Domain Adaptation for Image Classification

Sijia Li 518030910294

Abstract—Domain Adaptation is a subcategory of transfer learning which apply an algorithm trained in one or more source domains to a different but related target domain. In this project, we conduct several unsupervised traditional domain adaptation methods on Office-Home dataset with four domains and observe the performance gain brought by domain adaptation methods.

Index Terms—Transfer Learning, Domain Adaptation

1 Domain Adaptation

Transfer learning refers to through transferring labeled data or knowledge structures from a source domain, completing or improving the learning performance of the task on target domain . Common concepts involved in transfer learning include:

- **Domain (Domain).** Domain consists of data features and data distributions, and is the main body of learning. It is usually considered that the domain consists of two parts: the feature space of the input χ and the edge probability distribution $P(\mathbf{X})$ of the input $\mathbf{X}$
- **Source Domain.** The domain of existing data and knowledge.
- **Target Domain.** The domain to be studied.
- **Task.** Task: It consists of the objective function and the learning outcome, which is the result of learning.

Usually the source domain has more data and features, while the target domain has less data or features or none.

With transfer learning, a model can be trained on the source domain and then the resulting model can be used to learn on the unlabeled data on the target domain.

Although transfer learning presents good results in learning across domains by pre-training on the source task and then fine-tuning on the final target task, it has a prerequisite assumption that high precision knowledge structure transfer relies on the data distribution being similar between the source and target domains. However, there are many cases where the source and target domains have widely different data distributions, and then it is necessary to perform the transformation operation of domain adaptation on the data with different distributions on the source and target domains.

In this project, we will mainly use two traditional domain adaptation methods, i.e. TCA and CORAL on the image classification task.

2 TCA: Transfer Component Analysis

TCA [1] solves the domain adaptation problem by mapping the data from both source and target domains together into a high-dimensional regenerative kernel Hilbert space. In this space, the data distance between source and target is minimized while their respective internal properties are retained to the maximum extent.

We note the data on source and target domain is $\mathbf{X}_s$, $\mathbf{X}_t$, and try to find a common latent representation for them by the transformation ϕ.The authors define the distance between $\mathbf{X}_s$ and $\mathbf{X}_t$ with MMD (Maximum Mean Discrepancy), i.e.

$$\operatorname{dist}(\mathbf{X}_s', \mathbf{X}_t') = ||\frac{1}{n_1}\sum_{i=1}^{n_1}\phi(x_{s_i}) - \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(x_{t_i})||^2$$

Pan et al [2], with the help of kernel function trick, transformed the problem into a kernel learning problem by expressing the distance as

$$d(\mathbf{X}_s, \mathbf{X}_t)^2 = tr(\mathbf{KL}),$$

in which $\mathbf{K} = \begin{bmatrix} \mathbf{K}_{ss} & \mathbf{K}st \\ \mathbf{K}_{ts} & \mathbf{K}_{tt} \end{bmatrix}$, $\mathbf{K}_{ss}, \mathbf{K}_{tt}, \mathbf{K}_{st}$ denote the kernel matrix defined by $\mathbf{K}$ over the source, target and cross domains, respectively $L_{i,j} = \begin{cases} \dfrac{1}{n_s^2}, \text{if } i,j \in S \\ \dfrac{1}{n_t^2}, \text{if } i,j \in T \\ -\dfrac{1}{n_s n_t}, \text{otherwise} \end{cases}$

The kernel matrix $\mathbf{K}$ can be factorized as

$$\mathbf{K} = (\mathbf{K}\mathbf{K}^{-\frac{1}{2}})(\mathbf{K}^{-\frac{1}{2}}\mathbf{K}) = \hat{\phi}(\mathbf{X}_{st})^T\hat{\phi}(\mathbf{X}_{st})$$

