Manifold Guided Label Transfer for Deep Domain Adaptation

Breton Minnehan Andreas Savakis
Rochester Institute of Technology, Rochester, New York 14623, USA
{blm2144, andreas.savakis}@rit.edu

Abstract

We propose a novel domain adaptation method for deep learning that combines adaptive batch normalization to produce a common feature-space between domains and label transfer with subspace alignment on deep features. The first step of our method automatically conditions the features from the source/target domain to have similar statistical distributions by normalizing the activations in each layer of our network using adaptive batch normalization. We then examine the clustering properties of the normalized features on a manifold to determine if the target features are well suited for the second of our algorithm, label-transfer. The second step of our method performs subspace alignment and k-means clustering on the feature manifold to transfer labels from the closest source cluster to each target cluster. The proposed manifold guided label transfer methods produce state of the art results for deep adaptation on several standard digit recognition datasets.

1. Introduction

The aim of Domain Adaptation (DA) is to adapt a pre-trained classifier from a training dataset (source) to a test dataset (target) without performance degradation due to domain shifts between the datasets. The problem known as dataset bias [1] indicates that a model trained on a particular dataset is inherently biased to the properties of that dataset. This causes poor generalization across domains due to variations in pose, illumination, sensor properties, background and changes in the environment. Figure 1 shows sample images from different domains that represent the same object categories, but contain visually dissimilar images due to domain shifts. In these example datasets the domain shifts are a result differences in both data acquisition, handwritten vs street numbers, and imaging modality, grayscale vs. RGB imagery.

In this paper, we address the problem of visual domain adaptation for deep networks utilizing transfer learning, where the training and testing datasets have the same object categories but the domain-shift is unknown [2]. In this work we consider the case of unsupervised domain adaptation, where no labeled samples from the target domain are available. Several approaches have been proposed to tackle visual domain adaptation, as outlined in recent surveys and their references [3],[4].

Figure 1. Sample images from different domains showing variations in the same category across domains.

The approaches to domain adaptation can be grouped into two primary categories, supervised and unsupervised. Supervised methods adapt to a new domain by using a small set of labeled data in the target domain. During unsupervised domain transfer, the classifier must adapt to a new dataset without any knowledge of class labels. Thus, it is often difficult to confirm the classifier is constructively adapting to the target domain. This is why we consider a classification metric to make the target dataset better suited for adaptation using label transfer.

One of the main challenges with domain adaptation is how to select features that are suitable for both source and target domains. While traditional features, such as SIFT [5] and HOG [6], have been used in multiple domain adaptation works, recent focus has been on adapting the features extracted using deep learning. The approach of pre-training and fine-tuning [7] a deep neural network has been widely adopted as a solution to the problem of supervised domain adaptation. The problem often faced with fine-tuning is overfitting the small adaptation set. Thus, most work in the field of supervised domain adaptation has focused on developing an adaptation methodology that allows the network to adapt to the new data without overfitting. Various strategies have been considered to minimize overfitting including: training only a subset of the network layers, reducing the learning rate, and introducing dropout to increase regularization. The conclusion is that there is no single optimal approach for fine-tuning a network. Each adaptation problem is unique and thus the optimal approach

depends on the specific conditions.

Unsupervised domain transfer presents a unique challenge, as there are no labels that can be used to retrain the classifier. Many unsupervised adaptation techniques focus on adapting the feature extractor instead of the classifier itself. The goal of adapting the feature extractor is to make the features extracted from the target domain as similar as possible to those from the source [8]-[12]. The justification for this approach is that if the feature set of the target domain can be made indistinguishable from that of the source, the same classifier can be used for both.

The main contributions of this paper are the following. (a) We utilize adaptive batch normalization inspired by [13] to generate similar feature distributions across domains and make the alignment process more effective. (b) We leverage the similarities in the source and target features using a manifold inspired approach, based on subspace alignment, to guide the label transfer process during unsupervised deep domain adaptation. We propose a label transfer method to adapt the classifier for target domains that exhibit proper feature clustering behavior on their subspace-aligned PCA manifold. (c) We demonstrate that our approach achieves state-of-the-art results on standard digits datasets.

