1 Overview

1.1 Domain Adaptation Problem

The generic learning problem is: Devise a learning algorithm $\mathcal{A}$ such that given an empirical distribution $\bar{P}^A_{X,Y}$ drawn iid from true distribution $P^A_{X,Y}$, $\mathcal{A}$ has the "generalizes well" property:

$$\mathcal{A}(\bar{P}^A_{X,Y}) \mapsto f(\cdot) \text{ such that } f(\cdot) = \text{argmin}_{f\in\mathcal{F}}\, E_{X,Y\sim P^B_{X,Y}}[L(f(X),Y)] \qquad \text{(generalizes well)} \tag{1}$$

for some loss function $L(\cdot,\cdot)$, function class $\mathcal{F}$, and distribution $P^B_{X,Y}$. In the typical "stationary" learning scenario, $P^A_{X,Y} = P^B_{X,Y}$. In the *domain adaptation* scenario, this equality does not hold. Instead, $P^B_{X,Y} = P^B_X P^A_{Y|X}$, for $P^B_X \neq P^A_X$. In return for this inconvenience, we are also given empirical distribution $\bar{P}^B_X$. We also assume we are armed with some algorithm $\mathcal{A}$ that is assumed to perform well in the "stationary" learning scenario where $P^A_{X,Y} = P^B_{X,Y}$.

1.2 Projection-based Methods

These methods reduce the domain adaptation scenario to the stationary scenario by finding a transformation $\phi(\cdot)$ such that $P^A_{\phi(X),Y} = P^B_{\phi(X),Y}$. Then, $E_{X,Y\sim P^B_{\phi(X),Y}}[L(f(\phi(X)),Y)]$ should be low, where $f = \mathcal{A}(\hat{P}^A_{\phi(X),Y})$, due to our assumptions on $\mathcal{A}$ generalizing well in the stationary scenario. For $P^A_{\phi(X),Y} = P^B_{\phi(X),Y}$ to hold, it is *sufficient* ϕ satisfies:

$$P^A_{\phi(X)} = P^B_{\phi(X)} \qquad \text{(feature distribution similarity)} \tag{2}$$

$$Y \perp\!\!\!\perp X|\phi(X) \;\text{ where }\; (X,Y) \sim P^A_{X,Y} \qquad \text{(sufficient subspace)} \tag{3}$$

Most methods only check that ϕ satisfies Condition 2. Few methods actually check for Condition 3, which says that the projected space is (all that is) useful for predicting Y. Note that given a ϕ one can only *estimate* whether these conditions hold given the available $\bar{P}^A_{X,Y}, \bar{P}^B_X$.

Instead of finding ϕ to take us to the stationary scenario so that $\mathcal{A}$ generalizes well, why not directly find ϕ such that $\mathcal{A}$ generalizes well? Firstly, we do not necessarily need stationarity for good generalization, so that enforcing it is perhaps an unnecessary constraint. Secondly, to achieve stationarity, one needs to choose between methods for finding subspaces satisfying the 2 conditions, checking whether assumptions of those methods hold, and probably end up having to choose a trade-off parameter balancing the 2 conditions. Thirdly, such methods are not tailored to any specific downstream predictive method. We can bypass all these issues with a direct approach.

1.3 Contribution of This Work

We propose a novel, direct formulation to handle the domain adaptation problem, jointly learning a projection and predictive function by performing empirical risk minimization that relies on an unbiased estimate of in-sample test loss.

2 Formulation

2.1 Assumptions

This work takes place in the domain adaptation scenario: Given N^A samples $(x^A_i, y^A_i) \sim \bar{P}^A_{X,Y}$, N^B samples $x^B_i \sim \bar{P}^B_X$, with $x^A_i, x^B_i \in \mathbb{R}^M$ and $y^A_i \in \mathcal{Y}$, the label space, our goal is to jointly find a feature projection $\phi(\cdot)$ and predictive function f minimizing expected loss under loss function L and test distribution $P^B_{X,Y}$: $E_{X,Y\sim P^B_{X,Y}}[L(f(\phi(X)),Y]$. We will assume ϕ is a linear projection from $\mathbb{R}^M$ to $\mathbb{R}^K$ for given $K < M$, so that $\phi(x) = M^T x$ where $M \in S(N,K)$, the set of Stiefel manifolds, which consist of the $N \times K$ matrices with orthonormal columns. Thus, f has domain $\mathbb{R}^K$. We also assume f to be linear, parameterized by $\theta \in \mathbb{R}^K$ so that we may write $f(\cdot;\theta)$.

