The Role of Embedding Complexity in Domain-invariant Representations

Ching-Yao Chuang [1] **Antonio Torralba** [1] **Stefanie Jegelka** [1]

Abstract

Unsupervised domain adaptation aims to generalize the hypothesis trained in a source domain to an unlabeled target domain. One popular approach to this problem is to learn a domain-invariant representation for both domains. In this work, we study, theoretically and empirically, the explicit effect of the embedding on generalization to the target domain. In particular, the complexity of the class of embeddings affects an upper bound on the target domain's risk. This is reflected in our experiments, too.

1. Introduction

Domain adaptation is critical in many applications where collecting large-scale supervised data is prohibitively expensive or intractable, or conditions at prediction time can change. For instance, self-driving cars must be robust to various conditions such as different weather, change of landscape and traffic. In such cases, the model learned from limited source data should ideally generalize to different target domains. Specifically, unsupervised domain adaptation aims to transfer knowledge learned from a labeled source domain to similar but completely unlabeled target domains.

One popular approach to unsupervised domain adaptation is to learn domain-invariant representations (Ben-David et al., 2007; Long et al., 2015; Ganin et al., 2016), by minimizing a divergence between the representations of source and target domains. The prediction function is learned on the latent space, with the aim of making it domain-independent. A series of theoretical works justifies this idea (Ben-David et al., 2007; Mansour et al., 2009; Ben-David et al., 2010; Cortes & Mohri, 2011).

Despite the empirical success of domain-invariant representations, exactly matching the representations of source and target distribution can sometimes fail to achieve domain adaptation. For example, Wu et al. (2019) show that exact matching may increase target error if label distributions are different between source and target domain, and propose a new divergence metric to overcome this limitation. Zhao et al. (2019) establish lower and upper bounds on the risk when label distributions between source and target domains differ. Johansson et al. (2019) point out the information lost in non-invertible embeddings, and propose different generalization bounds based on the overlap of the supports of source and target distribution.

In contrast to previous analyses that focus on changes in the label distributions or on joint support, we here study the effect of the complexity of the joint representation. In particular, we show a general bound on the target risk that reflects a tradeoff between the embedding complexity and the divergence of source and target in the latent representation space. In particular, a too powerful class of embedding functions can result in overfitting the source data and the distribution matching, leading to arbitrarily high target risk. Hence, a restriction (taking into account assumptions about correspondences and invariances) is needed. Our experiments reflect these trends empirically, too.

2. Unsupervised Domain Adaptation

For simplicity, we consider binary classification with input space $\mathcal{X} \subseteq \mathbb{R}^n$ and output space $\mathcal{Y} = \{0, 1\}$. Define $\mathcal{H}$ to be the hypothesis class from $\mathcal{X}$ to $\mathcal{Y}$. The learning algorithm obtains two datasets: labeled source data $\mathcal{X}_S$ with distribution p_S, and unlabeled target data $\mathcal{X}_T$ with distribution p_T. We will use p_S and p_T to denote the joint distribution on data and labels X, Y and the marginals, i.e., $p_S(X)$ and $p_S(Y)$. Unsupervised domain adaptation seeks a hypothesis $h \in \mathcal{H}$ that minimizes the risk in the target domain measured by a loss function ℓ (here, zero-one loss):

$$R_T(h) = \mathbb{E}_{x,y\sim p_T}[\ell(h(x), y)]. \tag{1}$$

We will not assume common support in source and target domain, in line with standard benchmarks for domain adaptation such as adapting from MNIST to M-MNIST.

2.1. Domain-invariant Representations

A common approach to domain adaptation is to learn a joint embedding of source and target data (Ganin et al., 2016;

[1]Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), Cambridge, USA.. Correspondence to: Ching-Yao Chuang <cychuang@mit.edu>.

Proceedings of the 1st Adaptive & Multitask Learning Workshop, Long Beach, California, 2019.