2. Related Work

2.1. Domain Adaptation

There are many supervised domain adaptation approaches that are well studied include transformative learning [14] and metric learning [2]. However, supervised domain adaptation has received less interest in recent years as applications are increasingly requiring domain adaptation without any labeled data. Thus, the focus of many recent works in the field of domain adaptation has been on unsupervised domain adaptation strategies. These methods often consider dimensionality reduction, such as principal component analysis (PCA) for domain representations [15]-[19]. For example, domain adaptation methods based on dimensionality reduction are proposed in [18],[19]. These approaches try to discover a latent space that minimizes the mismatch in the distributions between the two domains.

Methods based on manifold alignment look for a projection that preserves the local neighborhood information [20], [21]. Adaptation in [22] is performed by aligning the basis vectors of the source domain to the target domain by learning a transformation that minimizes the Bregman divergence. Domain adaptation methods based on metric learning and canonical correlation analysis (CCA) are outlined in [2], [23], [24].

Grassmannian based domain adaptation explores intermediate feature representations on the manifold [15]-[17]. In [15], intermediate subspaces are sampled from the manifold geodesic curve and are combined to obtain a domain invariant space. The methods proposed in [16] and [17] integrate the subspaces on the geodesic between the source and target domain to learn a transformation matrix.

There are many approaches proposed for shallow unsupervised domain adaptation using a variety of methods such as: manifold learning [15], [25], aligning principal components [22], and leaning explicit mappings between the domains [26]. However, each of these methods are computationally complex and do not adapt well to the high-dimensional features common with most deep learning networks. Additionally, because these methods are based on shallow features, they do not account for the hieratical manner in which deep features are formed, and instead focus entirely on building a mapping between the source and target feature domains leaving the feature extraction process unaltered.

In order to leverage the advances in deep learning feature representations for domain adaptation many methods have focused on adapting the feature extraction network. In one of the first methods designed for domain adaptation of deep features [8], the authors propose using second order statistics to “whiten” and “recolor” the target features to match the source features. This was done by minimizing what they called the CORAL loss. This method then extended in [9] to retrain the entire target feature extraction network based on minimizing the CORAL loss. Similarly, other works have used different loss functions to adapt the feature extraction network in Deep Domain Confusion [10], Deep Adaptation Networks [11], and Deep Transfer Networks [12].

Adversarial Learning is a growing area of deep learning that has recently received a lot of attention [27]. Adversarial methods focus on iteratively training two networks with opposing objectives to learn optimal feature representations. These networks were originally developed to randomly generate synthetic imagery that was “believable” or looking similar to actual imagery. Recently these networks have been used for domain adaptation, first in [28], [29] and later in [30]. There are major differences between these two methods, the primary difference being [28], [29] use a single symmetric feature extraction network, whereas [30] uses an asymmetric feature extraction configuration with two separate feature extraction networks for source and target data.

