Deep Generative and Discriminative Domain Adaptation

Extended Abstract

Han Zhao
Carnegie Mellon University
han.zhao@cs.cmu.edu

Junjie Hu
Carnegie Mellon University
junjieh@cs.cmu.edu

Zhenyao Zhu
Google
zhuzychn@gmail.com

Adam Coates
Apple
acoates@cs.stanford.edu

Geoff Gordon
Carnegie Mellon University &
Microsoft Research Montreal
ggordon@cs.cmu.edu

ABSTRACT

The ability to adapt to and learn from different domains and environments is crucial for agents to generalize. In this paper we propose a probabilistic framework for domain adaptation that blends both generative and discriminative modeling in a principled way. Under this framework, generative and discriminative models correspond to specific choices of the prior over parameters. By maximizing both the marginal and the conditional log-likelihoods, our models can use both labeled instances from the source domain as well as unlabeled instances from *both* source and target domains. We show that the popular reconstruction loss of autoencoder corresponds to an upper bound of the negative marginal log-likelihoods of unlabeled instances, and give a generalization bound that explicitly incorporates it into the analysis. We instantiate our framework using neural networks, and build a concrete model, DAuto.

KEYWORDS

Deep Learning; Adversarial Machine Learning; Domain Adaptation; Generative Models; Discriminative Models

ACM Reference Format:
Han Zhao, Junjie Hu, Zhenyao Zhu, Adam Coates, and Geoff Gordon. 2019. Deep Generative and Discriminative Domain Adaptation. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

The ability to be able to adapt to and learn from different domains and environments is crucial for agents to generalize. However, making accurate predictions relies heavily on the existence of labeled data for the desired tasks. On the other hand, generating labeled data for new learning tasks is often time-consuming. As a result, this poses an obstacle for applying machine learning methods to broader application domains. *Domain adaptation* focuses on the situation where we only have access to labeled data from source domain, which is assumed to be different from, but related to the target domain we want to apply our model to. The goal of domain adaptation algorithms under this setting is to generalize better in the target domain by exploiting labeled data in the source domain and unlabeled data in the target domain.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

In this paper we propose a probabilistic framework for domain adaptation that combines both generative and discriminative modeling in a principled way. We start from a simple yet general generative model, and show that a special choice on the prior distribution of model parameters leads to the usual discriminative modeling. This provides us a general way to interpolate between generative and discriminative extremes through different choices of priors. Due to the generative nature, the framework provides us a principled way to use unlabeled instances from both the source and the target domains. Under this framework, if we use non-parametric kernel density estimators for the marginal distribution over instances, we can show that the popular reconstruction loss of autoencoders corresponds to an upper bound of the negative marginal log-likelihoods of unlabeled instances. This provides us a novel probabilistic interpretation on why unsupervised training with general autoencoders may help with discriminative tasks. Theoretically, we provide a generalization bound that incorporates the reconstruction loss of autoencoders into analysis, showing that the reconstruction loss can be used as a data-dependent measure that characterizes the complexity of the dataset. From this perspective, our interpretation may also be used to explain the recent success of autoencoders in semi-supervised learning [5]. To the best of our knowledge, this is the first probabilistic interpretation for general autoencoders, though interpretations exist for specific variants, e.g., denoising autoencoders [7] and contractive autoencoders [6].

To better understand how the proposed model works in practice, we instantiate our framework with flexible neural networks, which are powerful function approximators, leading to a concrete model, *DAuto*. DAuto is designed to achieve the following three objectives simultaneously in a unified model:

(1) It learns representations that are informative for the main learning task in the source domain.
(2) It learns domain-invariant features that are indistinguishable between the source and the target domains.
(3) It learns robust representations under reconstruction loss for instances in both domains.

Under mild assumptions, we also provide a theoretical analysis for DAuto that explains why these three objectives are necessary in order to achieve a small generalization error.

