Incremental Dictionary Learning for Unsupervised Domain Adaptation

Boyu Lu[1]
bylu@umiacs.umd.edu
Rama Chellappa[1]
rama@umiacs.umd.edu
Nasser M. Nasrabadi[2]
nasser.m.nasrabadi.civ@mail.mil

[1] Department of Electrical and Computer Engineering
University of Maryland
College Park, MD, USA

[2] U.S. Army Research Lab.
Adelphi, MD, USA

Abstract

Domain adaptation (DA) methods attempt to solve the domain mismatch problem between source and target data. In this paper, we propose an incremental dictionary learning method where some target data called supportive samples are selected to assist adaptation. Supportive samples are close to the source domain and have two properties: first, their predicted class labels are reliable and can be used for building more discriminative classification models; second, they act as a bridge to connect the two domains and reduce the domain mismatch. Theoretical analysis shows that both properties are important for adaptation, enabling the idea of adding supportive samples to the source domain. A stopping criterion is designed to guarantee that the domain mismatch decreases monotonically during adaptation. Experimental results on several widely used visual datasets show that the proposed approach performs better than many state-of-the-art methods.

1 Introduction

Classification tasks often assume that the training and testing data are drawn from the same distribution. However, this assumption is often challenged by real applications where testing data may have large intra-class variances compared to the training data. For example, face recognition models trained on frontal faces with good resolution may be called upon to classify non-frontal or blurred faces. This domain shift has resulted in a large drop in classification performance and many DA methods have been developed to address this problem [1, 3, 6, 7, 9]. There are two main settings for DA: semi-supervised DA allows a few class labels in the target domain and in the case of unsupervised DA, target labels are not available. In this paper, we mainly focus on the more difficult unsupervised setting.

One class of unsupervised methods discover a domain-adaptive subspace for both domains. They learn a transformation and project samples from both domains into a common subspace, in which the distribution divergence between the two domain becomes smaller [1, 6, 7, 9]. Others attempt to reduce the domain mismatch by reweighting or selecting some source samples [8, 13]. Although, some of them take advantage of the source sample labels, most of these approaches fail to fully use the discriminative information in the target domain.

Pages 108.1-108.12
DOI: https://dx.doi.org/10.5244/C.29.108

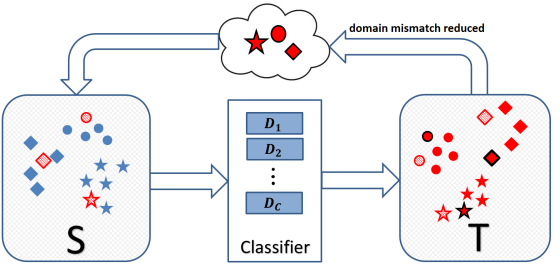


Figure 1: Scheme of the incremental dictionary learning for domain adaptation. The original source data is colored in *blue* and the target data is colored in *red*. Different shapes represent different classes. The red samples with shadow indicate the previously selected supportive samples that have been added to the source domain. The red samples with black border represent the supportive samples selected in the current iteration.

Since the distributions in both domains are different, exploiting only the source labels does not necessarily preserve the discriminative information in the target domain.

In contrast, some bootstrapping-based DA methods [8, 12, 19, 21] use the source classifier to predict some target labels and then add them to the source domain to adapt the initial classifier. However, a major limitation of these methods is that they need some heuristic threshold to determine when to stop the adaptation and they do not explicitly show whether the adaptation can reduce the domain mismatch in each iteration [8, 21]. As shown in [2, 16], the divergence between the two domains is the key factor that affects the classification performance. So it is desirable to guarantee that the dissimilarity decreases monotonically.

