_\theta(G_{T\to S}(X^T))$$

Note that

$$L_T(w \circ h) \le L_S(\theta) + d(h(X^S), h(X^T)) + \lambda^*_\Theta$$

where $\lambda^*_\Theta = arg\min_w L_S(w \circ h) + L_T(w \circ h)$. Previous methods minimize the first two terms, by choosing h, but they potentially make the λ^* explode. For example, h is a good classifier on the source: $h : X \to [0, 1]$ and w is an identity function, not necessarily making sure h is still expressive enough.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

[2] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[3] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *VLDB*, volume 4, pages 180–191. Toronto, Canada, 2004.

[4] Ievgen Redko, Amaury Habrard, and Marc Sebban. Theoretical analysis of domain adaptation with optimal transport. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part II 10*, pages 737–753. Springer, 2017.

[5] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[6] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.