Tzeng et al., 2017). The idea is that aligning source and target distributions in this latent space $\mathcal{Z}$ results in a domain-invariant representation, and hence a subsequent classifier f from the embedding to $\mathcal{Y}$ will generalize from source to target. Formally, this results in the following objective function on the hypothesis $h = fg := f \circ g$, where $\mathcal{G}$ is the class of embedding functions to $\mathcal{Z}$, and we minimize a divergence d between the distributions $p_S(Z_g) = p_S(g(X)), p_T(Z_g)$ of source and target after mapping to $\mathcal{Z}$:

$$\min_{f \in \mathcal{F}, g \in \mathcal{G}} R_S(fg) + \alpha d(p_S(Z_g), p_T(Z_g)). \tag{2}$$

The divergence d could be, e.g., the Jensen-Shannon (Ganin et al., 2016) or Wasserstein distance (Shen et al., 2017).

2.2. Upper bounds on the target risk

Ben-David et al. (2007) introduced the $\mathcal{H}\Delta\mathcal{H}$-divergence to bound the worst-case loss from extrapolating between domains. Let $R_D(h, h') = \mathbb{E}_{x \sim D}[\ell(h(x), h'(x))]$ be the expected disagreement between two hypotheses, then the $\mathcal{H}\Delta\mathcal{H}$-divergence is defined as follows.

Definition 1. ($\mathcal{H}\Delta\mathcal{H}$-divergence) *Given two domain distributions p_S and p_T over $\mathcal{X}$, and a hypothesis class $\mathcal{H}$, the $\mathcal{H}\Delta\mathcal{H}$-divergence between p_S and p_T is*

$$d_{\mathcal{H}\Delta\mathcal{H}}(p_S, p_T) = \sup_{h, h' \in \mathcal{H}} |R_S(h, h') - R_T(h, h')|.$$

This divergence allows to bound the risk on the target domain:

Theorem 1. (Ben-David et al., 2010) *For all hypotheses $h \in \mathcal{H}$, the target risk is bounded as*

$$R_T(h) \leq R_S(h) + d_{\mathcal{H}\Delta\mathcal{H}}(p_S, p_T) + \lambda_{\mathcal{H}},$$

where $\lambda_{\mathcal{H}}$ is the best joint risk

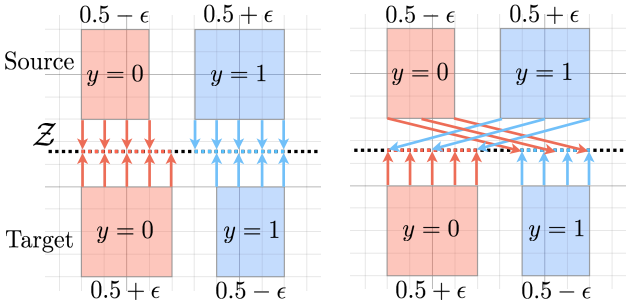
$$\lambda_{\mathcal{H}} := \inf_{h \in \mathcal{H}} [R_S(h) + R_T(h)]$$

Similar results have been obtained for continuous labels (Cortes & Mohri, 2011; Mansour et al., 2009).

Theorem 1 is an influential theoretical result in unsupervised domain adaptation, and motivated work on domain invariant representations. For example, recent work (Ganin et al. (2016); Johansson et al. (2019)) applied Theorem 1 to the hypothesis space $\mathcal{F}$ that maps the representation space $\mathcal{Z}$ induced by an encoder g to the output space:

$$R_T(fg) \leq R_S(fg) + d_{\mathcal{F}\Delta\mathcal{F}}(p_S(Z_g), p_T(Z_g)) + \lambda_{\mathcal{F}}(g) \tag{3}$$

where $\lambda_{\mathcal{F}}(g)$ is the best hypothesis risk with fixed g, i.e., $\lambda_{\mathcal{F}}(g) := \inf_{f \in \mathcal{F}}[R_S(fg) + R_T(fg)]$. The $\mathcal{F}\Delta\mathcal{F}$ divergence implicitly depends on the fixed g and can be small if g provides a suitable representation. However, if g induces a wrong alignment, then the best hypothesis risk $\lambda_{\mathcal{F}}(g)$ is large with any function class $\mathcal{F}$. The following example will illustrate such a situation, motivating to explicitly take the class of embeddings into account when bounding the target risk.



(a) Linear Embedding (b) Nonlinear Embedding

Figure 1. Illustrative example in 2D. The 1D representation space is illustrated as a dotted line and the arrows represent the embedding from 2D to 1D. (a) Optimal representations when $\mathcal{G}$ is the class of linear functions from 2D to 1D. (b) Optimal representation with a complex nonlinear function class with zero source and divergence loss; this representation destroys label consistency and leads to maximal target risk.

