Deep Domain Adaptation by Geodesic Distance Minimization

Yifei Wang, Wen Li, Dengxin Dai, Luc Van Gool
EHT Zurich
Ramistrasse 101, 8092 Zurich
yifewang@ethz.ch {liwen, dai, vangool}@vision.ee.ethz.ch

Abstract

In this paper, we propose a new approach called Deep LogCORAL for unsupervised visual domain adaptation. Our work builds on the recently proposed Deep CORAL method, which aims to train a convolutional neural network and simultaneously minimize the Euclidean distance of convariance matrices between the source and target domains. By observing that the second order statistical information (i.e. the covariance matrix) lies on a Riemannian manifold, we propose to use the Riemannian distance, approximated by Log-Euclidean distance, to replace the naive Euclidean distance in Deep CORAL. We also consider first-order information, and minimize the distance of mean vectors between two domains. We build an end-to-end model, in which we minimize both the classification loss, and the domain difference based on the first-order and second-order information between two domains. Our experiments on the benchmark Office dataset demonstrates the improvements of our newly proposed Deep LogCORAL approach over the Deep CORAL method, as well as the further improvement when optimizing both orders of information.

1. Introduction

One of the most fundamental assumption in traditional machine learning is that the training data and the test data have identical distributions. However, this may not always hold for real-world visual recognition applications. The limitations of collecting training data, and the large variance of test data in real-world applications make it difficult to guarantee that training and test data follow an identical distribution. As a result, the performance of the visual recognition model can significantly drop due to the distribution mismatch between training and test data. This is known as the ”domain adaptation” problem [5] [9] [4] [6] [12] [13] [21] [18] [10] [7] [17]. Torrabla and Efros [22] pointed out that each existing visual recognition dataset more or loss has its own bias, which reveals the common existence of visual domain adaptation problems.

In visual domain adaptation, the domain of the training data is referred to as the source domain, and the domain of the test data is referred to as the target domain. Visual domain adaptation aims to reduce the distribution mismatch between these two domains, such that the performance of visual recognition models learned from the source domain can be improved when testing on the target domain. Typically, the source domain contains a large number of labeled data for training the models, whereas the target domain contains only unlabeled data. Visual domain adaptation has attracted more and more attentions from computer vision researchers in recent years. It becomes even more important after the revival of Convolutional Neural Network (CNN), because CNN usually requires a large number of labeled training data to build a robust model, and it can be expensive to annotate a large number of training data which have an identical distribution as the test data.

A few papers have proposed unsupervised visual domain adaptation based on CNNs [7][21][16][8][23]. The recent Deep CORAL method was proposed to reduce the domain difference by minimizing the Euclidean distance between the covariance matrices in the source and target domains [20]. They built an end-to-end model, in which they simultaneously minimized the classification loss and the domain difference. While the Deep CORAL method improves the classification performance of CNN, it is still unclear if the naive Euclidean distance a good choice for minimizing the distance of two covraince matrices. Moreover, only the second-order statistical information (*i.e.*, the covariance matrix) is used in Deep CORAL, and other information is discard.

To cope with the first issue, we propose a new Deep LogCORAL approach which employs the geodesic distance to replace the naive Euclidean distance in Deep CORAL. Intuitively, the covariance matrix is a positive semi-definite (PSD) matrix, which lies on a Riemannian manifold. The source and target covariance matrices can be deemed as two points on the Riemannian manifold, so a more desirable metric is the geodesic distance between the two points on the Riemannian manifold. As inspired by [3], we employ

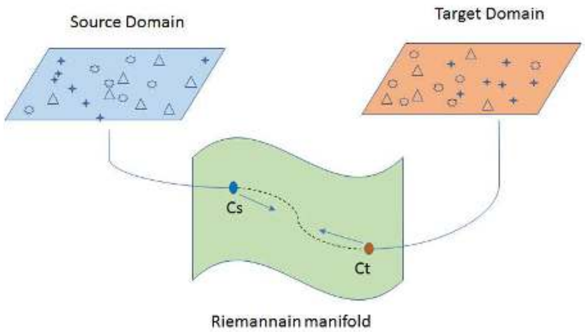


Figure 1: Illustration of our proposed domain adaptation method: minimizing geodesic distance between two domains in Riemannian manifold.

the LogEuclidean distance, which has been widely used for calculating the geodesic distance between PSD matrices. As shown in Figure 1, C_s and C_t represent the covariance matrices of the source and target domains, respectively. The crosses, triangles and stars in the blue (*resp.*, orange) rectangle denote the training samples of different classes from the source (*resp.*, target) domain. By minimizing the Log-Euclidean distance instead of the naive Euclidean distance between these two domains, we expect the domain shift between two domains can be reduced smoothly. We designed a new Log-Euclidean loss, which is integrated into the CNN for end-to-end training.

