Exploiting Low-Rank Structure from Latent Domains for Domain Generalization

Zheng Xu, Wen Li, Li Niu, and Dong Xu

School of Computer Engineering, Nanyang Technological University, Singapore

Abstract. In this paper, we propose a new approach for domain generalization by exploiting the low-rank structure from multiple latent source domains. Motivated by the recent work on exemplar-SVMs, we aim to train a set of exemplar classifiers with each classifier learnt by using only one positive training sample and all negative training samples. While positive samples may come from multiple latent domains, for the positive samples within the same latent domain, their likelihoods from each exemplar classifier are expected to be similar to each other. Based on this assumption, we formulate a new optimization problem by introducing the nuclear-norm based regularizer on the likelihood matrix to the objective function of exemplar-SVMs. We further extend Domain Adaptation Machine (DAM) to learn an optimal target classifier for domain adaptation. The comprehensive experiments for object recognition and action recognition demonstrate the effectiveness of our approach for domain generalization and domain adaptation.

Keywords: Latent domains, domain generalization, domain adaptation, exemplar-SVMs.

1 Introduction

Domain adaptation techniques, which aim to reduce the domain distribution mismatch when the training and testing samples come from different domains, have been successfully used for a broad range of vision applications such as object recognition and video event recognition [23,17,16,10,11,12,6,7,24]. As a related research problem, domain generalization differs from domain adaptation because it assumes the target domain samples are not available during the training process. Without focusing on the generalization ability on the specific target domain, domain generalization techniques aim to better classify testing data from any unseen target domain [26,22]. Please refer to Section 2 for a brief review of existing domain adaptation and domain generalization techniques.

For visual recognition, most existing domain adaptation methods treat each dataset as one domain [23,17,16,10,11,12,6,7,24]. However, the recent works show the images or videos in one dataset may come from multiple hidden domains[20,15]. In [20], Hofffman *et al.* proposed a constrained clustering method to discover the latent domains and also extended [23] for multi-domain adaptation by learning multiple transformation matrices. In [15], Gong *et al.* partitioned

D. Fleet et al. (Eds.): ECCV 2014, Part III, LNCS 8691, pp. 628–643, 2014.

the training samples from one domain into multiple domains by simultaneously maximizing distinctiveness and learnability. However, it is a non-trivial task to discover the characteristic latent domains by explicitly partitioning the training samples into multiple clusters because many factors (*e.g.*, pose and illumination) overlap and interact in images and videos in complex ways [15].

In this work, we propose a new approach for domain generalization by explicitly exploiting the intrinsic structure of positive samples from multiple latent domains without partitioning the training samples into multiple clusters/domains. Our work builds up the recent ensemble learning method exemplar-SVMs, in which we aim to train a set of exemplar classifiers with each classifier learnt by using one positive training sample and all negative training samples. While positive samples may come from multiple latent domains characterized by different factors, for the positive samples captured under similar conditions (*e.g.*, frontal-view poses), their likelihoods from each exemplar classifier are expected to be similar to each other. Using the likelihoods from all the exemplar classifiers as the feature of each positive sample, we assume the likelihood matrix consisting of the features of all positive samples should be low-rank in the ideal case. Based on this assumption, we formulate a new objective function by introducing a nuclear norm based regularizer on the likelihood matrix into the objective function of exemplar-SVMs in order to learn a set of more robust exemplar classifiers for domain generalization and domain adaptation.

To solve the new optimization problem, we further introduce an intermediate variable $\mathbf{F}$ modelling the ideal likelihood matrix, and arrive at a relaxed objective function. Specifically, we minimize the objective function of exemplar-SVMs and the nuclear norm of the ideal likelihood matrix $\mathbf{F}$ as well as the approximation error between $\mathbf{F}$ and the likelihood matrix. Then, we develop an alternating optimization algorithm to iteratively solve the ideal likelihood matrix $\mathbf{F}$ and learn the exemplar classifiers.

During the testing process, we can directly use the whole or a selected set of learnt exemplar classifiers for the domain generalization task when the target domain samples are not available during the training process. For domain adaptation, we propose an effective method to re-weight the selected set of exemplar classifiers based on the Maximum Mean Discrepancy (MMD) criterion, and we further extend the Domain Adaptation Machine (DAM) method to learn an optimal target classifier. We conduct comprehensive experiments for object recognition and human activity recognition using two datasets and the results clearly demonstrate the effectiveness of our approach for domain generalization and domain adaptation.

2 Related Work

Domain adaptation methods can be roughly categorized into feature based approaches and classifier based approaches. The feature based approaches aim to learn domain invariant features for domain adaptation. Kulis *et al.* [23] proposed a distance metric learning method to reduce domain distribution mismatch by

learning asymmetric nonlinear transformation. Gopalan *et al.* [17] and Gong *et al.* [16] proposed two domain adaptation methods by interpolating intermediate domains. To reduce the distribution mismatch, some recent approaches learnt a domain invariant subspace [2] or aligned two subspaces from both domains [14].

