Learning Unbiased Transferability for Domain Adaptation by Uncertainty Modeling

Anonymous ECCV submission

Paper ID 5157

1 Theoretical Analysis

The challenge of DA is to use the source classifier to minimize the target classification error $\mathbb{E}_{(x,y)\sim p_t}[\mathcal{L}_y(G_y(G_f(x)),y)]$ when target labels are unavailable. Since the existence of domain shift, the source classifier cannot work properly on target samples. To better account for the shift between source and target distributions, density ratio $w(x)=\frac{p_t(x)}{p_s(x)}$ [5,7] can be used as a metric of transferability to measure the discrepancy between domains. Specifically, the shift will be eliminate if and *only* if $w(x)=\frac{p_t(x)}{p_s(x)}=1$. Then, the target classification error can be estimated by the source distribution p_s as:

$$\begin{aligned}&\mathbb{E}_{(x,y)\sim p_t}[\mathcal{L}_y(G_y(G_f(x)),y))]=\int_{D_t}\mathcal{L}_y\left(G_y(G_f(x)),y\right)p_t(x)dx\\&=\int_{D_s}\frac{p_t(x)}{p_s(x)}\mathcal{L}_y(G_y(G_f(x)),y)p_s(x)dx=\mathbb{E}_{(x,y)\sim p_s}\left[w(x)\mathcal{L}_y(G_y(G_f(x)),y))\right].\end{aligned}\tag{1}$$

However, the density ratio $w(x)$ is often not accessible in DA, we follow [5] and use estimated $\hat{w}(x)$ to approximate the real $w(x)$. Specifically, LogReg [1,4] is used to estimate the density ratio by Bayesian formula:

$$\begin{aligned}\hat{w}(x)=\frac{p_t(x)}{p_s(x)}&=\frac{m(x|d=0)}{m(x|d=1)}=\frac{P(d=1)P(d=0|x)}{P(d=0)P(d=1|x)}\\&=\frac{n_s}{n_t}\cdot\frac{P(d=0|x)}{P(d=1|x)}=\frac{P(d=0|x)}{P(d=1|x)},\end{aligned}\tag{2}$$

where m is a distribution over $(x,d)\sim X\times(0,1)$, denoting the Cartesian product between sample sets and domain label sets and $d\sim Bernoulli(0.5)$ is a Bernoulli variable representing which domain x belongs to. Here, $\frac{n_s}{n_t}$ is a constant regarding to sample sizes and the source or the target dataset is randomly up-sampled to ensure $n_s=n_t$. In this way, $\hat{w}(x)$ only depends on $\frac{P(d=0|x)}{P(d=1|x)}$.

In DA, $w(x)$ can be treated as the real transferability, while $\hat{w}(x)$ can be considered as the estimated transferability. Ideally, when the source and target domain is matched perfectly, $P(d=0|x)=P(d=1|x)=0.5$, $w(x)=\frac{P(d=0|x)}{P(d=1|x)}=1$, target classification error can be expressed by source one as follows:

$$\begin{aligned}&\mathbb{E}_{(x,y)\sim p_t}[\mathcal{L}_y(G_y(G_f(x)),y))]=\mathbb{E}_{(x,y)\sim p_s}[w(x)\mathcal{L}_y(G_y(G_f(x)),y))]\\&=\mathbb{E}_{(x,y)\sim p_s}[\hat{w}(x)\mathcal{L}_y(G_y(G_f(x)),y))]=\mathbb{E}_{(x,y)\sim p_s}[\mathcal{L}_y(G_y(G_f(x)),y))]\end{aligned}\tag{3}$$

Therefore, when the gap between domains are eliminated, target classification error can be remoted, source classifier works well on trget samples. However, there are two

thing for us to notice. First, how to make estimated transferability $\hat{w}(x) = 1$ to remote domain discrepancy. Second, how to eliminate the estimated bias between estimated $\hat{w}(x)$ and true $w(x)$ to obtain unbiased transferability.