To cope with the second issue, we also exploit the first-order statistical information. We propose to minimize the distance between the mean vectors of two domains, which is closely related to the Maximum Mean Discrepancy (MMD) theory. We simultaneously minimize the mean distance and the Log-Euclidean distance, such that both the first-order and the second-order statistical information of two domains become consistent.

We conduct extensive experiments on the benchmark Office dataset, which shows a clear improvement of our newly proposed Deep LogCORAL when compared with the Deep CORAL method. We also demonstrate that both the first-order and the second-order information are necessary for effectively reducing the domain shift.

2. Related Work

For a comprehension summary for domain adaptation we refer readers to [17]. In [22][21] the concept of dataset bias was well introduced and attracted a lot of attention. Since then, many methods have been developed in order to overcome the build-in dataset bias.

Early domain adaptation method like [18] requires learning a regularized transformation, by using information-theoretic metric learning that maps data in the source domain to the target domain. However this method require labeled data from target domain which in many scenarios is unknown to testers.

Later unsupervised domain adaptation method [10][9][6][19][3] tried to improve the performance on the target domain by transferring knowledge from the target domain to the source domain without the need of target labels. In [10] it first extracts features that invariant to domain change then it models the different domains as points on Grassmann manifold and generate number of subspaces in between to train classifier on those subspaces. Similarly, to cut back the difference between source domain and test domain [3] generate subspaces in Riemannian manifold and [19] measure distance in Euclidean distance.

More relevant to this paper is domain adaptation method applied on CNN. The DLID method [2] is inspired by [10] to capture information from an ”interpolating path” between the source domain and the target domain. Instead of optimizing the representation to minimize some measure of domain shift such as geodesic distance, DRCN[8] and ADDA [23] alternatively reconstruct the target domain from the source representation. Gradient reversal method [7] tries to obtain a feed-forward net-work having the same or very similar distributions in the source and the target domains, while RTN [16] also wants to adapt target classifiers to the source classifiers by learning a residual function. In [24] and [14], new CNN architectures are proposed, in which [24] introduces an adaptation layer and an additional main confusion loss to learn a representation while in [14] all task-specific layers are embedded in a reproducing kernel Hilbert space.

Our work is mostly inspired by [20], which adds a new CORAL loss that calculates the Euclidean distance of two domain's covariance matrices before the softmax layer. This method looks very simple but the result shows it exceeds other methods that appears to be more complex. Another related paper is about symmetric positive definite (SPD) matrix learning [11], that presents a new direction of SPD matrix non-linear learning in the deep neural network model.

3. Methodology

In this section, we present our newly proposed Deep LogCORAL approach. We first give a brief review of the Deep CORAL method, and then introduce our Deep Log-CORAL layer as well as the mean layer.

3.1. Deep CORAL approach

As described in [20], CORAL loss is built to calculate the distance of second-order statistical information between two domains. It first calculates the covariance matrix of the features extracted from the ”fc8” layer for each domain,