Classifier based approaches directly learn the classifiers for domain adaptation, among which SVM based approaches are the most popular ones. Huang *et al.* [21] proposed a domain adaptation approach by re-weighting the source domain samples and then learning a weighted SVM classifier with the learnt weights. Duan *et al.* [10] proposed a new method called Adaptive MKL based on multiple kernel learning (MKL) [32], and a multi-domain adaptation method by selecting the most relevant source domains [13]. The work in [3] developed an approach to iteratively learn the SVM classifier by labeling the unlabeled target samples and simultaneously removing some labeled samples in the source domain.

There are a few works specifically designed for domain generalization. To enhance the domain generalization ability, Muandet *et al.* proposed to learn new domain invariant feature representations [26]. Given multiple source datasets/domains, Khosla *et al.* [22] proposed an SVM based approach, in which the learnt weight vectors that are common to all datasets can be used for domain generalization.

Our work is more related to the recent works for discovering latent domains [20,15]. In [20], a clustering based approach is proposed to divide the source domain into different latent domains. In [15], the MMD criterion is used to partition the source domain into distinctive latent domains. However, their methods need to decide the number of latent domains beforehand. In contrast, our method exploits the low-rank structure from latent domains without requiring the number of latent domains. Moreover, we directly learn the exemplar classifiers without partitioning the data into clusters/domains.

Our work builds up the recent work on exemplar-SVMs [25]. In contrast to [25], we introduce a nuclear norm based regularizer on the likelihood matrix in order to exploit the low-rank structure from latent domains for domain generalization. In multi-task learning, the nuclear norm based regularizer is also introduced to enforce the related tasks share similar weight vectors when learning the classifiers for multiple tasks [1,5]. However, their works assume the training and testing samples come from the same distribution without considering the domain generalization or domain adaptation tasks. Moreover, our regularizer is on the likelihood matrix such that we can better exploit the structure of positive samples from multiple latent domains.

3 Low-Rank Exemplar-SVMs

In this section, we introduce the formulation of our low rank exemplar-SVMs as well as the optimization algorithm. For ease of representation, in the remainder of this paper, we use a lowercase/uppercase letter in boldface to represent a vector/matrix. The transpose of a vector/matrix is denoted by using the superscript $'$. $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{m \times n}$ defines a matrix $\mathbf{A}$ with a_{ij} being its (i, j)-th

element for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. The element-wise product between two matrices $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{m \times n}$ and $\mathbf{B} = [b_{ij}] \in \mathbb{R}^{m \times n}$ is defined as $\mathbf{C} = \mathbf{A} \circ \mathbf{B}$, where $\mathbf{C} = [c_{ij}] \in \mathbb{R}^{m \times n}$ and $c_{ij} = a_{ij} b_{ij}$.

3.1 Formulation

In the exemplar-SVMs model, each exemplar classifier is learnt by using one positive training sample and all the negative training samples. Let $\mathcal{S} = \mathcal{S}^+ \cup \mathcal{S}^-$ denote the set of training samples, in which $\mathcal{S}^+ = \{\mathbf{s}_1^+, \ldots, \mathbf{s}_n^+\}$ is the set of positive training samples, and $\mathcal{S}^- = \{\mathbf{s}_1^-, \ldots, \mathbf{s}_m^-\}$ is the set of negative training samples. Each training sample $\mathbf{s}^+$ or $\mathbf{s}^-$ is a d-dimensional column vector, *i.e.*, $\mathbf{s}^+, \mathbf{s}^- \in \mathbb{R}^d$. In this work, we use the logistic regression function for prediction. Given any sample $\mathbf{t} \in \mathbb{R}^d$, the prediction function can be written as:

$$p(\mathbf{t}|\mathbf{w}_i) = \frac{1}{1 + \exp(-\mathbf{w}_i' \mathbf{t})}, \tag{1}$$

where $\mathbf{w}_i \in \mathbb{R}^d$ is the weight vector in the i-th exemplar classifier trained by using the positive training sample $\mathbf{s}_i^+$ and all negative training samples[1]. By defining a weight matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_n] \in \mathbb{R}^{d \times n}$, we formulate the learning problem as follows,

$$\min_{\mathbf{W}} J(\mathbf{W}) = \min_{\mathbf{W}} \|\mathbf{W}\|_F^2 + C_1 \sum_{i=1}^{n} l(\mathbf{w}_i, \mathbf{s}_i^+) + C_2 \sum_{i=1}^{n} \sum_{j=1}^{m} l(\mathbf{w}_i, \mathbf{s}_j^-), \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix, C_1 and C_2 are the tradeoff parameters analogous to C in SVM, and $l(\mathbf{w}, \mathbf{s})$ is the logistic loss, which is defined as:

$$l(\mathbf{w}_i, \mathbf{s}_i^+) = \log(1 + \exp(-\mathbf{w}_i' \mathbf{s}_i^+)), \tag{3}$$

$$l(\mathbf{w}_i, \mathbf{s}_j^-) \ = \log(1 + \exp(\mathbf{w}_i' \mathbf{s}_j^-)). \tag{4}$$