Introduce transformation matrix $\hat{\mathbf{W}}$ for fake representation $\hat{\phi}(\mathbf{X}_{st})$

$$\hat{\mathbf{K}} = (\hat{\phi}(\mathbf{X}_{st})^T\hat{\mathbf{W}})(\hat{\mathbf{W}}^T\hat{\phi}(\mathbf{X}_{st})) = \mathbf{K}\mathbf{W}\mathbf{W}^T\mathbf{K},$$

in which $\mathbf{W} = \mathbf{K}^{-\frac{1}{2}}\hat{\mathbf{W}}$

We rewrite the distance between source domain and target domain as

$$tr(\hat{\mathbf{K}}\mathbf{L}) = tr(\mathbf{K}\mathbf{W}\mathbf{W}^T\mathbf{K}\mathbf{L}) = tr(\mathbf{W}^T\mathbf{K}\mathbf{L}\mathbf{K}\mathbf{W})$$

We need to normalize $\operatorname{tr}\left(W^\top W\right)$ to control the complexity of $\mathbf{W}$, and the kernel learning problem can be:

$$\min_{\mathbf{W}} tr(\mathbf{W}^T\mathbf{K}\mathbf{L}\mathbf{K}\mathbf{W}) \ \text{ s.t. } \mathbf{W}^T\mathbf{W} = \mathbf{I}$$

This can be represented as

$$\min_{\mathbf{W}} tr(\mathbf{W}^T\mathbf{K}\mathbf{L}\mathbf{K}\mathbf{W}) + \mathbf{W}^T\mathbf{W}$$

$$\text{s.t. } \mathbf{W}^T\mathbf{K}\mathbf{H}\mathbf{K}\mathbf{W} = \mathbf{I}$$

The final result is

$$\mathbf{H} = \mathbf{I} - \frac{1}{n_s + n_t}\mathbf{1}\mathbf{1}^T$$

Algorithm 1 CORAL for Unsupervised Domain Adaptation

Input: Source Data D_S, Target Data D_T
Output: Adjusted Source Data D_s^*
$C_S = cov(D_S) + eye(size(D_S, 2))$
$C_T = cov(D_T) + eye(size(D_T, 2))$
$D_S = D_S * C_S^{\frac{-1}{2}}$ % whitening source
$D_S^* = D_S * C_T^{\frac{1}{2}}$ % re-coloring with target covariance

Fig. 1. CORAL Algorithm

After obtaining $\mathbf{W}$, use the new kernel $\mathbf{K} = \mathbf{K}\mathbf{W}\mathbf{W}^T\mathbf{K}$

3 CORAL: Correlation Alignment

CORAL [3] minimizes domain shift by aligning the second-order statistics of source and target distributions.

The method transforms source features in order to minimize the Frobenius norm between the correlation matrix of the input target data and the one of the transformed input source data.

The source features transformation is described by the following optimization problem [4]:

$\mathbf{C}_s$ and $\mathbf{C}_t$ are covariance matrices of $\mathbf{X}_s$ and $\mathbf{X}_t$
$\hat{\mathbf{C}}_s$: : covariance matrix of $\mathbf{A}^T\mathbf{X}_s$

$$\min_{\mathbf{A}} ||\hat{\mathbf{C}}_s - \mathbf{C}_t||_F^2$$
$$= \min_{\mathbf{A}} ||\mathbf{A}^T\mathbf{C}_s\mathbf{A} - \mathbf{C}_t||_F^2$$

where

$$\mathbf{C}_s = \mathbf{U}_s \sum\nolimits_s \mathbf{U}_s^T,\ \mathbf{C}_t = \mathbf{U}_t \sum\nolimits_t \mathbf{U}_t^T$$

The optimal solution:

$$\mathbf{A}^* = \mathbf{U}_s \sum\nolimits_s^{1\frac{1}{2}} \mathbf{U}_s^T \mathbf{U}_t[1:r] \sum\nolimits_t [1:r]^{\frac{1}{2}} \mathbf{U}_t[1:r]^T,$$

where

$$r = \min(rank(\mathbf{C}_s), rank(\mathbf{C}_t))$$

After obtaining $\mathbf{A}^*$, use $\mathbf{A}^{*T}\mathbf{X}_s$ and $\mathbf{X}_t$
The algorithm is simple to carry out as Algorithm1:

4 Experiments

4.1 Dataset

Office-Home dataset is a domain adaptation dataset, which consists of 65 categories of office depot from four domains (i.e., A: Art, C:Clipart, P:Product, R: Real-world).