Dictionary learning has shown good performance in classification [22, 23] and domain adaptation [17]. In this paper, we propose an incremental dictionary learning-based method which explicitly reduces the domain divergence, and simultaneously performs the adaptation and classification. Specifically, we iteratively find some *supportive samples* in the target domain and add them to the source domain. These supportive samples have two nice properties. First, the predicted labels of the supportive samples are reliable. So they are used to train a more powerful classification model. Second, the supportive samples are close to the source domain. So they reduce the domain divergence. As stated in [2], both properties are important for domain adaptation.

In addition, a good stopping criterion is crucial for efficient adaptation. We introduce a domain similarity measure and only conduct adaptation when the domain similarity value increases after each iteration. In this way, we automatically guarantee that our adaptation will monotonically reduce the domain mismatch. Both theoretical analysis and experimental results show the effectiveness of the stopping criterion.

The rest of the paper is organized as follows. A brief review of related works is presented in section 2. In section 3, we describe the proposed incremental DA approach and provide a theoretical analysis of the algorithm. Experimental results on object recognition and face recognition datasets are reported in section 4, with some more analysis on supportive sample selection and stopping criterion of adaptation. We conclude in section 5.

2 Related Work

Recently, domain adaptation has drawn great attention in vision community. One class of DA approaches closely related to our method is bootstrapping-based methods [3, 12, 19, 21]. Basically, they first train the source classifier which is used to predict the labels of target data. Then some of the highly confident labeled target samples are added to the source domain to retrain the classifier. This procedure is repeated iteratively until convergence. In [3, 12, 21], some samples close to the classifier margins are chosen as candidates to be added to the source. However, the predicted labels are not reliable enough and may mislead the classifier. To overcome this problem, [19] added a regularizer to the classifier. But they need to tune many heuristic thresholds which makes it difficult to generalize. In addition, for all these methods the domain mismatch may not decrease consistently.

Dictionary learning has also been widely used for domain adaptation. [17] built a smooth path of intermediate dictionaries from the source to the target domain by minimizing the reconstruction error and explicitly reducing the domain mismatch. However, labels are not used during adaptation. In [20], authors learned two domain-specific projections and a common dictionary to embed data into a low-dimensional space. They also considered the source labels to improve the discriminative ability of the dictionaries. There is no guarantee, however, that the low-dimensional space can reduce domain dissimilarity. In addition, none of these approaches exploit the target discriminative information.

3 Proposed Approach

In this section, we first present the proposed incremental dictionary learning-based DA method. We will then introduce a domain similarity measure and give some theoretical analysis to prove the effectiveness of the proposed method. We begin with describing some notations used in the paper.

We use $X^s = X^{(0)} = \{x_i^s\} \in \mathbb{R}^{d\times N_s}$, $X^t = \{x_i^t\} \in \mathbb{R}^{d\times N_t}$ to denote the data from source and target domains where N_s , N_t denote the number of samples respectively, and d is the dimension of data. Let $L = \{1,..,C\}$ represent the existing label set. Let $D^{(0)} = [D_1^{(0)}|...|D_C^{(0)}]$ denote the original dictionary trained on source domain where $D_j^{(0)} \in \mathbb{R}^{d\times K}$ denote the sub-dictionary that corresponds to class j, and K represents the number of atoms in each class-specific sub-dictionary. Let $P \in \mathbb{R}^{N_t\times C}$ denote the confidence matrix with each element $p_{ij} \in [0,1]$ representing the probability that target sample x_i^t belongs to the class j. Let $W \in \mathbb{R}^{N_t\times C}$ denote the binary selection matrix with each element $w_{ij} \in \{0,1\}$ indicating whether the target sample x_i^t is selected as supportive samples for class j. $X^{(k)}$, $D^{(k)}$, $P^{(k)}$, $W^{(k)}$ denote the augmented source domain, dictionary, confidence and selecting matrix in the k^{th} iteration.