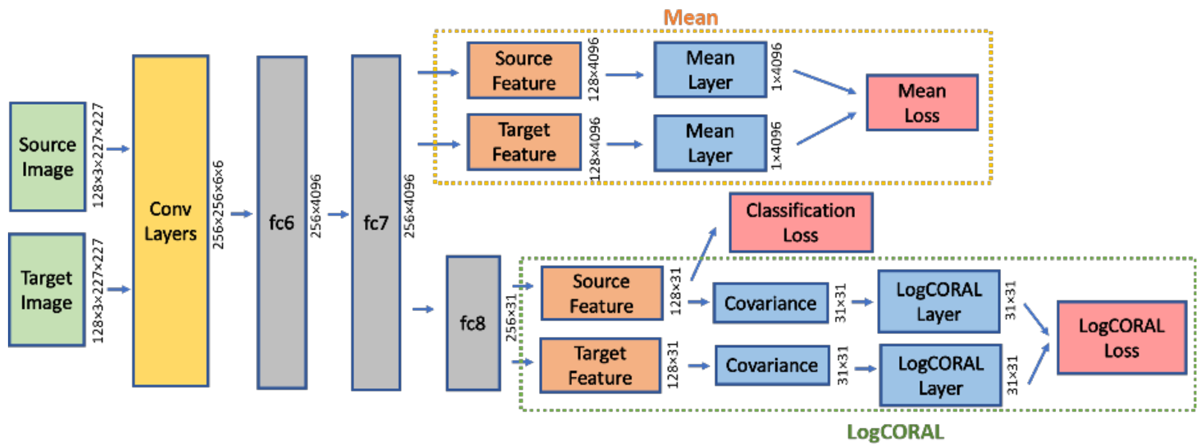


Figure 2: Structure of the model.

and then minimizes the Euclidean distance of the covariance matrices of two domains.

Formally, let us denote by $\mathbf{D_S} = [\mathbf{x_1}, \ldots, \mathbf{x_{ns}}]$ as the source domain features that are extracted out of "fc8" layer, in which $\mathbf{x_i}$ is the i-th source sample with d being the dimension of features. Similarly, $\mathbf{D_T} = [\mathbf{u_1}, \ldots, \mathbf{u_{n_T}}]$ denotes the target domain features extracted from the "fc8" layer, in which $\mathbf{u_i}$ is the i-th target sample. The CORAL loss is then defined as follows,

$$L_{CORAL} = \frac{1}{4d^2} \|\mathbf{C_S} - \mathbf{C_T}\|^2, \tag{1}$$

in which the covariance matrices $\mathbf{C_S}$ and $\mathbf{C_T}$ are given as follows:

$$\mathbf{C_S} = \frac{1}{n_S - 1}(\mathbf{D_S}^T\mathbf{D_S} - \frac{1}{n_S}(\mathbf{1}^T\mathbf{D_S})^T(\mathbf{1}^T\mathbf{D_S})) \tag{2}$$

$$\mathbf{C_T} = \frac{1}{n_T - 1}(\mathbf{D_T}^T\mathbf{D_T} - \frac{1}{n_T}(\mathbf{1}^T\mathbf{D_T})^T(\mathbf{1}^T\mathbf{D_T})) \tag{3}$$

where n_S, n_T is the batch size of the source domain and target domain respectively. d is the feature dimension, and $\mathbf{1}^T$ is a vector that all elements equals to 1.

The Deep CORAL method simultaneously minimizes the above loss and the classification loss, such that the domain distribution mismatch is minimized, and the discriminative ability is also preserved.

3.2. Deep LogCORAL approach

To push the classifier closer to target domain, an essential problem is to precisely model the distance between two domains. Deep CORAL use squared Euclidean distance to measure distance, while evidence shows that measure distance on other manifolds such as Riemannian manifold may be more precise in measuring matrix distance and can get better result in domain adaptation [11]. According to this assumption, we design a new LogCORAL layer after "fc8" to measure distance in Riemannian manifold.

Covariance matrix is a symmetric positive semi-definite (PSD) matrix, but adding a small ϵ to the eigenvalues of covariance matrix transforms it into SPD without significantly change its property. Therefore after getting covariance matrices from the source and target domains we can calculate the distance on Riemannian manifold.

Log-Euclidean Riemannian metric: Log-Euclidean metrics was first proposed in [1]. It has the capacity to endow Riemannian manifold and also demonstrated that the swelling effect which is clearly visible in the Euclidean case disappears in Riemannian cases. Logarithm operation on the eigenvalue of PSD matrices make the manifold to be flat, and then Euclidean distance can be calculated on this flat space, which makes it much easier to caculate the geodesic distance.

LogCORAL Forward Propagation: The input of this layer is the covariance matrices calculated in Equation (2) and (3). The forward process can be easily calculated by using singular value decomposition (SVD) to get the eigenvalues and eigenvectors of the covariance metrics, followed by applying logarithm operation on eigenvalues. The LogCORAL loss is defined as the Euclidean distance between the logarithm of covariance matrices:

$$L_{LogCORAL} = \frac{1}{4d^2} \|log(\mathbf{C_S}) - log(\mathbf{C_T})\|^2, \tag{4}$$

where the $log()$ operation is the logarithm of the PSD matrix. We take the source domain covariance matrix $\mathbf{C_S}$ as an example. Let us denote the eigen-decomposition of $\mathbf{C_S}$ as $\mathbf{C_S} = \mathbf{U_S}\mathbf{\Sigma_S}\mathbf{U_S}^T$, then the Logarithm operator is defined as $log(\mathbf{C_S}) = \mathbf{U_S}log(\mathbf{\Sigma_S})\mathbf{U_S}^T$, where $log(\mathbf{\Sigma_S})$ is calculated by applying the Logarithm operator on the diagonal elements of $\mathbf{\Sigma_S}$. The same procedure is applied to the target domain covariance matrix.