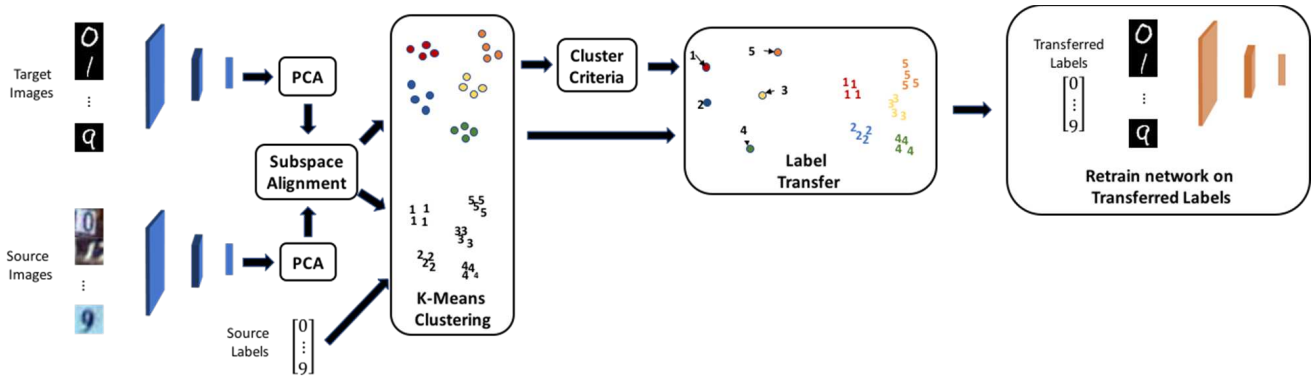

Figure 2. Overview of network training and unsupervised domain adaptation to the target domain. Adaptive batch normalization is used for training and adaptation in both source and target domains. Subspace alignment is performed for source and target features on the PCA manifold and the features are clustered to determine if label transfer is appropriate based on a clustering criterion. Label transfer is performed by assigning labels from the closest source cluster to each target cluster and using them to retrain the network.

2.2. Domain Adaptation with Self-Learning

Another group of domain adaptation techniques assume the source and target domains are close enough that the classifier itself could be used to retrain on the target domain. These domain adaptation techniques are known as self-learning [31]. These methods use "pseudo-labels" or "weak-labels" [32], labels produced by a pre-trained classifier, to bootstrap the network and adapt based on its the classifier's predictions. Asymmetric Tri-training (ATT) [33], the current state of the art method for domain adaptation, relies on a similar method for generating labels for the data except they train three independent networks to generate the labels in order to increase the accuracy of the "pseudo-labels." Both techniques use inductive-learning to adapt to the source domain. Unlike adversarial training, these methods adapt the feature representation and classifier jointly to minimize classification error. Adversarial methods focus on maximizing domain confusion.

A different subcategory of self-learning techniques uses transductive-labeling to adapt the network to the new domain. Transductive-labeling transfers the label from a sample in the source domain to a sample in the target domain to retrain the network. A common technique for transductive-labeling is using the dominant label of the closest K samples from the source domain, in feature space, to each sample from the target domain [25]. These techniques work well when the source and target domains are relatively similar, but fail when there is a stark difference between the source and target domain.

3. Methodology

We propose a new approach for unsupervised deep domain adaptation outlined in Figure 2. Our method first uses Adaptive Batch Normalization (ABN) [13] to produce features that are similar between source and target domains. The distributions of these deep features are compared using Silhouette score [34], a clustering metric commonly used in data-mining, to determine if the target features form clusters that are well suited for label transfer. If the target features are well suited for further feature adaption, we use a label transfer method to generate training labels for the target samples. This is the first method that combines the automatic feature-adaptation of ABN with the subspace aligned label transfer for domain adaptation. We overview each step of our method in the rest of this section.

3.1. Adaptive Batch Normalization

The first and often most important step in domain adaptation is to adapt the feature extraction method for the source and target domains in a way that the two sets of features are in the same subspace. The fact that two sets of features must share the same subspace has proven to be so important that most recent works focus entirely on feature adaptation without retraining the classifier [28], [30].

In this work we combine adaptive batch normalization [13] and subspace alignment [22] to perform an initial alignment of the features from the two domains. Then based on the clustering of the target domain features we further adapt the feature extraction network and classifier.

We selected ABN as the first step in our feature alignment process for its ease of use and impressive results without any additional training. The method of batch normalization [35] was originally developed to increase the robustness of training deep neural networks by normalizing inputs to each neuron, x_i, in the network across all the samples in the current mini-batch. The whitened input, $\hat{x}_i$, of each input to neurons is calculated by the equations:

$$\bar{x}_i = \frac{1}{T} \sum_{t \in (0,T)} x_i \qquad (1)$$

$$\sigma_i^2 = \frac{1}{T}\sum_{t\in(0,T)}(x_i - \bar{x}_i)^2 \quad (2)$$

$$\hat{x}_i = \frac{x_i - \bar{x}_i}{\sqrt{\sigma_i^2}} \quad (3)$$

where T denotes the number of samples in the mini-batch, and i is the dimension index of the input (if the input is multi-dimensional the index will be a vector).

Normalizing the activations across each mini-batch ensures the gradients for each update set are better suited for training, especially in early epochs with randomly initialized weights. Batch normalization has been shown to produce more stable networks that can be trained faster with higher learning rates [35].