Now we consider how to discover the latent domains in the training data. Intuitively, if there are multiple latent domains in the training data, the positive training samples should also come from several latent domains. For the positive samples captured under similar conditions (*e.g.*, frontal-view poses), their likelihoods from each exemplar classifier are expected to be similar to each other. Using the likelihoods from all the exemplar classifiers as the feature of each positive sample, we assume the likelihood matrix consisting of the likelihoods of all positive samples should be low-rank in the ideal case. Formally, we denote the likelihood matrix as $\mathbf{G}(\mathbf{W}) = [g_{ij}] \in \mathbb{R}^{n \times n}$, where each $g_{ij} = p(\mathbf{s}_i^+|\mathbf{w}_j)$ is the likelihood of the i-th positive training sample by using the j-th exemplar classifier. To exploit those latent domains, we thus enforce the prediction matrix

[1] Although we do not explicitly use the bias term in the prediction, in our experiments we append 1 to the feature vector of each training sample.

$\mathbf{G}(\mathbf{W})$ to be low-rank when we learn those exemplar-SVMs, namely, we arrive at the following objective function,

$$\min_{\mathbf{W}} J(\mathbf{W}) + \lambda \|\mathbf{G}(\mathbf{W})\|_*, \tag{5}$$

where we use the nuclear norm based regularizer $\|\mathbf{G}(\mathbf{W})\|_*$ to approximate the rank of $\mathbf{G}(\mathbf{W})$. It has been shown that the nuclear norm is the best convex approximation of the rank function over the unit ball of matrices [27]. However, it is a nontrivial task to solve the problem in (5), because the last term is a nuclear norm based regularizer on the likelihood matrix $\mathbf{G}(\mathbf{W})$ and $\mathbf{G}(\mathbf{W})$ is a non-linear term w.r.t. $\mathbf{W}$.

To solve the optimization problem in (5), we introduce an intermediate matrix $\mathbf{F} \in \mathbb{R}^{n \times n}$ to model the ideal $\mathbf{G}(\mathbf{W})$ such that we can decompose the last term in (5) into two parts: on one hand, we expect the intermediate matrix $\mathbf{F}$ should be low-rank as we discussed above; on the other hand, we enforce the likelihood matrix $\mathbf{G}(\mathbf{W})$ to be close to the intermediate matrix $\mathbf{F}$. Therefore, we reformulate the objective function as follows,

$$\min_{\mathbf{W},\mathbf{F}} J(\mathbf{W}) + \lambda_1 \|\mathbf{F}\|_* + \lambda_2 \|\mathbf{F} - \mathbf{G}(\mathbf{W})\|_F^2, \tag{6}$$

which can be solved by alternatingly optimizing two subproblems w.r.t. $\mathbf{W}$ and $\mathbf{F}$. Specifically, the optimization problem w.r.t. $\mathbf{W}$ does not contain the nuclear norm based regularizer, which makes the optimization much easier. Also, the nuclear norm based regularizer only depends on the intermediate matrix $\mathbf{F}$ rather than a non-linear term w.r.t. $\mathbf{W}$ (*i.e.*, the likelihood matrix $\mathbf{G}(\mathbf{W})$) as in (5), thus the optimization problem w.r.t. $\mathbf{F}$ can be readily solved by using the Singular Value Threshold (SVT) method [4] (see Section 3.2 for the details).

Discussions: To better understand our proposed approach, in Figure 1, we show an example of the learnt likelihood matrix $\mathbf{G}(\mathbf{W})$ from the "check watch" category in the IXMAS multi-view dataset by using Cam 0 and Cam 1 as the source domain. After using the nuclear norm based regularizer, we observe the block diagonal property of the likelihood matrix $\mathbf{G}(\mathbf{W})$ in Figure 1(a). In Figure 1(b), we also display some frames from the videos corresponding to the two blocks with large values in $\mathbf{G}(\mathbf{W})$. We observe that the videos sharing higher values in the matrix $\mathbf{G}(\mathbf{W})$ are also visually similar to each other. For example, the first two rows in Figure 1(b) are the videos from similar poses. More interestingly, we also observe our algorithm can group similar videos from different views in one block (*e.g.*, the last three rows in Figure 1(b) are the videos from the same actor), which demonstrates it is beneficial to exploit the latent source domains by using our approach.

3.2 Optimization

In this section, we discuss how to optimize the problem in (6). We optimize (6) by iteratively updating $\mathbf{W}$ and $\mathbf{F}$. The two subproblems w.r.t. $\mathbf{W}$ and $\mathbf{F}$ are described in detail as follows.