2.2 Optimization Problem

Given the above assumptions, the optimization problem we solve is as follows:

$$\min_{\substack{M\in S(N,K)\\ \theta\in\mathbb{R}^K\\ \hat{\beta}\in\mathbb{R}^K}} \quad \underbrace{\sum_i w(u^A_i)L(f(u^A_i;\theta),y^A_i)}_{\text{estimate of in-sample loss under}P^B_{\phi(X),Y}} \quad + \quad \underbrace{R(\theta)}_{\text{regularization}} \qquad \text{(empirical risk minimization)} \tag{4}$$

where

$$u_i^A = M^T x_i^A, \; u_i^B = M^T x_i^B \qquad \text{(defining projected features)} \quad (5)$$

$$f(u_i^A; \theta) = g(\theta^T u_i^A) \qquad \text{(defining predictive outputs)} \quad (6)$$

$$\hat{\beta} = \text{argmin}_\beta \sum_{u_i, z_i \in \{u_i^A, 1\} \cup \{u_i^B, 0\}} L^{\text{logistic}}(\text{logistic}(\beta^T u_i), z_i) \qquad \text{(running logistic regression)} \quad (7)$$

$$w(u_i^A) = \tfrac{N^A}{N^B} \, \text{logistic}(\hat{\beta}^T u_i^A) \qquad \text{(obtaining weights)} \quad (8)$$

Thus, given X, M defines projected random variable $U = M^T X$. $\mathcal{A}$ assumes $f(\cdot)$ to be a (generalized) linear function parameterized by θ (and fixed link function $g(\cdot)$). $\mathcal{A}$, given empirical *projected* distribution $\bar{P}_{U,Y}^A$, learns $f(\cdot)$ by attempting to minimize in-sample loss under $P_{U,Y}^B$ plus a regularization term $R(\theta)$. However, as we do not have $\bar{P}_{U,Y}^B$, we estimate that in-sample loss with $\sum_i w(u_i^A) L(f(u_i^A; \theta), y_i^A)$. This estimate is unbiased if $w(u) = \frac{P_U^B(u)}{P_U^A(u)}$[4]. We obtain these weights in a 2 step subproblem, following the approach of [2]: we first learn a logistic regression classifier (Equation 7) that differentiates between $u_i^B \sim \bar{P}_U^B$ (labelled 1) and $u_i^A \sim \bar{P}_U^A$ (labelled 0). Having learned the classifier, the weight estimates are given by Equation 8.

2.3 Solving the Optimization Problem

We solve the optimization problem via gradient descent, as there are non-linearities in both the constraints and objective function. Thus the optimization is straightforward, aside from two issues. The first issue is that as $M \in S(N, K)$, naive gradient steps for M will result in M no longer satisfying the Stiefel manifold constraints. Thus we will need to use manifold optimization methods [3]. The second issue is that $\hat{B}$ depends on $\vec{u} := \{u_i^A\} \cup \{u_i^B\}$ not through a specified functional relation, but as the solution to a (convex logistic regression) optimization problem parameterized by $\vec{u}$ (see Equation 7). Thus to calculate $\frac{d\hat{B}}{d\vec{u}}$ we will have to use implicit differentiation based on the first order optimality relation satisfied between $\hat{B}$ and $\vec{u}$, as described in [1].

References

[1] Yoshua Bengio. Gradient-based optimization of hyperparameters. *Neural computation*, 12(8):1889–1900, 2000.

[2] Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on Machine learning*, pages 81–88. ACM, 2007.

[3] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[4] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.