3.1 Incremental Dictionary Learning for DA

Given the dictionary $D^{(k)}$, we want to select a subset of target samples as supportive samples. We have two constraints on this selection. First, the supportive samples selected in the previous iterations should be excluded as we want to add new data for adaptation. Second, we select equal number of supportive samples for each class to ensure class balance during adaptation [8]. With these two constraints, we select the most confident samples

that minimize the reconstruction error when represented by $D^{(k)}$. Then we update the augmented source domain by adding supportive samples and retrain the dictionary. After that, the stopping criterion is checked to see whether adding new supportive samples will reduce the domain dissimilarity. The proposed approach is shown in Fig. 1.

Confidence Matrix Update: In the $(k+1)^{th}$ iteration, we update the confidence matrix $P^{(k+1)}$ using the current class-specific dictionaries $D^{(k)} = [D_1^{(k)}|...|D_C^{(k)}]$:

$$p_{ij}^{(k+1)} = \begin{cases} \frac{\frac{1}{\sqrt{2\sigma^2}}\exp(-\frac{e_{ij}^{(k+1)}}{2\sigma^2})}{\sum_{l=1}^{C}\frac{1}{\sqrt{2\sigma^2}}\exp(-\frac{e_{il}^{(k+1)}}{2\sigma^2})} & \text{if } j = \underset{l}{\operatorname{argmax}}\, p_{il}^{(k+1)} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where σ^2 is a normalization parameter and e_{ij} denotes the reconstruction error of target sample x_i^t using $D_j^{(k)}$:

$$e_{ij}^{(k+1)} = ||x_i^t - D_j^{(k)} \cdot Z_{ij}^{(k+1)}||_2^2 \tag{2}$$

where $Z_{ij}^{(k+1)}$ is the sparse code. Here $p_{ij}^{(k+1)} \neq 0$ only when j is the most likely class that sample i belongs to. This constraint guarantees that a sample cannot be selected as the supportive sample for multiple classes.

Supportive Samples Selection: We select new supportive samples using $W^{(k+1)}$ by solving the following optimization problem:

$$\begin{aligned} W_j^{(k+1)} &= \underset{W_j}{\operatorname{argmax}} \quad tr(W_j P_j^{(k+1)}) \\ s.t. \quad W_j^{(k+1)} &\cdot \sum_{l=1}^{k} W_j^{(l)} = 0, \quad ||W_j^{(k+1)}||_0 = Q, \quad j = 1,...,C \end{aligned} \tag{3}$$

where $W_j \in \mathbb{R}^{N_t \times N_t}$ are diagonal matrices containing the j^{th} column of W on the diagonal, *e.g.*, $W_j = diag\{w_{1j}, w_{2j}...\}$ and similarly $P_j = diag\{p_{1j}, p_{2j}...\}$. Q is the number of supportive samples for each class.

This objective function (3) maximizes the confidence of the selected supportive samples. The first constraint requires that the supportive samples in the $(k+1)^{th}$ iteration are disjoint from the previously chosen ones which ensures that we keep adding new supportive samples to the source domain. The second constraint ensures that the number of supportive samples for each class is balanced.

The solution to (3) is to find the corresponding Q supportive samples that maximize the confidence with the constraint that old supportive samples are excluded.

Augmented Source Domain Update: After selecting the supportive samples, we update the augmented source data by adding weighted supportive samples to the previous source data:

$$X_j^{(k+1)} = [X_j^{(k)}|X^t W_j^{(k+1)} P_j^{(k+1)}] \quad j = 1,...,C \tag{4}$$

Since the labels of the supportive samples may have error, each selected supportive sample

is weighted by its confidence. The weights indicate the reliability of the labels of the supportive samples and highly confident supportive samples will contribute more to the model.

Dictionary Update: The dictionary is updated by solving the following optimization problem:

$$D_j^{(k+1)} = \underset{D_j, Z_j}{\operatorname{argmin}} ||X_j^{(k+1)} - D_j \cdot Z_j||_F^2 + \lambda ||Z_j||_1 \quad j = 1, ..., C. \tag{5}$$

We solve (5) using the online dictionary learning method [15]. The dictionary obtained in the previous iteration is used as the initial dictionary in the next iteration. In this way, the computational cost is relatively low.