According to the above analysis, if the estimated transferability $\hat{w}(x)$ is equal to the real transferability $w(x)$, the transferability estimation can be unbiased. We formalize the target classification deviation as:

$$\begin{aligned}&\mathbb{E}_{(x,y)\sim p_t}[\mathcal{L}_y(G_y(G_f(x)),y))] = \int_{D_t} \mathcal{L}_y\left(G_y(G_f(x)),y\right) p_t(x)dx \\ &= \int_{D_s} \frac{p_t(x)}{p_s(x)} \mathcal{L}_y(G_y(G_f(x)),y) p_s(x)dx = \mathbb{E}_{(x,y)\sim p_s}\left[w(x)\mathcal{L}_y(G_y(G_f(x)),y))\right].\end{aligned} \tag{4}$$

Based on Cachy-Schwarz Ineqaulity and the inequality of arithmetic and geometric means, we have

$$\begin{aligned}&\left|\mathbb{E}_{(x,y)\sim p_s}(\hat{w}(x) - w(x))\mathcal{L}_y(G_y(G_f(x)),y))\right| \\ &\leq \sqrt{\mathbb{E}_{(x,y)\sim p_s}[(\hat{w}(x) - w(x))^2]\mathbb{E}_{(x,y)\sim p_s}[(\mathcal{L}_y(G_y(G_f(x)),y))^2]} \\ &\leq \frac{1}{2}\left(\mathbb{E}_{(x,y)\sim p_s}[(\hat{w}(x) - w(x))^2] + \mathbb{E}_{(x,y)\sim p_s}[(\mathcal{L}_y(G_y(G_f(x)),y))^2]\right),\end{aligned} \tag{5}$$

In the above inequality, since the second term is bounded by supervised learning in the labeled source domain, we only need to focus on the first term. We use a discriminator to alleviate the deviation between the estimated $\hat{w}(x)$ and the real $w(x)$. From our unbiased transferability perspective, we further formalize the transferability based on discriminator as

$$w(x) = \frac{B_t(x|d=0)}{B_s(x|d=1)}, \tag{6}$$

Here, distribution B is obtained from (x, d). In this case, $d \sim Bernoulli(0.5)$, if $d = 1$, $x \sim p_s$ or $x \sim p_t$. Furthermore, as the unbiased transferability is derived in the discriminator label space, $W(x)$ and $\hat{W}(x)$ are assumed to be the real and estimated transferability in this space. Assume we have upper bound $N \geq 0$ for $W(x)$ subject to $N \geq W(x) \geq 0$ according to the bounded importance weight assumption [2]. Since

$$W(x) = \frac{P_d(d=0|x)}{P_d(d=1|x)} = \frac{1 - P_d(d=1|x)}{P_d(d=1|x)} = \frac{1}{P_d(d=1|x)} - 1, \tag{7}$$

we have

$$\frac{1}{N+1} \leq P_d(d=1|x) \leq 1, \tag{8}$$

Furthermore, since the optimal discriminator is:

$$P_d(x) = B(d=1|x), \tag{9}$$

then we have

$$P_d(x) = B(d=1|x) = \frac{1}{1+W(x)}, \tag{10}$$

In this way, the first term of Eq. (5) is bounded:

$$\begin{aligned}
&\mathbb{E}_{x\sim p_s}\left[(\hat{w}(x_i)-w(x))^2\right]=\mathbb{E}_{x\sim p_{ds}}[(\hat{W}(x)-W(x))^2*\frac{p_s(x)}{p_{ds}(x)}]\\
\leq&2\mathbb{E}_{x\sim p_{ds}}[(\hat{W}(x_i)-W(x))^2]=2\mathbb{E}_{x\sim p_{ds}}\left[\left(\frac{P_d(d=1|x)-\hat{P}_d(d=1|x)}{P_d(d=1|x)\hat{P}_d(d=1|x)}\right)^2\right]\\
\leq&2(N+1)^4\mathbb{E}_{x\sim p_{ds}}\left[\Big(P_d(d=1|x)-\hat{P}_d(d=1|x)\Big)^2\right].
\end{aligned}\tag{11}$$

Here, p_{ds} and p_{dt} are the source and target discriminator distributions, a instance is sampled from p_d which equals to be sampled from p_{ds} or p_{dt}. The first two rows change the probability from source label space to source discriminator domain label space. Then, the deviation between the real and the estimated transferability is calculated in the discriminator label space. The second inequality in Eq. (11) can be further rewritten as:

$$\begin{aligned}
&2(N+1)^4\mathbb{E}_{x\sim p_{ds}}\left[\Big(P_d(d=1|x)-\hat{P}_d(d=1|x)\Big)^2\right]\\
=&2(N+1)^4\Big(\mathbb{E}_{x\sim p_{ds}}[(P_d(d=1|x))^2]+\mathbb{E}_{x\sim p_{ds}}[(\hat{P}_d(d=1|x))^2]\\
&-2\mathbb{E}_{x\sim p_{ds}}[\hat{P}_d(d=1|x)P_d(d=1|x)]\Big)\\
=&2(N+1)^4\Big(\mathbb{E}_{x\sim p_{ds}}[(P_d(d=1|x))^2]-(\mathbb{E}_{x\sim p_{ds}}[P_d(d=1|x)])^2\\
&+(\mathbb{E}_{x\sim p_{ds}}[P_d(d=1|x)])^2+\mathbb{E}_{x\sim p_{ds}}[(\hat{P}_d(d=1|x))^2]\\
&-(\mathbb{E}_{x\sim p_{ds}}[\hat{P}_d(d=1|x)])^2+(\mathbb{E}_{x\sim p_{ds}}[\hat{P}_d(d=1|x)])^2\\
&-2\mathbb{E}_{x\sim p_{ds}}[\hat{P}_d(d=1|x)P_d(d=1|x)]\Big)\\
=&2(N+1)^4\Big(\mathbb{V}ar_{x\sim p_{ds}}(P_d(d=1|x))+\mathbb{V}ar_{x\sim p_{ds}}(\hat{P}_d(d=1|x))\\
&+\Big(\mathbb{E}_{x\sim p_{ds}}[P_d(d=1|x)]-\mathbb{E}_{x\sim p_{ds}}[\hat{P}_d(d=1|x)]\Big)^2\Big),
\end{aligned}\tag{12}$$

where the variances of outputs under real and estimated probability distributions are $\mathbb{V}ar_{x\sim p_{ds}}(P_d(d=1|x))$ and $\mathbb{V}ar_{x\sim p_{ds}}(\hat{P}_d(d=1|x))$ respectively. Ideally, when the source and target domains are aligned, the output of discriminator should be 0.5, then, $\mathbb{V}ar_{x\sim p_{ds}}(P_d(d=1|x))=0$. Hence, the bias of transferability can be formulated as:

$$\begin{aligned}
\mathbb{E}_{x\sim p_{ds}}\Big[(\hat{w}(x)-w(x))^2\Big]\leq 2(N+1)^4\Big[&\mathbb{V}ar_{x\sim p_{ds}}\Big(\hat{P}_d(d=1|x)\Big)\\
&+\Big(\mathbb{E}_{x\sim p_{ds}}[P_d(d=1|x)]-\mathbb{E}_{x\sim p_{ds}}[\hat{P}_d(d=1|x)]\Big)^2
\end{aligned}\tag{13}$$

The second term of Eq. (13) is constrained since the domain adaptation process encourages $\mathbb{E}_{x\sim p_{ds}}(P_d(d=1|x))$ to approximate to $\mathbb{E}_{x\sim p_{ds}}(\hat{P}_d(d=1|x))$. Therefore, to learn unbiased transferability, we can minimize $\mathbb{V}ar_{x\sim p_{ds}}(\hat{P}_d(d=1|x))$.

We use MCDropout [3,6] to compute $\mathbb{V}ar_{x\sim p_d}(\hat{P}_d(d|x))$ as:

$$\mathbb{V}ar_{x\sim p_d}(\hat{P}_d(d|x))\approx\frac{1}{K}\sum_{k=1}^{K}\left((G_d(G_g(x)))_k-\left(\frac{1}{K}\sum_{k=1}^{K}(G_d(G_g(x)))_k\right)\right)^2,\tag{14}$$

where K is the number of times performing stochastic forward passes through the discriminator network. We set $u(x) = \mathbb{V}ar_{x\sim p_{ds}}(\hat{P}_d(d=1|x)) + \mathbb{V}ar_{x\sim p_{dt}}(\hat{P}_d(d=0|x))$, i.e., the modeled uncertainty. Here, We set $\mathbf{U} = [u(x_1), ...u(x_i), ...]$. $u(x_i)$ represents the modeled uncertainty for the ith sample.

Based on Eq. (14), we define the L2 regularized $\mathbf{U}$ as the bias loss and minimize the bias loss as:

$$\min \mathcal{L}_{bias} = \min ||\mathbf{U}||_2^2 = \min_{G_d, G_f} \sum_{i=1}^{n_t+n_s} \left(\mathbb{V}ar_{x_i\sim p_d}(\hat{P}_d(d|x_i))\right)^2, \tag{15}$$

When bias loss becomes small, the upper bound of $\mathbb{E}_{x\sim\hat{p}_s}[(\hat{w}(x) - w(x))^2]$ can be lower,leading to a smaller bias of the target classification transfer error. an unbiased estimation of the target classification error is achieved if $\mathcal{L}_{bias} = 0$, where the estimated $\hat{w}(x)$ are equal to the true $w(x)$.

