Domain Adaptation for Image Classification

CS245 Project 4

Xiang Gu
5130309729

Tao Zhu
515020910243

Weihong Lin
515030910643

Chacha Chen
515021910302

1. Dataset

The Office-Home dataset (Venkateswara et al. 2017) has been created to evaluate domain adaptation algorithms for object recognition using deep learning. It consists of images from 4 different domains: Artistic images, Clip Art, Product images and Real-World images. For each domain, the dataset contains images of 65 object categories found typically in Office and Home settings.

Figure 1
The Office-Home 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). The raw images can be downloaded from http://hemanthdv.org/OfficeHome-Dataset/. The 2048-dim ResNet50 deep learning features of all images can be downloaded from https://pan.baidu.com/s/1qvcWJCXVG8JkZnoM4BVoGg#list/path=%2F.

2. Methods

For traditional domian adaptation methods, we choose Transfer Component Analysis (TCA), CORAL and Geodesic Flow Kernel (GFK).

2.1 Transfer Component Analysis (TCA)

TCA (Pan et al. 2010) learns transfer components across domains in a Reproducing Kernel Hilbert Space (RKHS) using Maximum Mean Discrepancy (MMD). In the subspace spanned by these transfer components, data distributions in different domains are close to each other. As a result, with the new representations in this subspace, standard machine learning methods can be applied to train classifiers or regression models in the source domain for use in the target domain.

2.2 CORAL

CORAL (Sun and Saenko 2016) is a unsupervised domain adaptation method that aligns the second-order statistics of the source and target distributions with a linear transformation.

2.3 Geodesic Flow Kernel (GFK)

GFK (Gong et al. 2012) is a kernel-based method that takes advantage of such structures. The geodesic flow kernel models domain shift by integrating an infinite number of subspaces that characterize changes in geometric and statistical properties from the source to the target domain. This approach is computationally advantageous, automatically inferring important algorithmic parameters without requiring extensive crossvalidation or labeled data from either domain. It is also introduced in the paper that there is a metric that reliably measures the adaptability between a pair of source and target domains. For a given target domain and several source domains, the metric can be used to automatically select the optimal source domain to adapt and avoid less desirable ones.

2.4 DeepCORAL – a Deep Domain Adaptation Method

DeepCORAL (Sun and Saenko 2016) is an extended verson of CORAL which is to learn a nonlinear transformation that aligns correlations of layer activations in deep neural networks.

3. Experiments Setting

3.1 Data preprocessing

For tradditional methods, PCA (Principal Component Analysis) is used to reduce the dimensionality, due to the tremendous long running time. A variant of 95% is retained after PCA pre-processing. For deep learning methods, however, raw data is directly feed.

3.2 Model Parameters

- TCA
  - kernel ='linear'
  - dim = 30
  - lamb =1
  - gamma =1
- CORAL
- GFK
  - dim=20
  - epsilon = e^{-20}

- DeepCORAL
  - batch size = 32
  - number of epochs = 200 with early stopping
  - learning rate = 0.01
  - momentum = 0.9
  - L2 decay rate = 5e-4

4. Experiments Results

From Table 4, the conclusion could be drawn clearly that among all the three traditional methods, CORAL is the best. However, deep learning method, DeepCORAL, far outperforms the other three tradditional methods.

Table 1
Overall Performance of different methods on Office-home Dataset w.r.t different source domains

Methods	Art_RealWorld	Clipart_RealWorld	Product_RealWorld
TCA	0.6933	0.6164	0.6839
CORAL	**0.7484**	**0.6593**	**0.7342**
GFK	0.6745	0.5914	0.6839
DeepCORAL	**0.7608**	**0.6879**	**0.7517**

Below three tables, e.g. Table 2, Table 3 and Table 4, are performance comparison w.r.t different classifiers, the conclusion still holds for all the experiment settings.

Table 2
Performance of TCA on Office-home Dataset w.r.t different classifiers

Source_Target	Classifier	Accuracy
Art_RealWorld	Nearest Neighbors	0.6274
	Linear SVM	0.6862
	RBF SVM	**0.6933**
	Extra Tree	0.5632
	Neural Net	0.4459
Clipart_RealWorld	Nearest Neighbors	0.5691
	Linear SVM	**0.6164**
	RBF SVM	0.6132
	Extra Tree	0.5030
	Neural Net	0.5237
Product_RealWorld	Nearest Neighbors	0.6343
	Linear SVM	0.6538
	RBF SVM	**0.6839**
	Extra Tree	0.6630
	Neural Net	0.6665

Table 3
Performance of CORAL on Office-home Dataset w.r.t different classifiers

Source_Target	Classifier	Accuracy
Art_RealWorld	Nearest Neighbors	0.6571
	Linear SVM	0.7381
	RBF SVM	**0.7484**
	Extra Tree	0.6408
	Neural Net	0.7321
Clipart_RealWorld	Nearest Neighbors	0.5825
	Linear SVM	0.6463
	RBF SVM	**0.6593**
	Extra Tree	0.5944
	Neural Net	0.6132
Product_RealWorld	Nearest Neighbors	0.6883
	Linear SVM	0.7176
	RBF SVM	**0.7342**
	Extra Tree	0.7004
	Neural Net	0.7082

Table 4
Performance of GFK on Office-home Dataset w.r.t different classifiers

Source_Target	Classifier	Accuracy
Art_RealWorld	Nearest Neighbors	0.6036
	Linear SVM	0.6437
	RBF SVM	0.6660
	Extra Tree	0.6591
	Neural Net	**0.6745**
Clipart_RealWorld	Nearest Neighbors	0.5438
	Linear SVM	0.5647
	RBF SVM	0.5907
	Extra Tree	0.5526
	Neural Net	**0.5914**
Product_RealWorld	Nearest Neighbors	0.6343
	Linear SVM	0.6538
	RBF SVM	**0.6839**
	Extra Tree	0.6630
	Neural Net	0.6665

4.1 Running Time Comparison

In table 5, it can be seem that TCA took a tremendous time to train while the other 2 methods are super fast.

Table 5
Running time comparison of different domain adaptation methods

Methods	Average running time
TCA	1569.286s
CORAL	0.540s
GFK	0.764s

Table 6
Performance of DeepCORAL on Office-home Dataset w.r.t different source domains

Methods	Art_RealWorld	Clipart_RealWorld	Product_RealWorld
DeepCORAL	0.7608	0.6879	0.7517

5. Addendum

The code of our project could be downloaded at https://github.com/Xiang-Gu/CS245_Principle_Data_Science

References

Gong, Boqing, Yuan Shi, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, IEEE.

Pan, Sinno Jialin, Ivor W Tsang, James T Kwok, and Qiang Yang. 2010. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210.

Sun, Baochen and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pages 443–450, Springer.

Venkateswara, Hemanth, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. 2017. Deep hashing network for unsupervised domain adaptation. In *(IEEE) Conference on Computer Vision and Pattern Recognition (CVPR)*.