Stopping criterion: One trivial stopping criterion is to stop when there are no new supportive samples for one of the classes. But our goal is to guarantee that the adaptation process monotonically reduces the domain divergence. In this way, the classification error bound in target domain will decrease as stated in [2]. So we design in the next section a domain similarity measure and we only perform adaptation when the domain similarity increases after each iteration. The proposed approach is summarized in Algorithm 1.

Algorithm 1 Incremental dictionary learning for unsupervised DA

Input: Initial dictionary $D^{(0)} = [D_1^{(0)}|...|D_C^{(0)}]$ learned from the source data, the target domain data X^t, similarity measure of two domains $\rho(X^s, X^t)$, number of supportive samples Q per class, parameters λ.

Output: Class labels for target data X^t.

k = 0

repeat

1. Confidence update: For each input data x_i^t, compute the reconstruction error on each $D_j^{(k)}$ using (2). Update each element of the confidence matrix $P^{(k+1)}$ using (1)

2. Supportive sample selection: For each class j, select the supportive samples using $W_j^{(k+1)}$ by maximizing (3).

3. Augmented source domain update: Update the augmented source domain $X_j^{(k+1)}$ by adding the new supportive samples:

$$X_j^{(k+1)} = [X_j^{(k)}|X^t W_j^{(k+1)} P_j^{(k)}] \quad j = 1, ..., C \tag{6}$$

4. Dictionary update: Update each class-specific dictionary $D_j^{(k+1)}$ by minimizing (5)

5. $k \leftarrow k+1$.

until no supportive samples is selected or $\rho(X^{(k+1)}, X^t) <= \rho(X^{(k)}, X^t)$

classify X^t using the final dictionary.

3.2 Theoretical Analysis

In this section, we first introduce the domain similarity measure used for determining the stopping criterion. In order to quantify the domain similarity, several methods have been proposed [6, 17]. However, they need to design the dictionary or do PCA for both domains,

which may be time consuming when the data size is large. We introduce a simple domain similarity measure for X^s and X^t: $\rho(X^s,X^t) = \sqrt{\frac{1}{N_s N_t}\sum_i\sum_j (x_i^{sT}x_j^t)^2} = \sqrt{\frac{tr((X^s)^T X^t (X^t)^T X^s)}{N_s N_t}}$.

Since the classification accuracy on supportive samples is good, the main reason that causes the performance to drop in the target domain is that the source classifier behaves poorly on the non-supportive samples. It indicates that domain mismatch mainly lies between the source samples and the non-supportive samples. If the distance between supportive samples and non-supportive samples is smaller than the distance between the source domain and the non-supportive samples, selecting supportive samples can help reduce the domain mismatch and thus help classification as stated in [2]. The following theorem proves this notion and we present experimental results to validate the theoretical results in Section 4.

Theorem 1. *We divide the target samples into two part, supportive samples X_f and non-supportive samples X_n with N_f and N_n samples. With the definition of ρ above, and if $\rho(X_f,X_n) > \rho(X^s,X_n)$, then the domain similarity (or mismatch) will increase(or decrease) when we add some supportive samples to the source domain:*

$$\rho(X^s_{new},X^t) > \rho(X^s_{old},X^t) \tag{7}$$

where $X^s_{old} = X^s$ and $X^s_{new} = [X^s|X_f]$.