In their original implementation of batch normalization the authors suggested that the population statistics and whitening parameters should be learned using a moving average during the training phase and stay frozen at test time. Thus, the activations of all the test samples would be whitened based on the statistics of the training set, not the test set. However, it was suggested by [13] that by allowing the batch normalization layer to continue to adapt to the changing population statistics at test time one can achieve a domain adaptation without any additional training. This method of feature adaptation is known as adaptive batch normalization.

By normalizing the activations for each of the features in each mini-batch, the features are inherently adapting such that the features for each sample from the source and target domains have the same normalized distribution. In our experiments, we found adaptive batch normalization to be more effective than traditional feature transformation methods for aligning the source and domain feature spaces. Adaptive batch normalization outperforms many of the simpler feature transformation methods by normalizing, and indirectly aligning, the features for the two domains for each layer, as opposed to methods such as [22] which only learn a transformation for the final feature representation. Additionally, we found the feature alignment achieved with adaptive batch normalization alone was on par with the results from adversarial feature adaptation techniques.

3.2. Inductive-Clustering

Most approaches to unsupervised domain adaptation focus only on aligning the distributions of the high-dimensional feature representations of the source and target examples. It is assumed that if the feature distributions are ideally aligned, the classifier should perform equally well on the target and source examples. Unfortunately, this is not always the case. There are, potentially, many transformation that can align the features of the target and source domains. The work in [36] has shown that the optimal feature transformation for source and target features belongs to the set of transformations that perfectly align the two distributions. However, there is no guarantee that any transformation that aligns the two distributions is the optimal transformation. In our experiments, we have found that the feature alignment for visually distinct domains is just as likely to be destructive as it is to be beneficial to domain adaptation. The problem is that feature alignment is not enough for optimal domain adaptation when the domains are significantly separated. By adapting only the features and leaving the classifier unchanged, there is a significant amount of the adaptation potential in the system that is left untapped. The challenge, however, is that there are no labels for the target data, thus traditional methods for retraining deep networks, such as [7], cannot be used.

In order to get a better understanding of the clustering behavior of the source and target features, we plotted the feature representations using t-SNE plots [37]. The t-SNE algorithm is a dimensionality reduction method that is designed to faithfully represent the distribution of high-dimensional features in a much lower dimensional space. Examples of t-SNE plots for two datasets are shown in Figure 3. Guided by this analysis of the source and target features we discovered that the target features often form clusters that are relatively close to the source clusters.

We therefore propose to use the manifold learning technique known as label transfer to generate labels for the unlabeled target samples based on the pairing of the clusters from the source and target domains. However, this label transfer procedure only works when the target and source clusters are relatively close. Otherwise, this procedure can potentially reduce the accuracy of the network because of training on poor labels.

To ensure that the target features are well suited for our label transfer training, we first look at the distribution of the target feature clusters in PCA space. The metric we use to determine the suitability of the target features was Silhouette score [34], or the mean ratio of the inter-class variance to intra-class variance:

$$V_c = \sum_{i\in(0,\boldsymbol{C})}\sum_{j\in[i,\boldsymbol{C})}\left\|\bar{x}_i - \bar{x}_j\right\| \quad (4)$$

$$R_v = \frac{1}{\boldsymbol{C}} * \sum_{i\in(0,\boldsymbol{C})}\frac{\sum_{x_j\in C_i}\left\|x_j - \bar{x}_i\right\|}{V_c} \quad (5)$$

where $\boldsymbol{C}$ is the number of classes. x_j is the target feature vector in PCA space, $\bar{x}_i$is the centroid for the given cluster i, and C_i is the subset of features that belong the cluster i. We experimentally found that using the cutoff criteria of $R_v < 0.625$ was ideal for selecting well clustered target domains to perform the second step of our domain adaptation procedure updating the classifier using the transferred cluster labels.