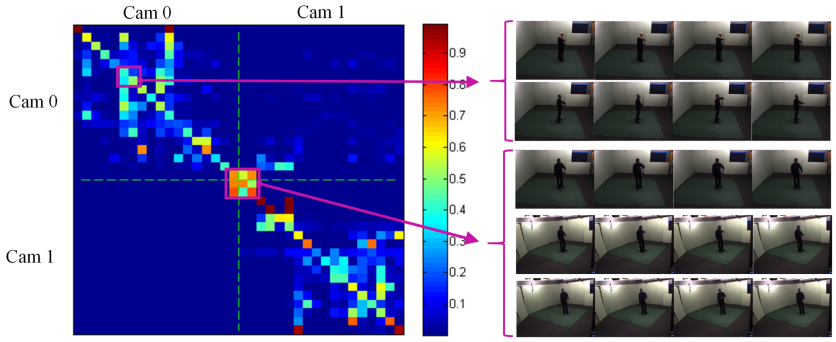

Fig. 1. An illustration of the likelihood matrix $\mathbf{G}(\mathbf{W})$, where we observe the block diagonal property of $\mathbf{G}(\mathbf{W})$ in (a). The frames from the videos corresponding to the two blocks with large values in $\mathbf{G}(\mathbf{W})$ are also visually similar to each other in (b).

Update W: When $\mathbf{F}$ is fixed, the subproblem w.r.t. $\mathbf{W}$ can be written as,

$$\min_{\mathbf{W}} J(\mathbf{W}) + \lambda_2 \|\mathbf{G}(\mathbf{W}) - \mathbf{F}\|_F^2, \tag{7}$$

where the matrix $\mathbf{F}$ is obtained at the k-th iteration, and $\mathbf{G}(\mathbf{W})$ is defined as in Section 3.1. We optimize the above subproblem by using the gradient descent technique. Let us respectively define $\mathbf{S}_1 = [\mathbf{s}_1^+, \ldots, \mathbf{s}_n^+] \in \mathbb{R}^{d \times n}$ and $\mathbf{S}_2 = [\mathbf{s}_1^-, \ldots, \mathbf{s}_m^-] \in \mathbb{R}^{d \times m}$ as the data matrices of positive and negative training samples, and also denote $H(\mathbf{W}) = \|\mathbf{G}(\mathbf{W}) - \mathbf{F}\|_F^2$. Then, the gradients of the two terms in (7) can be derived as follows,

$$\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}} = 2\mathbf{W} + C_1 \mathbf{S}_1 (\mathbf{P}_1 - \mathbf{I}) + C_2 \mathbf{S}_2 \mathbf{P}_2, \tag{8}$$

$$\frac{\partial H(\mathbf{W})}{\partial \mathbf{W}} = 2\mathbf{S}_1 \left(\mathbf{G}(\mathbf{W}) \circ (\mathbf{1}\mathbf{1}' - \mathbf{G}(\mathbf{W})) \circ (\mathbf{G}(\mathbf{W}) - \mathbf{F})\right), \tag{9}$$

where $\mathbf{P}_1 = diag\left(p(\mathbf{s}_i^+ | \mathbf{w}_i)\right) \in \mathbb{R}^{n \times n}$ is a diagonal matrix with each diagonal entry being the prediction on one positive sample by using its corresponding exemplar classifier, $\mathbf{P}_2 = [p(\mathbf{s}_i^- | \mathbf{w}_j)] \in \mathbb{R}^{m \times n}$ is the prediction matrix on all negative training samples by using all exemplar classifiers, $\mathbf{I} \in \mathbb{R}^{n \times n}$ is an identity matrix, and $\mathbf{1} \in \mathbb{R}^n$ is a vector with all entries being 1.

Update F: When $\mathbf{W}$ is fixed, we can calculate the matrix $\mathbf{G} = \mathbf{G}(\mathbf{W})$ at first, then the subproblem w.r.t. $\mathbf{F}$ becomes,

$$\min_{\mathbf{F}} \lambda_1 \|\mathbf{F}\|_* + \lambda_2 \|\mathbf{F} - \mathbf{G}\|_F^2, \tag{10}$$

which can be readily solved by using the singular value thresholding (SVT) method [4,31]. Specifically, let us denote the singular value decomposition of

Algorithm 1. Optimization for Low-rank Exemplar-SVMs (LRE-SVMs)

Input: Training data $\mathcal{S}$, and the parameters $C_1, C_2, \lambda_1, \lambda_2$.

1. Initialize $\mathbf{W} \leftarrow \mathbf{W}^0$, where $\mathbf{W}^0$ is obtained by solving (2).
2. **repeat**
3. Calculate the likelihood matrix $\mathbf{G}(\mathbf{W})$ based on the current $\mathbf{W}$.
4. Solve for $\mathbf{F}$ by optimizing the problem in (10) with the SVT method.
5. Update $\mathbf{W}$ by solving the problem in (7) with the gradient descent method.
6. **until** The objective converges or the maximum number of iterations is reached.