3. Influence of the representation

We begin with an illustrative toy example. Figure 1 shows a binary classification problem in 2D with disjoint support and a slight shift in the label distributions from source to target: $p_S(y = 1) = p_T(y = 1) + 2\epsilon$. Assume the representation space is one dimensional, so the embedding g is a function from 2D to 1D. If we allow arbitrary, nonlinear embeddings, then, for instance, the embedding shown in Figure 1(b), together with an optimal predictor, achieves zero source loss and a zero divergence, and is hence optimal according to the objective (2). However, the target risk of this combination of embedding and predictor is maximal: $R_T(fg) = 1$.

If we restrict the class $\mathcal{G}$ of embeddings to linear maps $g(x) = \mathbf{W}x$ where $\mathbf{W} \in \mathbb{R}^{1 \times 2}$, then the embeddings that are optimal with respect to the objective (2) are of the form $\mathbf{W} = \begin{bmatrix} a, 0 \end{bmatrix}$. Together with an optimal source classifier f, they achieve a non-zero value of 2ϵ for objective (2) due to the shift in class distributions. However, these embeddings retain label correspondences, and can lead to a zero target risk.

This example illustrates that a too rich class of embeddings can "overfit" the alignment, and hence lead to arbitrarily bad solutions. Hence, the complexity of the encoder class plays an important role in learning domain invariant representation too.

3.1. Bounds for Domain-invariant Representations

Motivated by the above example, we next expose how the bound on the target risk depends on the complexity of the embedding class. To do so, we apply Theorem 1 to the hypothesis $h = fg$:

$$R_T(fg) \leq R_S(fg) + d_{\mathcal{FG}\Delta\mathcal{FG}}(p_S, p_T) + \lambda_{\mathcal{FG}}. \quad (4)$$

Comparing the bound (4) to the previous bound (3), we notice two differences: the best in-class joint risk now minimizes over both $\mathcal{F}$ and $\mathcal{G}$, i.e.,

$$\lambda_{\mathcal{FG}} \coloneqq \inf_{f\in\mathcal{F}, g\in\mathcal{G}} [R_S(fg) + R_T(fg)], \quad (5)$$

which is smaller than $\lambda_{\mathcal{F}_g}$ and reflects the fact that we are learning both f and g. In return, the divergence term $d_{\mathcal{FG}\Delta\mathcal{FG}}(p_S, p_T)$ becomes larger than the one in bound (3). To better understand these tradeoffs, we derive a more interpetable form of the bound on the target risk. Before presenting the bound, we define an extended version of $\mathcal{H}\Delta\mathcal{H}$:

Definition 2. ($\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}$-divergence) *For two domain distributions p_S and p_T over $\mathcal{X}$, an encoder class $\mathcal{G}$, and predictor class $\mathcal{F}$, the $\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}$-divergence between p_S and p_T is*

$$d_{\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}}(p_S, p_T) = \sup_{\substack{f\in\mathcal{F} \\ g,g'\in\mathcal{G}}} |R_S(fg, fg') - R_T(fg, fg')|.$$

Note that the $\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}$-divergence is strictly smaller than the $\mathcal{FG}\Delta\mathcal{FG}$-divergence, since the two hypotheses in the supremum, fg and fg', share the same predictor f. We are ready to state the following result.

Theorem 2. *For all $f \in \mathcal{F}$ and $g \in \mathcal{G}$,*

$$R_T(fg) \leq R_S(fg) + \underbrace{d_{\mathcal{F}\Delta\mathcal{F}}(p_S(Z_g), p_T(Z_g))}_{\text{(i) Latent Divergence}} + \underbrace{d_{\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}}(p_S, p_T) + \lambda_{\mathcal{FG}}(g)}_{\text{(ii) Complexity Trade-off}}. \quad (6)$$

where $\lambda_{\mathcal{FG}}(g)$ is the best in-class joint risk defined as

$$\lambda_{\mathcal{FG}}(g) = \inf_{f'\in\mathcal{F}, g'\in\mathcal{G}} 2R_S(f'g) + R_S(f'g') + R_T(f'g').$$