2 THE MODEL

Let $\mathbf{x} \in \mathbb{R}^d$ be an input instance and y be its target variable: $y \in \{0, 1\}$ in the classification setting or $y \in \mathbb{R}$ in the regression

setting. A fully generative model can be specified as $p(\mathbf{x}, y; \phi, \psi) = p(\mathbf{x}; \phi)p(y \mid \mathbf{x}; \psi)p(\phi, \psi)$, where ϕ is the model parameter that governs the generation process of $\mathbf{x}$; ψ is the model parameter for the conditional distribution $y \mid \mathbf{x}$, and $p(\phi, \psi)$ is the prior distribution over both model parameters. Using the above joint model, if we assume that the prior distribution $p(\phi, \psi)$ factorizes as $p(\phi, \psi) = p(\phi)p(\psi)$, then we will have $\max_{\phi,\psi} p(\mathbf{x}; \phi)p(y \mid \mathbf{x}; \psi)p(\phi)p(\psi) = \max_{\psi} p(y \mid \mathbf{x}; \psi)p(\psi) \cdot \max_{\phi} p(\mathbf{x}; \phi)p(\phi)$. Note that in this case only the first term in R.H.S. is concerned with the prediction, which means unsupervised learning on $p(\mathbf{x}; \phi)p(\phi)$ does not help generalization. In other words, the independence assumption between ϕ and ψ equivalently reduces our joint model over both $\mathbf{x}$ and y into discriminative models that only contain parameters ψ if we only care about prediction accuracy. On the other extreme, if we have $\phi = \psi$, then this corresponds to having a prior $p(\phi, \psi) = p_0(\phi, \psi)\delta(\phi - \psi)$ that constrains ϕ and ψ to be shared in both generative processes: $\max_{\phi,\psi} p(\mathbf{x}; \phi)p(y \mid \mathbf{x}; \psi)p_0(\phi, \psi)\delta(\phi - \psi) = \max_{\phi} p(\mathbf{x}; \phi)p(y \mid \mathbf{x}; \phi)p_0(\phi)$, where p_0 is a base distribution and $\delta(\cdot)$ denotes the Kronecker delta function. It can be seen that when $\phi = \psi$, the formulation exactly reduces to the usual MAP inference criterion over both $\mathbf{x}$ and y.

The discussion shows that depending on the choice of the prior distribution over ϕ and ψ, we can easily recover both discriminative and generative modelings. In practice the sweet spot often lies in a mix of both models [4]: discriminative training usually wins at predictive accuracy, while generative modeling provides a principled way to use unlabeled data. To achieve the best of both world, let us consider the case where ϕ and ψ have a common subspace, i.e., some model parameters are shared in both the generation process of $\mathbf{x}$ and $y \mid \mathbf{x}$. To make our discussion concrete, think if we have $p(\mathbf{x}; \phi) = g'(f(\mathbf{x}; \zeta); \phi\backslash\zeta)$ and $p(y \mid \mathbf{x}; \psi) = h(f(\mathbf{x}; \zeta), y; \psi\backslash\zeta)$, where ζ are the shared parameters of both ϕ and ψ. Domain adaptation is possible under this setting whenever $f(\cdot; \zeta)$ forms a rich class of transformations so that unlabeled instances from both domains have similar induced marginal distribution. As a generative model, it also allows algorithms to use unlabeled instances from both domains to optimize the marginal likelihood function $p(\mathbf{x}; \phi)$, which also helps the predictive task $p(y \mid \mathbf{x}; \psi)$.

Now we use our probabilistic framework and instantiate it with proper choices of both the marginal distributions as well as the conditional distributions. To this end, we propose to use nonparametric kernel density estimator (KDE) to model $p(\mathbf{x}; \phi)$. Specifically, let $K(\cdot)$ be the chosen kernel and $\{\mathbf{x}_i\}_{i=1}^n$ be a set of unlabeled instances. Our KDE for $p(\mathbf{x}; \phi)$ is given by:

$$p(\mathbf{x}; \phi) \propto \frac{1}{nw} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - g(f(\mathbf{x}_i; \zeta); \phi\backslash\zeta)}{w}\right) \tag{1}$$

where $w > 0$ is the bandwidth and $f : \mathbb{R}^d \to \mathbb{R}^D$ and $g : \mathbb{R}^D \to \mathbb{R}^d$ are two feature transformations. Our definition of KDE differs from the original one [8] by the additional parametric transformations $g \circ f$ applied to $\mathbf{x}$, and when $g \circ f = I$, our definition reduces to the original definition.