Table 1: Accuracy comparison for the CNN (without adaptation), CORAL (baseline adaptation) and combine method (extended adaptation method combined with LogCORAL and mean model). Note that A: amazon, W:webcam, D: DSLR, A-W means use amazon as source domain and use webcam as target domain (analogous for the rest).

Accuracy	A - W	D - W	A - D	W - D	W - A	D - A	AVG
CNN	63.34±0.88	95.21±0.52	65.14±1.26	99.26±0.06	49.23±0.22	51.37±0.48	70.59
CORAL	66.12±0.45	95.24±0.49	66.38±2.54	99.24±0.10	50.71±0.24	**53.12±0.69**	71.80
Ours (LogCORAL+Mean)	**70.15±0.57**	**95.45±0.07**	**69.41±0.51**	**99.46±0.31**	**51.57±0.46**	51.15±0.32	**72.87**

LogCORAL Back Propagation: Back-propagation can be derived following the technique described in [11]. For simplicity, we denote $\mathbf{C_{S'}} = log(\mathbf{C_S})$ and $\mathbf{C_{T'}} = log(\mathbf{C_T})$. Taking the source domain as an example, the gradients can be derived as follows:

$$\frac{\partial L_{LogCORAL}}{\partial \mathbf{C_S}} = \frac{1}{2d^2}(\mathbf{C_{S'}} - \mathbf{C_{T'}})\frac{\partial \mathbf{C_{S'}}}{\mathbf{C_S}}, \tag{5}$$

where

$$\frac{\partial \mathbf{C_{S'}}}{\mathbf{C_S}} = \mathbf{U_S}(\mathbf{P}^T \circ (\mathbf{U_S}^T d\mathbf{U_S}))_{sym}\mathbf{U_S}^T + \mathbf{U_S}(d\mathbf{\Sigma_S})_{diag}\mathbf{U_S}^T \tag{6}$$

$$d\mathbf{U_S} = 2(\frac{\partial L_{LogCORAL}}{\partial \mathbf{C_{S'}}})_{sym}\mathbf{U_S} log(\mathbf{\Sigma_S}) \tag{7}$$

$$d\mathbf{\Sigma_S} = \mathbf{\Sigma_S}^{-1}\mathbf{U_S}^T(\frac{\partial L_{LogCORAL}}{\partial \mathbf{C_{S'}}})_{sym}\mathbf{U_S} \tag{8}$$

$$\mathbf{P(i,j)} = \begin{cases} \frac{1}{\sigma_i - \sigma_i}, i \neq j, \\ 0, i = j \end{cases} \tag{9}$$

where $\circ$ is Hadamard product, *i.e*., element wise product, sym operation is defined as $\mathbf{A_{sym}} = \frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$, $diag$ operator is to keep only diagonal values of $\mathbf{A}$ and set the rests to zeros, and σ_i denotes the i-th eigenvalue in $\mathbf{\Sigma_S}$. For the target domain, the calculation is the same except the sign is negative.

After implementing the LogCORAL layer, we can build our Deep LogCORAL structure. As in Deep CORAL, we have source domain with label and target domain without label as shown in the green rectangular in Figure 2.

3.3. Mean Layer

Note that we have considered the second-order statistic information between two domains, intuitively we can also use the first order statistic information, the mean value, as it is also one type of the representative information for a dataset. As shown in the green rectangular of Figure 2. We create a mean layer after fc7 which calculates the mean value along the column to get a mean vector of features. Then we calculate the Euclidean distance of the mean vectors between two domains as mean loss. Definition is shown as follows:

$$\mathbf{L_{Meanloss}} = \frac{1}{2d}\|\mathbf{1}^T\mathbf{D_S} - \mathbf{1}^T\mathbf{D_T}\|^2 \tag{10}$$

By incorporating the mean loss into the model, now there are three losses: classification, LogCORAL and mean losses to be optimized. The whole structure is shown in Figure 2.