4.2 Results

4.3 Baseline

We first use SVM with linear and rbf kernel, setting C as 0.001, 0.01, 0.1, 1, 5, 10. The best result and corresponding parameter settings are as below:

domain	acc	kernel	C
A → R	0.747303	rbf	5
C→ R	0.656168	linear	0.01

TABLE 1
Baseline

4.4 TCA

For TCA, I test three kinds of kernel: primal, linear and rbf with different dimensions. The result shows that when we select primal kernel with suitable dimension, the accuracy improves compared to baseline.

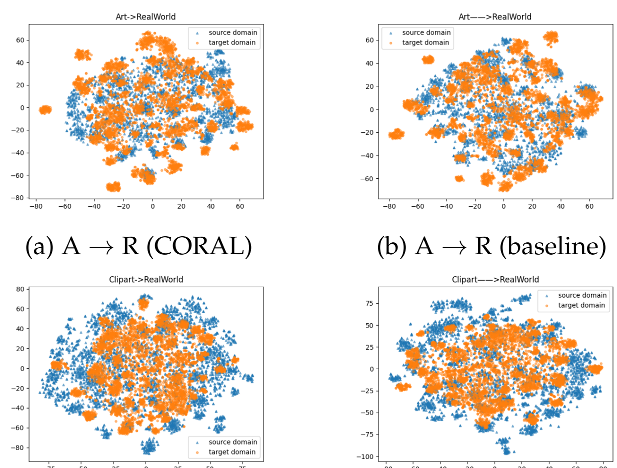
dim	A → R			C → R		
	primal	linear	rbf	primal	linear	rbf
16	0.6457	0.6408	0.6397	0.5669	0.5658	0.5869
32	0.7159	0.6982	0.7074	0.6445	0.6261	0.6339
64	0.7528	0.7289	0.7372	0.6592	0.6371	time out
256	**0.7574**	**0.7340**	0.7416	**0.6612**	**0.6385**	time out
512	0.7558	0.7340	**0.7429**	0.6612	0.6381	time out
1024	0.7553	0.7340	0.7423	0.6610	0.6381	time out

4.5 CORAL

The code is simple and there is no need to set any hyperparameter on the CORAL algorithm. However, we observe that on A → R and C→ R , the accuracy after applying CORAL are both lower than the baseline.

domain	CORAL acc	baseline acc
A → R	0.738123	0.747303
C→ R	0.651366	0.656185

TABLE 2
CORAL Accuracy



(a) A → R (CORAL) (b) A → R (baseline)
(c) C→ R (CORAL) (d) C→ R (baseline)

Fig. 2. Example of placing a figure with experimental results.

We carried out the visualization experiments to observe the data distribution but no obvious difference is found. We guess that CORAL may be more suitable for simpler dataset and the pictures are too complex for it to handle. In the future, we can consider Deep CORAL.

5 Conclusion and Summary

In this project, we review the definition of two typical distance metrics: TCA and CORAL. In our dataset, TCA has an improvement when selecting primal kernel with 128 dimension. However, for CORAL, we have no observation about the good performance, which inspires us to try Deep CORAL in the future.

As a senior, I was unable to take this course last year because of conflicts with other courses. However, after listening to my classmates' experiences, I am very interested in the course content, so I participated in the Principles of Data Science course at the end of my senior year. This semester, I interned at Tencent and completed my graduate thesis. My thesis design was also related to domain adaptation based on deep learning, but the related research area was not computer vision but recommendation systems. The related experiments have achieved excellent performance on Tencent's industrial dataset. Both experiences, the principles of data science course and the graduation design, have strengthened my understanding and application of domain adaptation methods.

References

[1] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," in *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, C. Boutilier, Ed., 2009, pp. 1187–1192. [Online]. Available: http://ijcai.org/Proceedings/09/Papers/200.pdf

[2] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, D. Fox and C. P. Gomes, Eds. AAAI Press, 2008, pp. 677–682. [Online]. Available: http://www.aaai.org/Library/AAAI/2008/aaai08-108.php

[3] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, D. Schuurmans and M. P. Wellman, Eds. AAAI Press, 2016, pp. 2058–2065. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12443

[4] "Adapt package document," https://adapt-python.github.io/adapt/generated/adapt.feature_based.CORAL.html#adapt.feature_based.CORAL.