Output: The weight matrix $\mathbf{W}$.

$\mathbf{G}$ as $\mathbf{G} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}'$, where $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n\times n}$ are two orthogonal matrices, and $\boldsymbol{\Sigma} = diag(\sigma_i) \in \mathbb{R}^{n\times n}$ is a diagonal matrix containing all the singular values. The singular value thresholding operator on $\mathbf{G}$ can be represented as $\mathbf{U}\mathcal{D}(\boldsymbol{\Sigma})\mathbf{V}'$, where $\mathcal{D}(\boldsymbol{\Sigma}) = diag((\sigma_i - \frac{\lambda_1}{2\lambda_2})_+)$, and $(\cdot)_+$ is a thresholding operator by assigning the negative elements to be zeros.

Algorithm: We summarize the optimization procedure in Algorithm 1 and name our method as *Low-rank Exemplar-SVMs (LRE-SVMs)*. Specifically, we first initialize the weight matrix $\mathbf{W}$ as $\mathbf{W}^0$, where $\mathbf{W}^0$ is obtained by solving the traditional exemplar-SVMs formulation in (2). Then we calculate the prediction matrix $\mathbf{G}(\mathbf{W})$ by applying the learnt classifiers on all positive samples. Next, we obtain the matrix $\mathbf{F}$ by solving the problem in (10) with the SVT method. After that, we use the gradient descent method to update the weight matrix $\mathbf{W}$. The above steps are repeated until the objective converges.

4 Ensemble Exemplar Classifiers

After training the low-rank exemplar-SVMs as described in Section 3.1, we obtain n exemplar classifiers. To predict the test data, we discuss how to effectively use those learnt classifiers in two situations. One is the domain generalization scenario, where the target domain samples are not available during the training process. And the other one is the domain adaptation scenario, where we have unlabeled data in the target domain during the training process.

4.1 Domain Generalization

In the domain generalization scenario, we have no prior information about the target domain. A simple way is to equally fuse those n exemplar classifiers. Given any test sample $\mathbf{t}$, the prediction $p(\mathbf{t}|\mathbf{W})$ can be calculated as,

$$p(\mathbf{t}|\mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n} p(\mathbf{t}|\mathbf{w}_i), \tag{11}$$

where $p(\mathbf{t}|\mathbf{w}_i)$ is the prediction from the i-th exemplar classifier.

Recall the training samples may come from several latent domains, a better way is to only use the exemplar classifiers in the latent domain which the test data likely belongs to. As mentioned before, on one hand, an exemplar classifier tends to output relatively higher prediction scores for the positive samples from the same latent domain, and relatively lower prediction scores for the positive samples from different latent domains; on the other hand, all exemplar classifiers are expected to output low prediction scores for the negative samples. Therefore, given the test sample $\mathbf{t}$ during the test process, it is beneficial to fuse only the exemplar classifiers that output higher predictions, such that we output a higher prediction score if $\mathbf{t}$ is positive, and a low prediction score if $\mathbf{t}$ is negative. Let us define $\mathcal{T}(\mathbf{t}) = \{\, i \mid p(\mathbf{t}|\mathbf{w}_i)$ is one of the top K prediction scores for $i = 1, \ldots, n\}$ as the set of the indices of those selected exemplar classifiers, then the prediction on this test sample can be obtained as,

$$p(\mathbf{t}|\mathbf{W}) = \frac{1}{K} \sum_{i:i\in\mathcal{T}(\mathbf{t})} p(\mathbf{t}|\mathbf{w}_i), \tag{12}$$

where K is the predefined number of exemplar classifiers that output high prediction scores for the test sample $\mathbf{t}$.

4.2 Domain Adaptation

When we have unlabeled data in the target domain during the training process, we can further assign different weights to the learnt exemplar classifiers to better fuse the exemplar classifiers for predicting the test data from the target domain. Intuitively, when the training data of one exemplar classifier is closer to the target domain, we should assign a higher weight to this classifier and vice versa. Let us denote the target domain samples as $\{\mathbf{t}_1, \ldots, \mathbf{t}_u\}$, where u is the number of samples in the target domain. Based on the Maximum Mean Discrepancy (MMD) criterion [18], we define the distance between the training data of one exemplar classifier and the target domain as follows,

$$d_i = \| \frac{1}{n+m} \left(n\phi(\mathbf{s}_i^+) + \sum_{j=1}^{m} \phi(s_j^-) \right) - \frac{1}{u} \sum_{j=1}^{u} \phi(\mathbf{t}_j) \|^2, \tag{13}$$

where $\phi(\cdot)$ is a nonlinear feature mapping function induced by the Gaussian kernel. We assign a higher weight n to the positive sample s_i^+ when calculating the mean of source domain samples, since we only use one positive sample for training the exemplar classifier at each time. In other words, we duplicate the positive sample $\mathbf{s}_i^+$ for n times and then combine the duplicated positive samples with all the negative samples to calculate the distance with the target domain.