4. Experiment and Discussion

4.1. Experimental settings

For a fair comparison, we follow the experimental setting in Deep CORAL, *i.e.* using the Office dataset and the ImageNet pretrained AlexNet model. The Office dataset contains images collected from three domains: *Amazon, DSLR, Webcam*, each has identical thirty one categories. We use one of the three domains as the source domain, and one of the rest two as the target domain, leading to six cases in total.

To make the training procedure more stable, moving average is employed when computing the losses. We use accumulated covariance and mean value to calculate LogCORAL loss and mean loss:

$$\mathbf{C} = 0.9 * \widetilde{\mathbf{C}} + 0.1 * \mathbf{C_{batch}} \tag{11}$$

$$\mathbf{M} = 0.9 * \widetilde{\mathbf{M}} + 0.1 * \mathbf{M_{batch}}, \tag{12}$$

where $\mathbf{C}$ is the moving average covariance matrix, $\widetilde{\mathbf{C}}$ is the accumulated covariance from last iteration, and $\mathbf{C_{batch}}$ is the covariance matrix from the current batch. The terms for mean value are similarly defined as above.

4.2. Experimental results

We compare our proposed model with the Deep CORAL method, and also include the CNN method as a baseline, in which we fine-tune the pretrained AlexNet model using source domain samples without considering domain adaptation.

The experimental results are shown in Table 1. After applying our combination model (LogCORAL+Mean), for each domain shift, five out of six shifts reach the highest

Table 2: Accuracy comparison for the Mean, LogCORAL and combined model (combined with LogCORAL and mean model).

Accuracy	A - W	D - W	A - D	W - D	W - A	D - A	AVG
Mean	66.29±0.74	**95.56±0.19**	68.67±0.46	99.51±0.23	49.83±0.85	50.74±0.74	71.77
LogCORAL	68.83±0.57	95.23±0.15	68.64±1.41	**99.52±0.41**	50.94±0.28	**51.73±0.61**	72.48
LogCORAL+Mean	**70.15±0.57**	95.45±0.07	**69.41±0.51**	99.46±0.31	**51.57±0.46**	51.15±0.32	**72.87**

accuracy. The average accuracy raised 2.28% and 1.07% compared to CNN and Deep CORAL.

We further conduct an ablation study as shown in Table 2. Optimizing the mean loss or the LogCORAL loss individually could gain performance improvements over the baseline CNN method, which shows the first-/second-order information is useful for domain adaptation. We also observe that optimizing LogCORAL outperforms the Deep CORAL method in Table 1 in terms of average accuracy, which demonstrates the effectiveness of minimizing geodesic distance instead of using simple Euclidean distance on covariance matrices. Combing the mean loss and logCORAL loss gives further improvements in general, and raises the average accuracy by 0.39%.

We take A-W as an example to show the learning curves in Figure 3. We observe that our proposed model converge fast. Moreover, minimizing the mean loss or the Deep LogCORAL loss individually improves the test accuracy compared to the baseline CNN method, and combining two losses further improves the accuracy. Furthermore, we

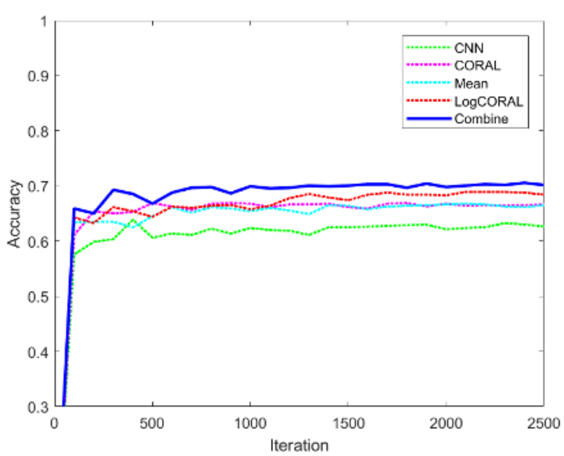


Figure 3: Comparison of learning curve of the models on A-W domain shift.

also conduct additional experiment by combining the mean loss with the Deep CORAL loss, which improves the average accuracy from 71.80% to 71.94%. This again verifies our motivation that it is beneficial to use both first-order and second-order statistical information for domain adaptation. However, this result is still worse than our final approach (logCORAL+Mean). We attribute this to the usage of geodesic distance on second order covariance matrices in Riemannian space which has only weak correlation with first order Euclidean information. We will demonstrate this in the following session.