Proof. Since $\rho(X_f,X_n) > \rho(X^s,X_n)$, we have:

$$\begin{aligned}
&\rho^2(X_f,X_n) - \rho^2(X^s,X_n)\\
&= \frac{tr(X_n^T X_f X_f^T X_n)}{N_n N_f} - \frac{tr(X_n^T X^s X^{sT} X_n)}{N_n N_s} = \frac{tr((N_s X_f X_f^T - N_f X^s X^{sT}) X_n X_n^T)}{N_n N_s N_f} > 0.
\end{aligned}$$

Then:

$$\begin{aligned}
&\rho^2(X^s_{new},X^t) - \rho^2(X^s_{old},X^t) = tr({X^s_{new}}^T X^t X^{tT} X^s_{new}) - tr({X^s_{old}}^T X^t X^{tT} X^s_{old})\\
&= \frac{tr(([X^s|X_f][\frac{X^{sT}}{X_f^T}][X_n|X_f][\frac{X_n^T}{X_f^T}])}{(N_s+N_f)(N_h+N_f)} - \frac{tr(([X_n|X_f][\frac{X_n^T}{X_f^T}]X^s X^{sT})}{N_s(N_h+N_f)} > 0\\
&\Leftrightarrow \frac{tr((X^s X^{sT} + X_f X_f^T)(X_n X_n^T + X_f X_f^T)}{(N_s+N_f)} - \frac{tr((X_n X_n^T + X_f X_f^T) X^s X^{sT})}{N_s} > 0\\
&\Leftrightarrow tr((N_s X_f X_f^T - N_f X^s X^{sT}) X_n X_n^T) > 0.
\end{aligned}$$

□

4 Experiments

In this section, we evaluate the proposed method for 2D object classification and face recognition. For object classification, we use the standard benchmark dataset *Office+Caltech* [10, 18] for domain adaptation. For face recognition, we follow [17] and conduct experiment on the CMU-PIE dataset [11]. We compare our method with several state-of-the-art unsupervised DA methods. Experimental results show that our method outperforms all other approaches significantly in most cases.

4.1 Object Recognition

Office+Caltech contains object images of four domains: Amazon (A), Webcam (W), DSLR (D), and Caltech (C). This leads to a total of 12 domain pairs for testing. 10 common classes are selected in all domains. For each class, A, C, D and W have about 100, 100, 15 and 30 images, respectively. We follow the protocol used in [8, 14] to generate the source and target domain data. DeCAF features [5] are used in our experiment.

We compare two non-adaptation (NA) methods, and five state-of-the-art unsupervised DA methods: SVM and Dictionary Learning Based Classification (DLC) are the two NA methods, Subspace Interpolation via Dictionary Learning (SIDL) [17], Geodesic Flow Kernel (GFK) [7], Transfer Joint Matching (TJM) [14], Landmarks [8] and DA-NBNN [21] are the unsupervised DA methods. DLC is implemented using the online dictionary learning method as [15] and is also used as the initial dictionary in the proposed approach.

GFK, SIDL and TJM are based on learning domain-invariant subspaces and they are fully unsupervised. In particular, SIDL share a similar idea with GFS [9], but they use dictionary as basis. Landmarks reweight and select some source samples to assist adaptation, and they also utilize source labels to learn a discriminative classifier. DA-NBNN is a bootstrapping based method and are most closely related to our proposed approach, while our method differ from DA-NBNN in that we use different sample selection and stopping criteria.

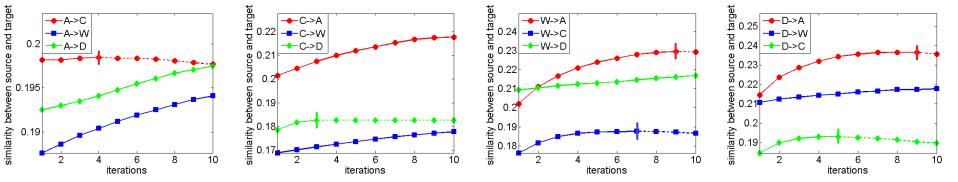
We set $\lambda = 0.05$ and $\sigma^2 = 0.05$. For λ, [24] has shown it is non-sensitive to classification . For σ^2, we use maximum likelihood estimation to estimate it in a similar way as suggested in [4] for each domain. In practice, we calculate the mean for all domains and set a uniform value for simplicity. For A, C, W and D, we set $K = 80, 80, 20$ and 8 respectively. Theoretically, Q can be set uniformly to 1. We can accelerate the convergence speed by setting Q to a reasonably larger value according to the dataset size. For A,C, W and D we set $Q = 8, 8, 2$ and 1, respectively. Due to the space limitation, we only show the sensitivity analysis results on K and Q in section 4.