A detailed proof of the theorem may be found in the Appendix. The first term of the bound is the source risk. The second term (i) is the $\mathcal{F}\Delta\mathcal{F}$-divergence between the distributions $p_S(Z_g)$ and $p_T(Z_g)$ in the representation space; this also appears in the previous bound (3). The first term in (ii) measure the $\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}$-divergence between source and target distribution, which may decrease as the complexity of the encoder decreases. However, a less complex encoder class $\mathcal{G}$ can lead to increasing the best hypothesis risk $\lambda_{\mathcal{FG}}(g)$. Therefore, (ii) makes a trade-off explicit between the divergence and the model complexity. Note that, as opposed to different $\lambda_{\mathcal{FG}}$, $\lambda_{\mathcal{FG}}(g)$ also measures the correctness of the encoder in the source domain. If the encoder fails to provide informative representations in the source domain, then first term in $\lambda_{\mathcal{FG}}(g)$ can be large.

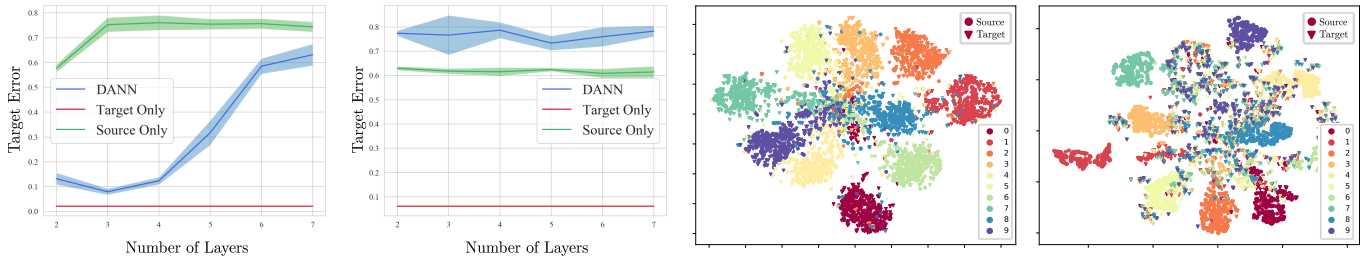
The last two terms in Theorem 1 express a similar complexity trade-off as (ii), but this time with repect to the hypothesis class $\mathcal{H}$, which here combines the encoder and predictor. Hence, directly applying Theorem 1 to the composition (Equation (4)) treats both jointly and does not make the role of the embedding as explicit as Theorem 2. For example, Theorem 2 shows that we can also make the bound tighter by minimizing the divergence between the corresponding distributions in the embedding space, as long as the encoder provides useful representations in the source domain. If (i) is sufficiently small, the $\mathcal{FG}\Delta\mathcal{FG}$-divergence reduces to the $\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}$-divergence, which is strictly smaller than the $\mathcal{FG}\Delta\mathcal{FG}$-divergence.

Comparing to the previous bound in Equation (3), which assumes a fixed g, we do not assume a known encoder and instead quantify the effect of the encoder family. Moreover, the term $\lambda_{\mathcal{F}}(g)$ in bound (3) involves the source and target risk, whereas in $\lambda_{\mathcal{FG}}(g)$ the encoder g only affects the source risk, which can be estimated empirically.

Importantly, without restricting the complexity of the encoder or embedding, the $\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}$-divergence can be large, indicating that the target risk may be large too. This suggests that restricting the model complexity of the embedding is crucial for domain invariant representation learning.

3.2. Practical Implications

To reduce the worst case divergence (i), we need to restrict the encoder family to those that can approximately minimize (i), in coordination with the predictor class $\mathcal{F}$. Practically, we can optimize the original objective of domain invariant representations in Equation 2 to align the latent distributions. Term (ii) implies that we should choose the minimal complexity encoder class $\mathcal{G}$ that is is still expressive enough to encode the data from both domains. Practically, this can be done by regularizing the encoder, e.g., restricting Lipschitz constants or norms of weight matrices. More explicitly, one may limit the number of layers of a neural network, or apply inductive biases via selecting network architectures. For instance, comparing to fully connected networks (FCs), convolutional neural networks (CNNs) restrict the output to be spatially consistent with respect to the input.