4.3. Visualization

After demonstrating that minimizing the LogCORAL and mean losses individually helps shorten the distance between the two domains, we further investigate if the Euclidean and Riemannian distances correlated to each other.

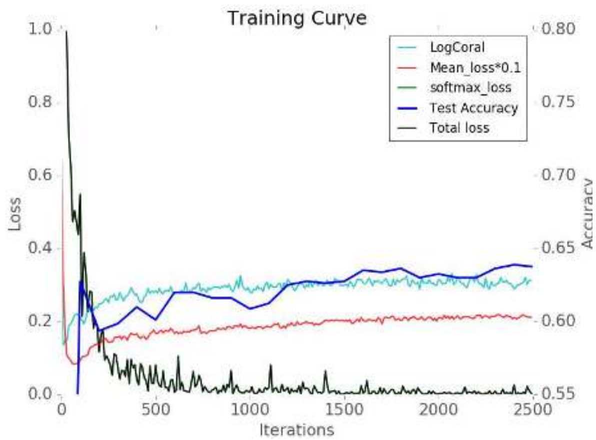
Figure 4a shows the learning curve of CNN, and also calculate the mean and LogCORAL loss even they are not used in learning procedure. Figure 4b shows the learning curve of mean model (*i.e.* optimizing mean loss and classification loss), and we also calculate the LogCORAL loss. Figure 4c shows the other way around which optimizes LogCORAL loss and classification loss, and we also calculate the mean loss.

From Figure 4a we observe that, without optimizing any of those distances, mean loss and LogCORAL loss would both go up. When optimizing mean loss, LogCORAL loss remain stable while mean loss has a obvious drop down, see Figure 4b. However if we optimize LogCORAL loss in Figure 4c, LogCORAL loss goes down but mean loss goes up. Those results indicate that those two losses have very weak correlation. This also explains why minimizing the two distances at the same time can achieve even better accuracy for domain adaptation.

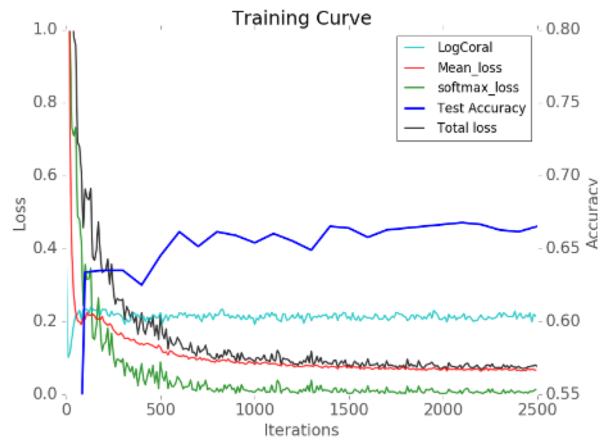
We further show the learning curve in Figure 4d, where we optimize both losses. In this case, both the mean and LogCORAL losses go down, and we achieve better result.

4.4. Discussion with other state-of-the-arts

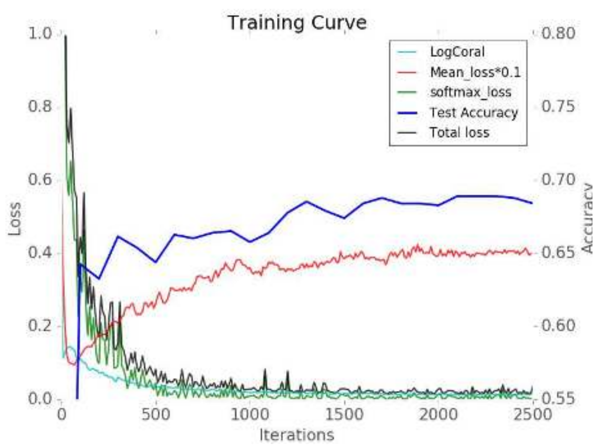
Deep domain adaptation is a fast growing research area. Many state of the art methods have been proposed on this topic. Generally speaking those methods can be classified into two main categories. First category is discrepancy based model, for example DAN [14], which directly minimizes the domain discrepancy (*eg*, MMD) to bring the source and target domains closer. The second category is