	Method	A→C	A→D	A→W	C→A	C→D	C→W	W→A	W→D	W→C	D→A	D→C	D→W
NA	SVM	85.04	87.90	78.98	91.44	89.81	80.00	75.68	99.36	71.95	87.06	78.81	98.64
	DLC	85.31	82.17	75.59	91.34	87.90	78.64	78.40	98.72	76.05	88.10	81.56	**99.32**
DA	GFK [7]	77.29	84.71	81.02	88.52	85.99	80.34	81.84	**100**	73.91	85.80	75.96	97.29
	SIDL [17]	84.51	81.53	74.24	90.92	89.81	78.31	75.05	**100**	71.15	87.89	80.14	**99.32**
	TJM [14]	80.14	84.71	75.25	89.04	85.35	76.94	84.86	**100**	78.01	87.37	77.38	98.64
	DA-NBNN [21]	83.44	80.89	76.61	89.67	87.90	80.34	88.00	**100**	82.46	91.34	86.11	97.97
	Landmarks [8]	84.68	85.99	82.37	92.38	**92.35**	84.07	84.03	98.73	71.68	77.04	74.35	95.25
	Proposed method	**86.73**	**92.36**	**88.47**	**93.31**	88.54	**95.59**	**92.80**	**100**	**88.69**	**93.11**	**89.13**	**99.32**

Table 1: Recognition accuracies on 12 pairs of cross-domain unsupervised object recognition. **A: Amazon, C: Caltech, W: Webcam, D: Dslr**

4.1.1 Results on recognition rate:

The recognition rates for all 12 domain pairs are summarized in Table 1. Our proposed approach outperforms other methods on most pairs. We notice that the difficulty for the 12 adaptation tasks vary a lot. Our method tends to gain more over other approaches on more difficult pairs, *e.g.*, A→W, W→C, and behaves similar to other methods on the easier pairs, *e.g.*, D→W. This indicates that the proposed method can boost more on those pairs where domain dissimilarity is relatively large. The reason is that large domain discrepancy provides more space for our adaptation process, which means adding the supportive samples can continuously reduce the domain divergence. In contrast, if the initial domain dissimilarity



(a) A as source domain (b) C as source domain (c) W as source domain (d) D as source domain

Figure 2: The change in domain similarity when the supportive samples are added to the source domain. Solid and dotted lines represent the iterations in which the domain similarity increases and decreases respectively. In our experiments, we only continue our adaptation as long as the similarity value goes up, which is represented by the solid lines before the slash symbols. **A: Amazon, C: Caltech, W: Webcam, D: Dslr**

is small, adding the supportive samples may not reduce the domain distance in a significant way, and our method is likely to stop early and thus behave similar to other techniques.

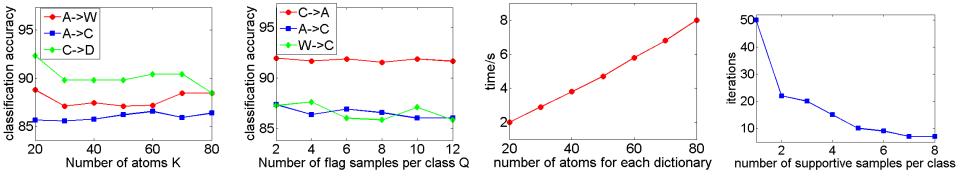
We notice that [8] performs better than baselines when A or C acts as the source domain. It demonstrates the effectiveness of selecting easier adaptive samples. However, its performance drops significantly when W or D acts as source domain. This is because when the source domain is relatively small, the selection of landmarks will further reduce source domain size and leads to insufficient training data. In addition, the performance of [21] is good when W or D acts as the source domain. It reveals that it is very important to exploit the target discriminate information when the source domain is small.

