Principles of Data Science Project 4 Domain Adaptation

Hongzhou Liu
517030910214
deanlhz@sjtu.edu.cn

Xuanrui Hong
517030910227
hongxuanrui.1999@sjtu.edu.cn

Qilin Chen
517030910155
1017856853@sjtu.edu.cn

***Abstract*—In this project, we tried different domain adaptation methods on the Office-Home dataset, which contains 65 categories of things from 4 domains. The four domains are Art, Clipart, Product and Real-World. In our experiments, we take Art, Clipart and Product as source domains and Real-World as target domain. For traditional methods, we tried KMM, CORAL, GFK, TCA, and EasyTL. For deep learning methods, we only tried DAN due to the scarce of computation resources and time limitation. We compared performances among those methods and discussed the difference among them.**

***Index Terms*—Domain Adaptation, Transfer Learning**

I. Introduction

In this project, we tried different unsupervised domain adaptation methods on the Office-Home dataset, which contains 65 categories of things from 4 domains. The four domains are Art, Clipart, Product and Real-World. There are two parts in this section. Firstly, we will introduce several traditional transfer learning methods we used in our project, including KMM, CORAL, GFK TCA and EasyTL. Then we will introduce deep transfer learning method DAN to compare with the traditional transfer learning methods.

A. Transfer Component Analysis (TCA)

For domain adaptation, Transfer Component Analysis (TCA) [1] tries to learn some transfer components across domains in a Reproducing Kernel Hilbert Space (RKHS) using Maximum Mean Discrepancy (MMD). It minimizes the distance between domain distributions by projecting data onto the learned transfer components.

The basic assumption of TCA is

$$P\left(X_s\right) \neq P\left(X_t\right)$$

where X_s denotes source domain data and $P\left(X_s\right)$ denotes its marginal distributions, X_t denotes target domain data and $P\left(X_t\right)$ denotes its marginal distributions. The motivation of TCA is to find a map Φ which could preserve the most data properties after projection, which means obtain the most variance, i.e.

$$P\left(\phi\left(\mathbf{x}_s\right)\right) \approx P\left(\phi\left(\mathbf{x}_t\right)\right)$$

or we can find conditional distribution of the two will also be similar as:

$$P\left(y_s \mid \phi\left(\mathbf{x}_s\right)\right)) \approx P\left(y_t \mid \phi\left(\mathbf{x}_t\right)\right))$$

We ccan give the maximum mean discrepancy (MMD) formula as:

$$MMD(X,Y) = \left\| \frac{1}{n_1}\sum_{i=1}^{n_1}\Phi\left(x_i\right) - \frac{1}{n_2}\sum_{j=1}^{n_2}\Phi\left(y_j\right)\right\|^2$$

where n_1, n_2 are the number of instances of the two domains. Then by changing the solution of this function to the solution of the kernel function, we can get:

$$\operatorname{Dist}\left(X_S', X_T'\right) = \operatorname{tr}(KL)$$

where

$$K = \left[\begin{array}{cc} K_{S,S} & K_{S,T} \\ K_{T,S} & K_{T,T} \end{array}\right] \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$$

, then, Q Yang [1] decomposed K to transfer this problem to:

$$\left\{\begin{array}{l} \min tr\left(W^T KLKW\right) + \mu tr\left(W^T W\right) \\ \text{s.t. } W^T KHKW = I_m \end{array}\right.$$

Finally, we can get the solution $W*$ as the m leading eigenvectors of

$$(KLK + \mu I)^{-1} KHK$$

B. Easy Transfer Learning (EasyTL)

Most traditional and deep learning migration algorithms are parametric methods, which require a lot of time and money to train those hyperparameters. In order to overcome these drawbacks, Easy Transfer Learning (EasyTL) [2] learns non-parametric transfer features through intra-domain alignment, and learns transmission classification through intra-domain programming. EasyTL can also improve the performance of existing TL methods through in-domain programming as the final classifier, the procedure of EasyTL can be shown in Fig. I-B.

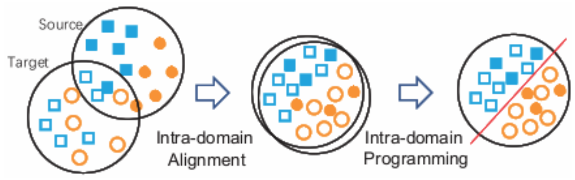


Fig. 1. procedure of EasyTL [2]

C. Deep Adaptation Network (DAN)

Recent research shows that deep neural networks can learn transferable features, which can be well extended to new fields to adapt to tasks. Deep Adaptation Network (DAN) use deep net to optimize the loss function and distribution distance in Regenerative Nuclear Hilbert Space (RKHS) [3].