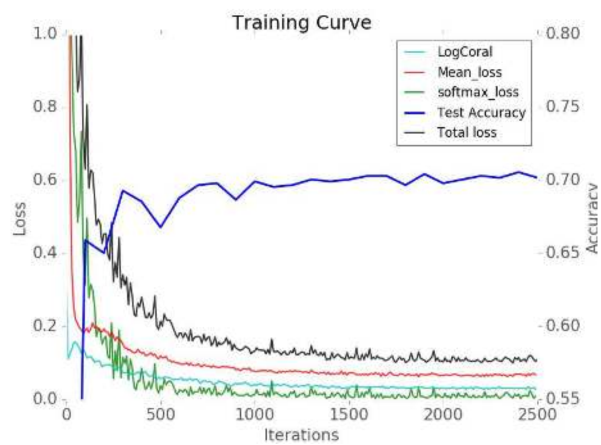
(a)



(b)



(c)



(d)

Figure 4: learning curves of different models on domain shift A-W. (a):Learning curve of CNN (*i.e.* only optimize classification loss). (b):Learning curve of mean model (*i.e.* optimize mean loss and classification loss). (c):Learning curve of Deep LogCORAL (*i.e.* optimize LogCORAL loss and classification loss). (d):Learning curve of combination model (*i.e.* optimize mean loss, LogCORAL loss and classification loss).

the adversarial model, for example GRL [7], which aims to confuse a domain classifier to learn transferable features.

Our method falls into the first category. We extend the CORAL method by using second order Log-Euclidean distance and combining with first order mean loss. We show that a proper distance is important for employing second-order statistical information to minimize the domain discrepancy. Some recent works give even higher accuracy on the Office dataset by using both the discrepancy and adversarial principles (for example, the JAN+A method [15]), and we believe it would be interesting to incorporate the proposed Log-Euclidean distance into those works to further boost the performance. We leave this to the future work for further study.

5. Conclusion and Future Work

In this paper, we proposed a new Deep LogCORAL method to minimize the geodesic distance on Riemannian manifold. We used the Log-Euclidean distance to replace the Euclidean distance in the Deep CORAL method, and also proposed a mean distance to additionally exploit the first-order satistical information for domaina adaptation. Our experimental results showed that our new Deep LogCORAL method generally outperformed the deep LogCORAL method for unsupervised domain adaptation using the benchmark Office dataset. In the future, we would like to incorporate the proposed LogCORAL loss into more models to futher improves the existing state-of-the-art methods.

References

[1] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM journal on matrix analysis and applications*, 29(1):328–347, 2007.

[2] S. Chopra, S. Balakrishnan, and R. Gopalan. Dlid: Deep learning for domain adaptation by interpolating between domains. In *ICML workshop on challenges in representation learning*, volume 2, 2013.

[3] Z. Cui, W. Li, D. Xu, S. Shan, X. Chen, and X. Li. Flowing on riemannian manifold: Domain adaptation by shifting covariance. *IEEE transactions on cybernetics*, 44(12):2264–2273, 2014.

[4] L. Duan, I. W. Tsang, and D. Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479, 2012.

[5] L. Duan, D. Xu, I. W.-H. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1667–1680, 2012.

[6] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2960–2967, 2013.

[7] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

[8] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.

[9] B. Gong, F. Sha, and K. Grauman. Overcoming dataset bias: An unsupervised domain adaptation approach. In *NIPS Workshop on Large Scale Visual Recognition and Retrieval*, volume 3. Citeseer, 2012.

[10] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012.

[11] Z. Huang and L. Van Gool. A riemannian network for spd matrix learning. *arXiv preprint arXiv:1608.04233*, 2016.

[12] W. Li, L. Duan, D. Xu, and I. W. Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1134–1148, 2014.

[13] W. Li, Z. Xu, D. Xu, D. Dai, and L. Van Gool. Domain generalization and adaptation using low rank exemplar svms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[14] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015.

[15] M. Long, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. *arXiv preprint arXiv:1605.06636*, 2016.

[16] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[17] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015.

[18] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. *Computer Vision–ECCV 2010*, pages 213–226, 2010.

[19] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. *arXiv preprint arXiv:1511.05547*, 2015.

[20] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops*, pages 443–450. Springer, 2016.

[21] T. Tommasi, T. Tuytelaars, and B. Caputo. A testbed for cross-dataset analysis. *arXiv preprint arXiv:1402.5923*, 2014.

[22] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE, 2011.

[23] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464*, 2017.

[24] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.