4.1.2 Domain Similarity Evaluation:

In section 3.2, by setting up the stopping criterion, we proved that adding supportive samples reduce the domain divergence under a mild assumption. In this section, we compute the similarity of the source and target domains as the supportive samples are gradually added to the source domain. Results are shown in Fig. 2. Here we set the adaptation iteration number to be 10 to monitor how the similarity value changes as adaptation is performed. In our experiments, we only continue our adaptation as long as the similarity value goes up, which are represented by the solid lines. The dotted lines show that adding more supportive samples may enhance the domain mismatch after some iterations. In this situation, the adaptation process should be terminated.

We compare the changes in domain similarity in Fig. 2 with our classification results in Table 1, and find that we are likely to gain more from our method when the domain similarity value continues to go up as more supportive samples are added to the source, *e.g.*, A→W. It indicates that reducing domain dissimilarity indeed helps the classification task.

It can be observed from Fig. 2 that when the domain similarity, before adaptation, is high it often means the NA methods can work well with high classification performance. However, in this case, as we add more supportive samples to the source domain, the domain similarity may change very little or even decrease, where the adaptation may bring no additional benefits or even harm the classification performance. In contrast, if the original domain similarity value is low, the condition in Theorem 1 is easy to satisfy and the domain similarity can increase continuously as more supportive samples are added. Therefore, better results can be achieved as our adaptation process goes on. This explains why the proposed



(a) dictionary atoms K (b) supportive samples Q (c) dictionary learning time (d) iterations to converge

Figure 3: (a) and (b) show the classification accuracy when K or Q varies. (c) shows the dictionary training time when K varies. (d) shows the number of iterations needed for convergence when Q varies. **A: Amazon, C: Caltech, W: Webcam, D: Dslr**

method performs well in hard cases.

4.1.3 Parameter Sensitivity:

We conduct a sensitivity analysis on parameter K and Q and show results on three pairs. Other pairs behave in a similar way and results are omitted due to space limitation. We can see from Fig. 3 (a) and (b) that the performance does not depend much on K and Q. However, a relatively small K makes the dictionary more compact and a relatively large Q accelerates the rate of convergence, as showed by Fig. 3 (c) and (d). In (c), we use Caltech data to train the dictionary and it indicates that the dictionary learning time increases almost linearly as the number of dictionary atoms increase. In (d), W→C is used to run the algorithm and shows that the number of iterations needed for convergence drops rapidly as Q increases. This is not surprising since the total number of supportive samples should be almost the same regardless of the value of Q.

4.2 Face Recognition

Here we show the experimental results for face recognition on the CMU-PIE dataset. We follow the protocol presented in [17] and consider the proposed approach for face recognition under blur and illumination variations.

4.2.1 Across blur and illumination variance:

We select faces from 34 classes with 21 lighting conditions for each class, in which 11 samples for training and 10 samples for testing. We add Gaussian blur and motion blur to testing samples to evaluate different situations. 6 situations are considered in our experiments: Gaussian blur with standard deviation of 3, 4, or 5, motion blur with lengths of 9, 11, or 13. In our experiments, λ is set to be 0.05. σ^2 is chosen to be 10. We set $K = 10$ and $Q = 1$. We compare our results with the same baseline methods as in section 4.1.

Results presented in Table 2 show that the proposed method outperforms other approaches by a large margin. We see that DLC approach gives us a good initial point for adaptation. It indicates that dictionary-based classification methods are robust to Gaussian blur and motion blur, as well as illumination changes. We normally gain 5% -10% from the initial point and similar to object recognition, we tend to gain more when the initial mismatch between source and target is relatively large. Our method can overcome some blur variations at the

beginning and then further reduce domain mismatch through adaptation from the source to target.