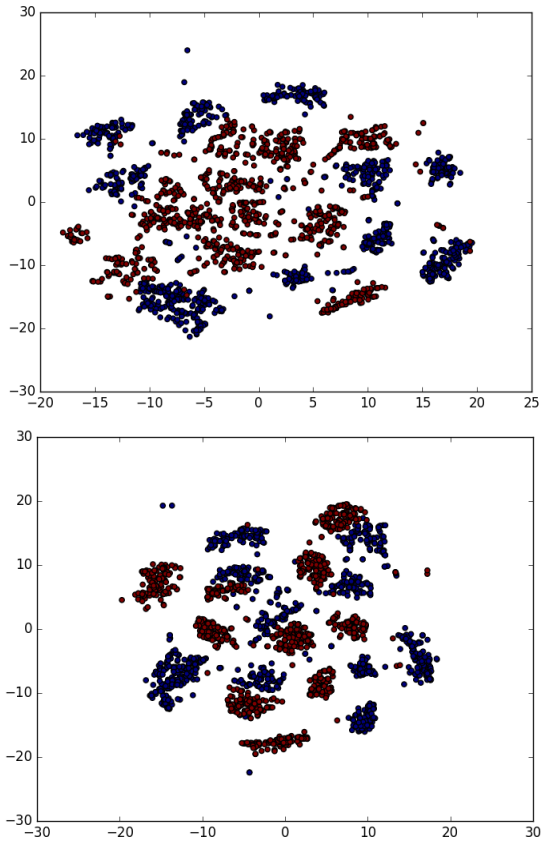

Figure 3: Top: t-SNE Plot of features extracted using a network trained on the USPS dataset. Bottom: t-SNE Plot features extracted using a network trained on the USPS dataset and adapted to the MNIST dataset. USPS (Blue) MNIST (Red).

The target features that satisfy our Silhouette score criteria generally form clusters that are close to the source clusters after subspace alignment, but they do not match perfectly. This suggests that further improvement of the classification accuracy can be made by retraining the classifier on the target features.

3.3. Label Transfer via Manifold Clustering

Previous methods for retraining the classifier have turned to two sources to generate the target sample labels: weak-labels (inductive-learning) [33], or label transfer (transductive-learning) [25]. In our work, we selected a label transfer approach based on the clustering behavior of the features from the target domain. During this step we found additional gains in accuracy by first aligning the PCA subspaces of the source and target domains using the technique proposed in [22]. Although this step was not required, we found it slightly increased the final network's accuracy. We also found this step reduced computation time for the clustering procedure by reducing the dimensionality from 1024 to 15 principal components.

The subspace alignment process consists of performing PCA analysis on the source and target feature sets and combining the two transformations to recolor the features from the source domain to match the target domain. Because our method iteratively updates the target features during its retraining procedure, a subset of the target features must be resampled periodically to update the PCA transformation for the target features based on the changes to the classification network.

The iterative update of the target features in our method also adds a large overhead to the subspace alignment procedure as proposed in [22]. The authors suggest that the optimal alignment for domain adaptation is produced by recoloring the source features to match the target features. However, because our target features are constantly changing, recoloring the source features to match the target features introduces a great deal of computational overhead without a significant improvement to the classification accuracy.

Thus, in this work we invert the subspace alignment procedure to recolor the target features, to match the source features, as follows:

$$S_s = F_s X_s \tag{6}$$

$$S_t = F_t X_t X_t' X_s \tag{7}$$

where S_s and S_t are the resulting features for the source and target domains, respectively, the in the aligned subspace, and X_s and X_t are the PCA transformations for the source and target features, respectively.

This change significantly reduces the processing time for our domain adaptation process, because it only requires K-means clustering to be run once on the source features (K=10). The method used in the original subspace alignment procedure constantly updates the K-means clustering of the source features and target features and is much slower.

After the subspace alignment transformation is recalculated, K-Means clustering (K=10) is rerun only on the updated aligned target features. The centroids of these target clusters are then greedily matched to the closest cluster centroid of the source clusters, maintaining a strict one-to-one correspondence. The dominant label in each source cluster is assigned to all the samples in the corresponding target cluster. These transferred labels are subsequently used for training the network. This process is iterated until convergence

The plots in Figure 3 depict the impact of label transfer learning on the feature representation for each domain. Features were extracted from test samples from the USPS [38] and MNIST [39] datasets using a network trained on the USPS dataset. Self-learning on the clustering of the data results in tighter clusters for the MNIST data leading to a 25.7% reduction in classification error.