(a) Target Error (CNN) (b) Target Error (MLP) (c) t-SNE (CNN) (d) t-SNE (MLP)

Figure 2. Experiment Results on MNIST → M-MNIST. (a) (b): Target error with respect to number of encoder's layers. Complexities of encoders have a direct impact on the target errors in CNN case. However, without inductive biases, DANNs with MLP encoder consistently perform worse than hypotheses only trained in source domain. (c) (d): t-SNE projections of representations with different inductive biases. CNN encoders result in target representations that are well align with those in source domain. However, MLP encoders lose label-consistency while minimizing the latent divergence between domains.

4. Experiments

Next, we empirically test Theorem 2 via one example of domain-invariant representations: Domain-Adversarial Neural Networks (DANN) (Ganin et al., 2016), which measure the latent divergence via a domain discriminator (Jensen-Shannon divergence). We use the standard benchmark MNIST → MNIST-M (Ganin & Lempitsky (2014)), where the task is to classify unlabeled handwritten digits overlayed with random photographs (MNIST-M) based on labeled images of digits alone (MNIST). We consider two categories of complexity: number of layers and inductive bias (CNN).

4.1. Number of Layers of Encoder

To analyze the effect of the encoder's complexity, we augment the original two-layer CNN encoders with 1 to 5 additional CNN layers, leaving other settings unchanged. We retrain each model for 5 times and plot the mean and standard deviation of target error with respect to the number of layers in Figure 2(a): Initially, the target error decreases, and then increases when more layers are added. This corroborates our theory: the CNN encoder without additional layers does not have enough expressive power. As a consequence, the best hypothesis risk term $\lambda_{\mathcal{FG}}$ is larger. However, when more layers are added, the complexity increases and subsequently makes the disagreements larger.

4.2. Inductive Bias of Encoder

To investigate the importance of inductive bias in domain invariant representations, we replace the CNN encoder with an MLP encoder. The experimental results are shown in 2(b). Comparing the target error between (a) and (b) in Figure 2, we can see that the target error with an MLP encoder is significantly higher than with a CNN encoder. Comparing to CNNs, which encode invariance via pooling and learned filters, MLPs do not have any inductive bias and lead to worse performance. In fact, the target error with MLP-based domain adaptation is higher than just training on the source, suggesting that, without an appropriate inductive bias, learning domain invariant representations can even worsen the performance. To gain deeper insight, we use t-SNE (Maaten & Hinton, 2008) to visualize source and target embedding distributions in Figure 2(c),(d). With the inductive bias of CNNs, the representations of the target domain aligns well with those of source domain. In contrast, the MLP encoder results in a strong label mismatch.

4.3. Discussion

The experiments show that the complexity of the encoder can have a direct effect on the target error. A more complex encoder class leads to larger theoretical bound on the target error, and, indeed, aligned with the theory, we see a significant performance drop in target domain. Moreover, the experiments suggest that inductive bias is important too. With a suitable inductive bias such as CNNs, DANN achieves higher performance than the with the MLP encoder, even if the CNN encoder has twice the number of layers. CNNs are standard for many vision tasks, such as digit recognition. However, explicit supervision may be required to identify the encoder class when we have less prior knowledge about the task (Motiian et al., 2017; Chen & Chien, 2015).

5. Conclusion

In this work, we study the role of embedding complexity for domain-invariant representations. We theoretically and empirically show that restricting the encoder is necessary for successful adaptation, a fact that has mostly been overlooked by previous work. In fact, without carefully selecting the encoder class, learning domain invariant representations might even harm the performance. Our observations motivate future research on identifying eappropriate encoder classes for various tasks.

References

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pp. 137–144, 2007.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

Chen, H.-Y. and Chien, J.-T. Deep semi-supervised learning for domain adaptation. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2015.

Cortes, C. and Mohri, M. Domain adaptation in regression. In *International Conference on Algorithmic Learning Theory*, pp. 308–323. Springer, 2011.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Johansson, F. D., Ranganath, R., and Sontag, D. Support and invertibility in domain-invariant representations. *arXiv preprint arXiv:1903.03448*, 2019.

Long, M., Cao, Y., Wang, J., and Jordan, M. I. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579–2605, 2008.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Motiian, S., Jones, Q., Iranmanesh, S., and Doretto, G. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 6670–6680, 2017.

Shen, J., Qu, Y., Zhang, W., and Yu, Y. Wasserstein distance guided representation learning for domain adaptation. *arXiv preprint arXiv:1707.01217*, 2017.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7167–7176, 2017.