Denote $\mathcal{H}_k$ as the reproducing kernel Hilbert space (RKHS) endowed with a characteristic kernel k. The average embedding of the distribution p in $\mathcal{H}_k$ is a unique element $k(p)$, making $\mathbf{E}_{\mathbf{x}\sim p} f(\mathbf{x}) = \langle f(\mathbf{x}), \mu_k(p)\rangle_{\mathcal{H}_k}$ for all $f \in \mathcal{H}_k$. Define the MK-MMD $d_k(p,q)$between the probability distributions p and q as the average embedding distance RKHS of p and q, and define the square formula of MK-MMD as

$$d_k^2(p,q) \triangleq \left\|\mathbf{E}_p\left[\phi\left(\mathbf{x}^s\right)\right] - \mathbf{E}_q\left[\phi\left(\mathbf{x}^t\right)\right]\right\|_{\mathcal{H}_k}^2$$

and the kernel defined by the multiple cores is

$$\mathcal{K} \triangleq \left\{k = \sum_{u=1}^{m} \beta_u k_u : \sum_{u=1}^{m} \beta_u = 1, \beta_u \geqslant 0, \forall u\right\}$$

Global optimization goal consists of two parts: loss function and distribution distance. The loss function is used to measure the difference between the predicted value and the true value.

DAN use adaptive method based on mk-mmd and CNNs to onercome that the target domain has no or only limited label information, so it is impossible to adapt CNN directly to the target domain through fine-tuning, or it is easy to overfit. Fig. I-C gives a description of the proposed DAN model.

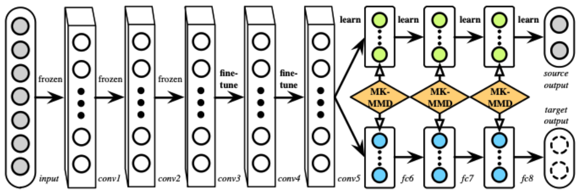


Fig. 2. The DAN architecture for learning transferable features.Since deep features eventually transition from general to specific along the network, (1) the features extracted by convolutional layers conv1–conv3 are general, hence these layers are frozen, (2) the features extracted by layers conv4–conv5 are slightly less transferable, hence these layers are learned via fine-tuning, and (3) fully connected layers fc6–fc8 are tailored to fit specific tasks, hence they are not transferable and should be adapted with MK-MMD. [3]

DAN fine-tuned the source of the labeled examples, requiring that under the hidden representation of the fully connected layers f $c6$ f $c8$, the distribution of the source and target becomes similar. This can be achieved by adding a multi-layer adaptive regularizer (1) based on mk-mmd to the risk (3) of CNN:

$$\min_{\Theta} \frac{1}{n_a} \sum_{i=1}^{n_a} J\left(\theta\left(\mathbf{x}_i^a\right), y_i^a\right) + \lambda \sum_{\ell=l_1}^{l_2} d_k^2\left(\mathcal{D}_s^{\ell}, \mathcal{D}_t^{\ell}\right)$$

where $\lambda > 0$ is a penalty parameter, l_1 and l_2 are layer indices between which the regularizer is effective.

II. Experiments

In this part, we will show the experimental results and comparative analysis of the results through two types of traditional transfer learning methods and deep transfer learning methods.

A. Transfer Component Analysis (TCA)

In this experiment, we test TCA method on kernels amount $[primal, linear, rbf]$, and various aim dimension in $[32, 64, 128, 256, 512, 1024, 2048]$, and different domain transfer tasks in [Art->RealWorld, Clipart->RealWorld, Product->RealWorld]. We can show our experimental results in Tab. . To figure out why TCA can have better performance, we plot the source domain data and target domain data in the same figure. Here, we can see the data distribution of case A-R, C-R and P-R in Fig. 3.

TABLE I
Accuracy of STFT features based on BOW(Z-Score) model

	Art->RealWorld			Clipart->RealWorld			Product->RealWorld		
32									
64									
128									
256									
512									
1024									
2048									

B. Easy Transfer Learning (EasyTL)

C. Deep Adaptation Network (DAN)

III. Conclusion

[4]

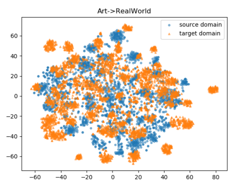
References

[1] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

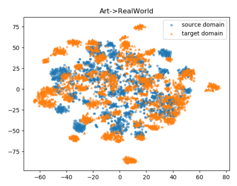
[2] J. Wang, Y. Chen, H. Yu, M. Huang, and Q. Yang, "Easy transfer learning by exploiting intra-domain structures," 2019.

[3] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," 2015.

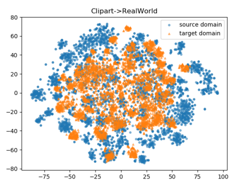
[4] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.



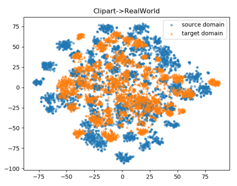
(a) Art, RealWorld distribution before TCA



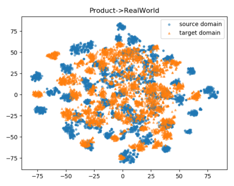
(b) Art, RealWorld distribution after TCA



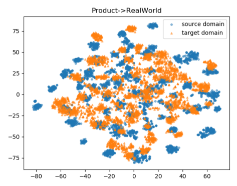
(c) Clipart, RealWorld distribution after TCA



(d) Clipart, RealWorld distribution after TCA



(e) Product, RealWorld distribution after TCA



(f) Product, RealWorld distribution after TCA

Fig. 3. TCA and baselines