4. Experiments

In order to examine the performance of the proposed unsupervised domain adaptation method, we tested it on standard domain adaptation problems of digit classification across multiple datasets [38]–[40]. The MNIST [39] dataset is one of the first digit recognition datasets used in deep learning. It contains 60,000 training sample and 10,000 test samples of 18x18 black and white examples of handwritten digits 0-9. The USPS [38] dataset is quite similar to the MNIST dataset, however, the number of samples is much smaller and the images are in gray scale.

The works in [28], [29], presented a more challenging digits dataset that was generated by combining the MNIST images with randomly cropped patches from the BSDS500 dataset [41]. Unfortunately, the exact dataset used in [28], [29] is not released publicly. However, we followed an identical procedure to that proposed in their work to generate a similar dataset that we refer to as MNIST_M.

The most recent digits dataset is the Street View House Numbers (SVHN) [40] dataset in which a 600,000 labeled examples are extracted from RGB images, many include additional digits in the bounding box for each individual digit. The SVHN dataset is the most diverse and challenging of the digits datasets. It is common for domain adaptation methods to work well when adapting from SVHN to a different domain, however, it is not common for methods to work transferring to the SVHN domain because of its complexity. The SVHN dataset is a practically hard dataset because the image patches often include multiple digits, as illustrated in Figure 1.

For our experiments we implemented a simplified version of the network proposed in [39], shown in Figure 4, with two convolutional layers using 32 and 64 5x5 filters each, followed by a fully connected layer with 1024 hidden nodes whose output was connected to a softmax function. All datasets were converted to grayscale and scaled to 28x28 pixels, so that the same network could be used across all the datasets. Batch normalization was implemented in the network by normalizing the features prior to the activation function being applied. For this network we used a ReLU nonlinearity.

One key difference between our network and the original architecture proposed by [39], besides the inclusion of Batch Normalization, is that all max pooling layers were replaced with skip convolutions to reduce the number of computations without much loss in accuracy.

The networks were trained on the source datasets for 20 epochs using the Stochastic Gradient Decent with a momentum of 0.9, a learning rate of 0.01, 50% dropout rate and a mini-batch size of 64. The results in Table 1 demonstrate that this slightly altered network performed well for the many of the domain adaptation tasks without any retraining.

The network was adapted using label transfer for 5 epochs using the same learning rates and optimizer as was done for the original training of the network. In order to adapt the subspace alignment to changes in target feature representation, the target PCA transformation was updated using a subset of target features after every 20th mini-batch update. The new set of target features were generated from a random subset of 100 target mini-batches. These updated features were used to calculate an updated PCA transformation for the target features and the corresponding subspace alignment transformation.

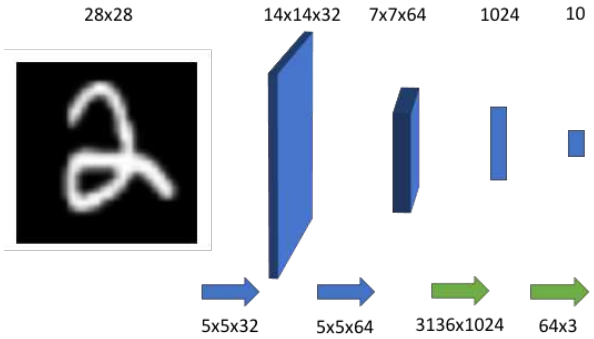

Figure 4: Digit classification network.

Table 1. Domain Adaptation for Digits Datasets

Method	M->U	S->M	M_M->M	U->M	M->S
Source only	0.856	0.822	0.910	0.624	0.071
Source w/ABN	0.823	0.830	**0.923**	0.721	0.253
Source w/ABN and Label Transfer	**0.915**	**0.977**	**0.923**	0.721	0.253
DANN [10]	0.771	0.739	81.5	**0.730**	**0.357**
ADDA [11]	**0.894**	0.760	N.A.	**0.901**	N.A.
ATT [14]	N.A.	**0.85**	N.A.	N.A.	**0.528**
UTDA [23]	N.A.	0.774	0.835	N.A.	0.323

After the updated target features were realigned to the original features from the source domain, K-Means clustering was rerun on the subspace aligned target features and the clusters were paired to the source domain clusters. The K-means clustering trained on the random target samples was then used to determine the target cluster for each sample in the training mini-batches, which was used to transfer the labels from the source clusters to the target samples.