With the above distance, we then obtain the weight for each exemplar classifier by using the RBF function as $v_i = \exp(-d_i/\sigma)$, where σ is the bandwidth parameter, and is set to be the median value of all distances. Then, the prediction on a test sample $\mathbf{t}$ can be obtained as,

$$p(\mathbf{t}|\mathbf{W}) = \sum_{i:i\in\mathcal{T}(\mathbf{t})} \tilde{v}_i p(\mathbf{t}|\mathbf{w}_i), \tag{14}$$

where $\mathcal{T}(\mathbf{t})$ is defined as in Section 4.1, and $\tilde{v}_i = v_i / \sum_{i:i\in\mathcal{T}(\mathbf{t})} v_i$.

One potential drawback with the above ensemble method is that we need to perform the predictions for n times, and then fuse the top K prediction scores. Inspired by Domain Adaptation Machine [13], we propose to learn a single target classifier on the target domain by leveraging the predictions from the exemplar classifiers. Specifically, let us denote the target classifier as $f(\mathbf{t}) = \tilde{\mathbf{w}}'\phi(\mathbf{t}) + b$. We formulate our learning problem as follows,

$$\min_{\tilde{\mathbf{w}},b,\xi_i,\xi_i^*,\mathbf{f}} \ \frac{1}{2}\|\tilde{\mathbf{w}}\|^2 + C\sum_{i=1}^{u}(\xi_i + \xi_i^*) + \frac{\lambda}{2}\Omega(\mathbf{f}), \tag{15}$$

$$\text{s.t.} \ \ \tilde{\mathbf{w}}'\phi(\mathbf{t}_i) + b - f_i \le \epsilon + \xi_i, \quad \xi_i \ge 0, \tag{16}$$

$$f_i - \tilde{\mathbf{w}}'\phi(\mathbf{t}_i) - b \le \epsilon + \xi_i^*, \quad \xi_i^* \ge 0, \tag{17}$$

where $\mathbf{f} = [f_1, \ldots, f_u]'$ is an intermediate variable, λ and C are the tradeoff parameters, ξ_i and ξ_i^* are the slack variables in the ϵ-insensitive loss similarly as in SVR, and ϵ is a predifined small positive value in the ϵ-insensitive loss. The regularizer $\Omega(\mathbf{f})$ is a smoothness function defined as follows,

$$\Omega(\mathbf{f}) = \sum_{j=1}^{u} \sum_{i:i\in\mathcal{T}(\mathbf{t}_j)} \tilde{v}_i \left(f_j - p(\mathbf{t}_j|\mathbf{w}_i)\right)^2, \tag{18}$$

where we enforce each intermediate variable f_j to be similar to the prediction scores of the selected exemplar classifiers in $\mathcal{T}(\mathbf{t}_j)$ for the target sample $\mathbf{t}_j$. In the above problem, we use the ϵ-insensitive loss to enforce the prediction score from target classifier $f(\mathbf{t}_j) = \tilde{\mathbf{w}}'\phi(\mathbf{t}_j) + b$ to be close to the intermediate variable f_j. At the same time, we also use a smoothness regularizer to enforce the intermediate variable f_j to be close to the prediction scores of the selected exemplar classifiers in $\mathcal{T}(\mathbf{t}_j)$ for the target sample $\mathbf{t}_j$. Intuitively, when $\tilde{v}_i$ is large, we enforce the intermediate variable f_j to be closer to $p(\mathbf{t}_j|\mathbf{w}_i)$, and vice versa. Recall the weight $\tilde{v}_i$ models the importance of the i-th exemplar classifier for predicting the target sample, we expect the learnt classifier $f(\mathbf{t})$ performs well for predicting the target domain samples.

By introducing the dual variables $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_u]'$ and $\boldsymbol{\alpha}^* = [\alpha_1^*, \ldots, \alpha_u^*]'$ for the constraints in (16) and (17), we arrive at its dual form as follows,

$$\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} \ \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)\tilde{\mathbf{K}}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \mathbf{p}'(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \epsilon\mathbf{1}_u'(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*), \tag{19}$$

$$\text{s.t.} \ \ \mathbf{1}'\boldsymbol{\alpha} = \mathbf{1}'\boldsymbol{\alpha}^*, \mathbf{0} \le \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \le C\mathbf{1}, \tag{20}$$

where $\tilde{\mathbf{K}} = \mathbf{K} + \frac{1}{\lambda}\mathbf{I} \in \mathbb{R}^{u\times u}$ with $\mathbf{K}$ being the kernel matrix of the target domain samples, $\mathbf{p} = [p(\mathbf{t}_1|\mathbf{W}), \ldots, p(\mathbf{t}_u|\mathbf{W})]'$ with each entry $p(\mathbf{t}_j|\mathbf{W})$ defined in (14) being the "virtual label" for the target sample $\mathbf{t}_j$ as in DAM [13]. In DAM [13], the virtual labels of all the target samples are obtained by fusing the same set of source classifiers. In contrast, we use the predictions from different selected exemplar classifiers to obtain the virtual labels of different target samples. Therefore, DAM can be treated as a special case of our work by using the same classifiers for all test samples in (15).