We can also interpret the physical meaning of the supportive sample faces. Since the light condition changes smoothly from source to target, the supportive samples should have closer illumination conditions with the source domain than other non-supportive samples. Once the supportive samples are added to the source domain, the rest of the samples in the target are easier to classify because the supportive samples reduce the illumination mismatch from the source to the target.

Methods	$\sigma = 3$	$\sigma = 4$	$\sigma = 5$	$len = 9$	$len = 11$	$len = 13$
SVM	76.18	71.47	69.71	80.00	74.71	67.06
DLC	88.82	87.35	86.18	91.18	82.06	75.00
GFK [9]	78.53	77.65	74.71	84.41	73.82	64.71
SIDL [22]	80.29	77.94	76.76	85.88	81.18	73.53
TJM [24]	76.18	72.06	70.29	78.24	65.88	53.24
DA-NBNN [21]	62.35	58.53	57.94	65.59	54.12	42.65
Landmarks [8]	80.29	77.94	77.06	82.65	76.18	70.59
Proposed method	**94.70**	**93.24**	**90.29**	**96.47**	**93.24**	**92.35**

Table 2: Recognition accuracies on face recognition under illumination and blur mismatch.

5 Conclusion

In this paper, we propose a novel incremental dictionary learning method for unsupervised domain adaptation. Supportive samples are iteratively selected to smoothly connect the source and target domains. We utilize the supportive samples to reduce the domain mismatch, as well as build a more discriminate classifier, both of which are crucial for classification performance. We design an efficient stopping criterion to guarantee the adaptation reduces the domain dissimilarity monotonically. Extensive experiments on both object classification and face recognition datasets show promising results compared to many state-of-the-art DA methods.

6 Acknowledgments

This research was supported by a MURI grant from the US Office of Naval Research under N00014-10-1-0934.

References

[1] Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 769–776. IEEE, 2013.

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[3] Lorenzo Bruzzone and Mattia Marconcini. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):770–787, 2010.

[4] Yi-Chen Chen, Vishal M Patel, Jaishanker K Pillai, Rama Chellappa, and P Jonathon Phillips. Dictionary learning from ambiguously labeled data. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 353–360. IEEE, 2013.

[5] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of The 31st International Conference on Machine Learning*, pages 647–655, 2014.

[6] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2960–2967. IEEE, 2013.

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

[8] Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Proceedings of The 30th International Conference on Machine Learning*, pages 222–230, 2013.

[9] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 999–1006. IEEE, 2011.

[10] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[11] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

[12] Amaury Habrard, Jean-Philippe Peyrache, and Marc Sebban. Iterative self-labeling domain adaptation for linear structured image classification. *International Journal on Artificial Intelligence Tools*, 22(05), 2013.

[13] Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2006.

[14] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer joint matching for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1410–1417. IEEE, 2014.

[15] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696. ACM, 2009.

[16] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Multiple source adaptation and the rényi divergence. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 367–374. AUAI Press, 2009.

[17] Jie Ni, Qiang Qiu, and Rama Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 692–699. IEEE, 2013.

[18] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010*, pages 213–226. Springer, 2010.

[19] Chun-Wei Seah, Yew-Soon Ong, and Ivor W Tsang. Combating negative transfer from predictive distribution differences. *Cybernetics, IEEE Transactions on*, 43(4):1153–1165, 2013.

[20] Sumit Shekhar, Vishal M Patel, Hien V Nguyen, and Rama Chellappa. Generalized domain-adaptive dictionaries. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 361–368. IEEE, 2013.

[21] Tatiana Tommasi and Barbara Caputo. Frustratingly easy nbnn domain adaptation. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 897–904. IEEE, 2013.

[22] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009.

[23] Meng Yang, David Zhang, and Xiangchu Feng. Fisher discrimination dictionary learning for sparse representation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 543–550. IEEE, 2011.

[24] D Zhang, Meng Yang, and Xiangchu Feng. Sparse representation or collaborative representation: Which helps face recognition? In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 471–478. IEEE, 2011.