We compare the results produced by our method with the results reported by the state-of-the-art methods for digit domain adaptation on the digits datasets in Table 1. The results with the highest accuracy are in Red and the second

highest shown in Bold. These results illustrate that the proposed method improves significantly on the state of the art results.

The first observation from the results of Table 1 is the increase in accuracy for most domain adaptation tasks when adaptive batch normalization is used. However, some adaptation tasks do not benefit from ABN. This is likely due to the lower visual disparity between the two domains, thus the adaptation of the features to fit the target domain is not as useful. The impact of ABN on the more visually distinct domains, such as the MNIST and SVHN datasets, is much more apparent. ABN more than triples the accuracy of the classifier on the MNIST to SVHN adaptation task, without any retraining of the network. These results are consistent with the increase in performance achieved by ABN in [13], even though their experiments were on a different domain transfer task.

Another key observation is that self-learning appears to only work with ABN. We attempted to train the network with self-learning alone, freezing the batch normalization statistics, and we found that the network performance slowly deteriorated over the course of training.

It is important to point out that our method works best in situations where the original network with ABN is relatively accurate in the target domain. We found that the primary factor in the ability of our method to adapt to new domains was the visual diversity in the source dataset. As is often the case when training deep networks, when the training set has a limited amount of visual diversity the network learns a set of sub-optimal features. Deep networks require a lot of visual variation in the training set in order learn features that generalize well to other domains. Our experiments demonstrated just how large of a role dataset size and variation play in the network's ability to generalize to new domains.

In our experiments we obtained the worst results when the source domain was USPS, the smallest dataset with the least variation. In the case of the USPS to MNIST adaptation task, the target features did not even pass the clustering criteria. Interestingly our method performed quite well on the inverse adaptation task, MNIST to USPS. This disparity between directions of adaptation is likely because the MNIST dataset is larger and includes more variation in the data.

Although our method performed well on the MNSIT to USPS domain adaptation task, the network trained on the MNIST dataset did not adapt well to either the MNIST_M or the SVHN dataset. This is most likely because the MNIST_M and SVHN datasets are visually distinct from the MNIST dataset, and thus it is harder for a network trained on simpler domains to adapt to more difficult domains.

It is important to point out that for the inverse adaptation task, SVHN to MNIST, our method achieves far superior results. This demonstrates that visual disparity between domains is not important so long as the source domain is more visually diverse than the target domain. Our method performs optimally when the source domain includes more visual variation than the target domain, such as the SVHN and MNIST_M dataset. This is because the features trained from more diverse datasets tend to be more generalizable to new domains.

These results suggest that the proposed method will work well for any domain adaptation task where two requirements are satisfied: (a) the source contains sufficient visual variation for the network to learn high quality features. (b) the target domain is close enough to the source domain so that ABN combined with subspace alignment can form ideal clusters for our label transfer procedure.

5. Conclusion

In this paper, we presented a novel two step procedure for adapting a deep neural network in an unsupervised manner to a new, unlabeled, domain. Our method first uses adaptive batch normalization to make sure that the source and target features exist in similar subspaces. Then transductive-label transfer is used to better align the clusters in the source and target domains. The results produced by the proposed approach outperforms other state of the art methods for many of the domain adaptation datasets.

References

[1] A. Torralba and A. A. Efros, "Unbiased Look at Dataset Bias," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[2] T. D. Brian Kulis, Kate Saenko, "What You Saw is Not What You Get: Domain Adaptation Using Asymmetric," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1785–1792.

[3] V. Patel and R. Gopalan, "Visual Domain Adaptation: A survey of recent advances," *IEEE Signal Processing Magazine*, vol. 32.3, pp. 53–69, 2015.

[4] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[5] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the IEEE International Conference on Computer Vision*, 1999, vol. 2, no. 8, pp. 1150–1157.

[6] N. Dalal and W. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, vol. 1, no. 3, pp. 886–893.

[7] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and

description," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[8] B. Sun, J. Feng, and K. Saenko, "Return of Frustratingly Easy Domain Adaptation," in *AAAI Conference on Artificial Intelligence*, 2016, pp. 2058–2065.