5 Experiments

In this section, we evaluate our low-rank exemplar-SVMs (LRE-SVMs) approach for domain generalization and domain adaptation, respectively.

5.1 Experimental Setup

Following the work in [15], we use the Office-Caltech dataset [28,16] for visual object recognition and the IXMAS dataset [30] for multi-view action recognition.

Office-Caltech [28,16] dataset contains the images from four domains denoted by A, C, D, and W, in which the images are from Amazon, Caltech-256, and two more datasets captured with digital SLR camera and webcam, respectively. The ten common categories among the 4 domains are utilized in our evaluation. We extract the DeCAF_6 feature [9] for the images in the Office-Caltech dataset, which has achieved promising results in visual recognition.

IXMAS dataset [30] contains the videos from eleven actions captured by five cameras (Cam 0, Cam 1, ..., Cam 4) from different viewpoints. Each of the eleven actions is performed three times by twelve actors. To exclude the irregularly performed actions, we keep the first five actions (check watch, cross arms, scratch head, sit down, get up) performed by six actors (*Alba, Andreas, Daniel, Hedlena, Julien, Nicolas*), as suggested in [15]. We extract the dense trajectories features [29] from the videos, and use K-means clustering to build a codebook with $1,000$ clusters for each of the five descriptors (*i.e.*, trajectory, HOG, HOF, MBHx, MBHy). The bag-of-words features are then concatenated to a $5,000$ dimensional feature for each video sequence.

Following [15], we treat the images from different sources in the Office-Caltech dataset as different domains, and treat the videos from different viewpoints in the IXMAS dataset as different domains, respectively. In our experiments, we mix several domains as the source domain for training the classifiers and use the remaining domains as the target domain for testing. For the domain generalization task, the samples from the target domain are not available during the training process. For the domain adaptation task, the unlabeled samples from the target domain can be used to reduce the domain distribution mismatch in the training process.

We compare our low-rank exemplar-SVMs with several state-of-the-art unsupervised domain adaptation methods, as well as the methods specifically proposed for discovering the latent domains. Note that our approach does not require domain labels for both domain generalization and domain adaptation tasks.

5.2 Domain Generalization

We first evaluate our low-rank exemplar-SVMs (LRE-SVMs) for domain generalization. We compare our proposed method with the domain generalization method by undoing dataset bias (Undo-Bias) in [22], and the latent domain discovering methods [20,15]. We additionally report the results from the discriminative sub-categorization(Sub-C) method [19], as it can also be applied to

Table 1. Recognition accuracies (%) of different methods for domain generalization. Our LRE-SVMs approach does not require domain labels or target domain data during the training process. The best results are denoted in boldface.

Source	A,C	D,W	C,D,W	Cam 0,1	Cam 2,3,4	Cam 0,1,2,3
Target	D,W	A,C	A	Cam 2,3,4	Cam 0,1	Cam 4
SVM	82.68	76.06	90.73	71.70	63.83	56.61
Sub-C [19]	82.61	78.65	90.75	78.11	76.90	64.04
Undo-Bias [22]	80.49	69.98	90.98	69.03	60.56	56.84
[20](Ensemble)	79.23	68.06	80.75	71.55	51.02	49.70
[20](Match)	71.26	61.42	72.03	63.81	60.04	48.91
[15](Ensemble)	84.01	77.11	91.65	75.04	68.98	57.64
[15](Match)	80.63	76.52	90.84	71.59	60.73	55.37
E-SVMs	82.73	80.85	91.47	76.86	68.04	72.98
LRE-SVMs	**84.59**	**81.17**	**91.87**	**79.96**	**80.15**	**74.97**

our application. As the Undo-Bias method [22] requires the domain label information, we provide the groundtruth domain labels to train the classifier for this method. For all other methods, we mix multiple domains as the source domain for training the classifiers.

For the latent domain discovering methods [20,15], after partitioning the source domain data into different domains using their methods, we train an SVM classifier on each domain, and then fuse those classifiers for predicting the test samples. We employ two strategies to fuse the learnt classifiers as suggested in [15], which are referred to as the *ensemble* strategy and the *match* strategy, respectively. The ensemble strategy is to re-weight the decision values from different SVM classifiers by using the domain probabilities learnt with the method in [20]. In the match strategy, we first select the most relevant domain based on the MMD criterion, and then use the SVM classifier from this domain to predict the test samples.