For the conditional distribution $y \mid \mathbf{x}$, depending on whether $y \in \mathbb{R}$ or $y \in \{0, 1\}$, typical choices include linear regression or logistic regression. To make the model more expressive, we can first augment them with nonlinear transformation f applied to the input instance. The transformation f is shared between both $p(\mathbf{x}; \phi)$ and $p(y \mid \mathbf{x}; \psi)$. Our model is completed by specifying the prior distribution as $p(\phi, \psi) = p_0(\phi, \psi)\delta(\phi(\zeta) - \psi(\zeta))$. The $\delta(\cdot)$ constrains the common parameter ζ to be shared by both $p(\mathbf{x}; \phi)$ and $p(y \mid \mathbf{x}; \psi)$. The base distribution $p_0(\phi, \psi)$ can be chosen as a flat (possibly improper) prior, which corresponds to the usual MLE criterion; or other forms of distributions that effectively introduce regularizations on both ϕ and ψ. Putting all together, maximizing a combination of conditional and marginal likelihoods correspond to the following unconstrained minimization problem:

$$\max_{\psi,\phi} \sum_{i=1}^{m} \log p(y_i \mid \mathbf{x}_i; \psi) - \lambda \sum_{j=1}^{n} ||\mathbf{x}_j - g(f(\mathbf{x}_j; \zeta); \phi\backslash\zeta)||_2^2$$

To instantiate our framework, in this work we consider neural networks as flexible function approximators for our desired transformations f and g. Specifically, we use fully-connected neural networks to parametrize f and g and softmax function to parametrize h. If $y \in \mathbb{R}$, we can simply change the softmax function to be an affine function as the output. For the simplicity of discussion, assume we only use a one-layer fully connected network to represent f and g: $f(\mathbf{x}) = \sigma(W_f\mathbf{x})$ and $g(\mathbf{z}) = W_g\mathbf{z}$, where $W_f \in \mathbb{R}^{D\times d}$, $W_g \in \mathbb{R}^{d\times D}$ and $\sigma(\cdot)$ is an element-wise nonlinear activation function. Let $h(\mathbf{z}) = \text{softmax}(W_h\mathbf{z})$ be the softmax layer to compute the conditional probability of class assignment.

Although our model has the capacity to learn the shared transformation f under which unlabeled data from both domains have similar marginal distributions, the objective function discussed so far does not necessarily induce such a transformation. For the purpose of domain adaptation, it is necessary to add a regularizer that enforces this constraint. One popular and effective choice is the $\mathcal{H}$-divergence introduced by [1–3]. It can be shown that the $\mathcal{H}$-divergence can be approximated by the binary classification error of the domain classifier that discriminates instances from the source or the target domain [2]. The intuition here is: given a fixed class of binary labeling functions, if there exists a function that is easy to tell instances in the source domain from those in the target domain, then the distance between these two domains is large. Let $\tilde{h} = \text{softmax}(W_d\mathbf{z})$ be the domain classifier where $\mathbf{z} = \sigma(W_f\mathbf{x})$ is the shared representation constructed by encoder f. The regularizer takes the form as a convex surrogate loss for the binary 0-1 error. Putting all together, the optimization problem of our joint model can be formulated as follows:

$$\begin{aligned}\min_{W_f, W_g, W_h} \max_{W_d} \quad & \sum_{i=1}^{m} \mathcal{L}_y(\mathbf{x}_i, y_i; W_f, W_h) \\ & + \lambda \sum_{j=1}^{n} \mathcal{L}_r(\mathbf{x}_j; W_f, W_g) - \mu \sum_{j=1}^{n} \mathcal{L}_d(\mathbf{x}_j; W_f, W_d)\end{aligned} \tag{2}$$

where $\mathcal{L}_y(\cdot, \cdot)$ is the prediction loss, $\mathcal{L}_r(\cdot)$ is the reconstruction loss and $\mathcal{L}_d(\cdot)$ is the domain classification loss. As a result, DAuto is designed to achieve the following three objectives simultaneously in a unified framework: 1). It learns representations that are informative for the main learning task in the source domain. 2). It learns robust representations under reconstruction loss. 3). It learns domain-invariant features that are indistinguishable between the source and the target domains.

REFERENCES

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79, 1-2 (2010), 151–175.

[2] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. 2007. Analysis of representations for domain adaptation. *Advances in neural information processing systems* 19 (2007), 137.

[3] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. 2004. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 180–191.

[4] Andrew Y Ng and Michael I Jordan. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems* 2 (2002), 841–848.

[5] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. 2015. Semi-supervised learning with ladder networks. In *NIPS*. 3546–3554.

[6] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. 2011. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 833–840.

[7] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*. ACM, 1096–1103.

[8] Larry Wassermann. 2006. All of nonparametric statistics. (2006).