2 Implement Details

Our results are all obtained without heavy engineering tricks. We randomly run our methods with batch size 64 for three times with different random seeds $\{0, 1, 2\}$ via **PyTorch**, and report the mean accuracy. We run all the experiments on a single Titan V100 (32G) GPU with 20 epochs for **Office-31** and 30 epochs for **Office-Home** and **VisDA**.

2.1 Implement Details for UDA scenarios

For the **Office-31** and **Office-Home** datasets, we use the pretrained ResNet-50 model as the backbone, while selecting ResNet-101 for larger and more challenging **VisDA** dataset. Following DANN, CADA, MDD and TransPar, we replace the original FC layer with a bottleneck layer (256 units for DANN-based and CADA-based, 1024 units for MDD-based and TransPar-based). Specifically, we reimplement CADA on our own, leading to more similar implemention with DANN.

We update the unbiased transferbility each epoch by using the normalized variance of 10 times MCDropout forward propagation. Specifically, the normalization is conducted following the formulation $x_i = \frac{x_i - min(x_i)}{max(x_i)}$ and the normalized variance is seen as unbiased transferbility for the next adversarial adaptation and pseudo label learning. Meanwhile, the $\mathcal{L}_{bias}$ loss is obtained by 3 times MCDropout forward propagation each batch during training.

As a plug-in, we adopt the same learning rate scheduler, base learning rate, weight decay factor and use the same optimizer as DANN, CADA, MDD and TransPar. For all experiments, the training augmentation contains RandomResizedCrop and RandomHorizontalFlip except CenterCrop and RandomHorizontalFlip for **VisDA**. The default hyper-parameter α_{tce} is 1, α_{bias} is 0.01.

2.2 Implement Details for SSDA scenarios

Under SSDA setting, we follow the basic setting of UDA, but in this case, not only source domain, but also a part of target domain have labels. We follow the normal setting in SSDA, only 1% of target samples are randomly selected out as target labeled domain. During training, each batch has 64 samples, 16 samples from source domain, 16 samples are from target labeled domain and 16 samples are from target unlabeled domain. The parameter setting is the same as UDA.

2.3 Implement Details for SSL scenarios

Under SSL setting, we make a little change to the DA setting, we treat target domain under DA setting in two parts: labeled target domain and unlabeled target domain. The labeled target domain contains 3 labeled samples for each class setting as source domain, the remained unlabeled target domain is set as unlabeled domain. Here, there is no domain gap between labeled and unlabeled domain while there is still some discrepancy between labeled and unlabeled domain due to label lacking.

We use ResNet-34 as backbone networks and adopt SGD with learning rate of 1e-3, momentum of 0.9 and weight decay factor of 5e-4. We decay the learning rate with a multiplier 0.1 when training process reach three quarters of the total iterations. The batch size is set as 64 for **Office-Home**. There are 32 labeled samples and 32 unlabeled samples in each batch for training. For adversarial training, we use gradient reversal layer (GRL) to flip gradient in the backpropagation between feature encoder and domain discriminator to obtain domain invariant.

References

1. Bickel, S., Scheffer, T.: Dirichlet-enhanced spam filtering based on biased samples. In: NIPS (2007)
2. Cortes, C., Mansour, Y., Mohri, M.: Learning bounds for importance weighting. In: NIPS (2010)
3. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: ICML (2016)
4. Qin, J.: Inferences for case-control and semiparametric two-sample density ratio models. Biometrika **85**(3), 619–630 (1998)
5. Wang, X., Long, M., Wang, J., Jordan, M.: Transferable calibration with lower bias and variance in domain adaptation. Advances in Neural Information Processing Systems pp. 19212–19223 (2020)
6. Wen, J., Zheng, N., Yuan, J., Gong, Z., Chen, C.: Bayesian uncertainty matching for unsupervised domain adaptation. arXiv preprint arXiv:1906.09693 (2019)
7. You, K., Wang, X., Long, M., Jordan, M.: Towards accurate model selection in deep unsupervised domain adaptation. In: International Conference on Machine Learning. pp. 7124–7133. PMLR (2019)