Wu, Y., Winston, E., Kaushik, D., and Lipton, Z. Domain adaptation with asymmetrically-relaxed distribution alignment. *arXiv preprint arXiv:1903.01689*, 2019.

Zhao, H., Combes, R. T. d., Zhang, K., and Gordon, G. J. On learning invariant representation for domain adaptation. *arXiv preprint arXiv:1901.09453*, 2019.

Proofs

Theorem 2. *For all $f \in \mathcal{F}$ and $g \in \mathcal{G}$,*

$$R_T(fg) \leq R_S(fg) + d_{\mathcal{F}\Delta\mathcal{F}}(p_S(Z_g), p_T(Z_g)) + d_{\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}}(p_S, p_T) + \lambda_{\mathcal{F}\mathcal{G}}(g).$$

where $\lambda_{\mathcal{F}\mathcal{G}}(g)$ is the best in-class joint risk defined as

$$\lambda_{\mathcal{F}\mathcal{G}}(g) = \inf_{f' \in \mathcal{F}, g' \in \mathcal{G}} 2R_S(f'g) + R_S(f'g') + R_T(f'g') \tag{1}$$

Proof. We first define the optimal composition hypothesis $f^\star g^\star$ with respect to an encoder g to be the hypothesis which minimizes the following error

$$f^\star g^\star = \operatorname*{arg\,min}_{f' \in \mathcal{F}, g' \in \mathcal{G}} 2R_S(f'g) + R_S(f'g') + R_T(f'g') \tag{2}$$

By the triangle inequality for classification error (Ben-David et al. (2007)),

$$\begin{aligned} R_T(fg) &\leq R_T(f^\star g^\star) + R_T(fg, f^\star g^\star) && (3)\\ &\leq R_T(f^\star g^\star) + R_T(fg, f^\star g) + R_T(f^\star g, f^\star g^\star) && (4)\end{aligned}$$

The second term in the R.H.S of Eq. 4 can be bounded as

$$\begin{aligned} R_T(fg, f^\star g) &\leq R_S(fg, f^\star g) + |R_S(fg, f^\star g) - R_T(fg, f^\star g)| && (5)\\ &\leq R_S(fg, f^\star g) + \sup_{f, f' \in \mathcal{F}} |R_S(fg, f'g) - R_T(fg, f'g)| && (6)\\ &= R_S(fg, f^\star g) + d_{\mathcal{F}\Delta\mathcal{F}}(p_S(Z_g), p_T(Z_g)) && (7)\\ &\leq R_S(fg) + R_S(f^\star g) + d_{\mathcal{F}\Delta\mathcal{F}}(p_S(Z_g), p_T(Z_g)) && (8)\end{aligned}$$

The third term in the R.H.S of Eq. 4 can be bounded as

$$\begin{aligned} R_T(f^\star g, f^\star g^\star) &\leq R_S(f^\star g, f^\star g^\star) + |R_S(f^\star g, f^\star g^\star) - R_T(f^\star g, f^\star g^\star)| && (9)\\ &\leq R_S(f^\star g, f^\star g^\star) + \sup_{f \in \mathcal{F}, g, g' \in \mathcal{G}} |R_S(f'g, f'g') - R_T(f'g, f'g')| && (10)\\ &= R_S(f^\star g, f^\star g^\star) + d_{\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}}(p_S(X), p_T(X)) && (11)\\ &\leq R_S(f^\star g) + R_S(f^\star g^\star) + d_{\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}}(p_S(X), p_T(X)) && (12)\end{aligned}$$

Combine the above bounds, we have

$$R_T(fg) \leq R_S(fg) + d_{\mathcal{F}\Delta\mathcal{F}}(p_S(Z_g), p_T(Z_g)) + d_{\mathcal{F}_{\mathcal{G}\Delta\mathcal{G}}}(p_S(X), p_T(X)) + \lambda_{\mathcal{F}\mathcal{G}}(g) \tag{13}$$

where

$$\begin{aligned} \lambda_{\mathcal{F}\mathcal{G}}(g) &= 2R_S(f^\star g) + R_S(f^\star g^\star) + R_T(f^\star g^\star) && (14)\\ &= \inf_{f' \in \mathcal{F}, g' \in \mathcal{G}} 2R_S(f'g) + R_S(f'g') + R_T(f'g') && (15)\end{aligned}$$

□

References

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pp. 137–144, 2007.