[9] B. Sun and K. Saenko, "Deep CORAL: Correlation Alignment for Deep Domain Adaptation," *arXiv Preprint*, 2016.

[10] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep Domain Confusion: Maximizing for Domain Invariance.," *arXiv Preprint*, 2014.

[11] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning Transferable Features with Deep Adaptation Networks," *arXiv Preprint*.

[12] X. Zhang, S. Wang, F. X. Yu, and S.-F. Chang, "Deep Transfer Network: Unsupervised Domain Adaptation," *arXiv Preprint*, 2015.

[13] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting Batch Normalization For Practical Domain Adaptation," *arXiv Preprint*, 2016.

[14] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proceedings of European Conference on Computer Vision*, 2010, vol. 6314 LNCS, no. PART 4, pp. 213–226.

[15] R. Gopalan, R. Li, and R. Chellappa, "Domain Adaptation for Object Recognition : An Unsupervised Approach," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 999–1006.

[16] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2066–2073.

[17] J. Zheng and P. J. Phillips, "A Grassmann Manifold-based Domain Adaptation Approach," in *Proceedings of the IEEE International Conference on Pattern Recognition*, 2012, no. ICPR, pp. 2095–2099.

[18] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Unsupervised domain adaptation by domain invariant projection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 769–776.

[19] S. Pan, J. Kwok, and Q. Yang, "Transfer Learning via Dimensionality Reduction.," in *AAAI Conference on Artificial Intelligence*, 2008, pp. 677–682.

[20] C. Wang and S. Mahadevan, "Manifold alignment without correspondence," in *Proceedings of International Joint Conference on Artificial Intelligence*, 2009, pp. 1273–1278.

[21] C. Wang, "Heterogeneous Domain Adaptation Using Manifold Alignment," in *Proceedings of International Joint Conference on Artificial Intelligence*, 2010, pp. 1541–1546.

[22] 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*, 2013, pp. 2960–2967.

[23] M. Chen, K. Q. Weinberger, and J. C. Blitzer, "Co-Training for Domain Adaptation," in *Advances in Neural Information Processing Systems*, 2011, pp. 1–9.

[24] I. H. Jhuo, D. Liu, D. T. Lee, and S. F. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2168–2175.

[25] O. Sener, H. O. Song, A. Saxena, and S. Savarese, "Unsupervised Transductive Domain Adaptation," in *Advances in Neural Information Processing Systems*, 2016.

[26] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain Adaptation via Transfer Component Analysis Domain Adaptation via Transfer Component Analysis," *IEEE Trans. Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[27] I. J. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[28] Y. Ganin *et al.*, "Domain-Adversarial Training of Neural Networks," *JMLR*, vol. 17, pp. 1–35, 2016.

[29] Y. Ganin, G. Ru, V. Lempitsky, and L. Ru, "Unsupervised Domain Adaptation by Backpropagation," in *Proceedings of the International Conference on Machine Learning*, 2015, pp. 1180–1189.

[30] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," *arXiv Preprint*, 2016.

[31] D. Yarowsky, "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods," in *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, 1995, pp. 189–196.

[32] D. H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proceedings of the International Conference on Machine Learning Workshop on Challenges in Representation Learning*, 2013, no. July 2013.

[33] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric Tri-training for Unsupervised Domain Adaptation," *arXiv Preprint*, 2017.

[34] P. J. Rousseeuw, "Silhouettes: A graphical aid to

the interpretation and validation of cluster analysis," *Computational and Applied Mathematics*, vol. 20, no. C, pp. 53–65, 1987.

[35] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv Preprint*, 2015.

[36] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1–2, pp. 151–175, 2010.

[37] L. J. P. Van Der Maaten and G. E. Hinton, "Visualizing high-dimensional data using t-sne," *JMLR*, vol. 9, pp. 2579–2605, 2008.

[38] J. J. Hull, "A Database for Handwritten Text Recognition Research," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, 1994.

[39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[40] Y. Netzer and T. Wang, "Reading digits in natural images with unsupervised feature learning," in *Advances in Neural Information Processing Systems*, 2011, pp. 1–9.

[41] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.