Moreover, we also report the results from the baseline SVM method, which is trained by using all training samples in the source domain. The results from exemplar-SVMs (E-SVMs) are also reported, which is a special case of our proposed LRE-SVMs, and we also use the method in (12) to fuse the selected top K exemplar classifiers for the prediction. For our method, we empirically fix $K = 5$, and $C_1 = 10C_2$ for all our experiments. We set the parameters $C_2 = 0.01$, $\lambda_1 = 100$, and $\lambda_2 = 1000$ on the Office-Caltech dataset, and $C_2 = 1$ and $\lambda_1 = \lambda_2 = 10$ on the IXMAS datast. For baseline methods, we choose the optimal parameters according to their recognition accuracies on the test dataset.

The experimental results on two datasets are summarized in Table 1. We observe that the Sub-C method is comparable or better than the SVM method. The results of Undo-Bias are worse than SVM in most cases even after providing the ground-truth domain labels. One possible explanation is that there are many factors (*e.g.*, pose and illumination) interacting in images and videos in complex ways [15], so even the ground truth domain labels may not be the optimal

ones for learning classifiers for domain generalization, which is also one of our motivations for this work.

For the two latent domain discovering method [20,15], the recently published method by Gong *et al.* [15] achieves quite competitive results when using the ensemble strategy (*i.e.*, [15](Ensemble) in Table 1). It achieves better results on all six cases when compared with SVM, which demonstrates it is beneficial to discover latent domains in the source domain. However, the method in [20] is not as effective as [15]. We also observe the match strategy generally achieves worse results than the ensemble strategy for those latent domain discovering methods, although the target domain information is used to select the most relevant discovered source domain in the testing process.

Our proposed LRE-SVMs method achieves the best results in all six cases on two datasets, which clearly demonstrates the effectiveness of our method by exploiting the low-rank structure in the source domain for domain generalization. We also observe that our special case (*i.e.*, the exemplar-SVMs (E-SVMs) method) also achieves better results than SVM. Note we also apply the prediction method using (12) for E-SVMs. By selecting the most relevant classifiers, we combine a subset of exemplar classifiers for predicting each test sample, leading to good results. By further exploiting the low-rank structure in the source domain, we implicitly employ the information from latent domains in our LRE-SVMs. In this way, the selected top K exemplar classifiers are more likely from the same latent domain that the test sample belongs to. Thus, our LRE-SVMs method outperforms its special case E-SVMs in all six cases for domain generalization.

5.3 Domain Adaptation

In this section, we further compare our proposed method with the baselines for the domain adaptation task, in which the unlabeled samples from the target domain are available in the training process. For domain adaptation, we adopt the approach proposed in Section 4.2 to fuse the exemplar classifiers learnt by using our LRE-SVMs method, and we refer to our approach for domain adaptation as *LRE-SVMs-DA*. We take the IXMAS multiview action recognition dataset as an example to report the results.

We first investigate the state-of-the-art unsupervised domain adaptation methods, including Kernel Mean Matching (KMM) [21], Sampling Geodesic Flow (SGF) [17], Geodesic Flow Kernel (GFK) [16], Selective Transfer Machine(STM) [8], Domain Invariant Projection (DIP) [2], and Subspace Alignment (SA) [14]. For all those methods, we combine the videos captured from multiple cameras to form one combined source domain, and use the remaining samples as the target domain. Then we apply all the methods for domain adaptation. For the feature-based approaches (*i.e.*, SGF, GFK, DIP and SA), we train an SVM classifier after obtaining the domain invariant features/kernels with those methods. We also select the best parameters for those baseline methods according to the test results.

Table 2. Recognition accuracies (%) of different methods for domain adaptation. The best results are denoted in boldface.

Source		Cam 0,1	Cam 2,3,4	Cam 0,1,2,3
Target		Cam 2,3,4	Cam 0,1	Cam 4
SVM		71.70	63.83	56.61
KMM		73.92	42.22	52.57
SGF		60.37	69.04	28.66
GFK		64.87	55.53	42.16
STM		68.69	70.53	51.05
DIP		65.20	70.03	62.92
SA		73.35	77.92	49.59
GFK (latent)	[20] (Match)	61.33	58.77	46.62
	[20] (Ensemble)	65.32	55.01	42.09
	[15] (Match)	65.32	64.43	47.22
	[15] (Ensemble)	69.12	68.87	51.30
SA (latent)	[20] (Match)	58.49	56.27	55.87
	[20] (Ensemble)	63.01	62.05	62.69
	[15] (Match)	66.27	67.00	63.01
	[15] (Ensemble)	71.04	76.64	72.26
DAM (latent)	[20]	77.92	76.99	53.76
	[15]	77.32	73.94	62.47
LRE-SVMs-DA		**81.79**	**82.43**	**75.26**

The results of those baseline methods are reported in Table 2. We also include the baseline SVM method trained by using all the source domain samples for the comparison. The cross-view action recognition is a challenging task. As a result, most unsupervised domain adaptation methods cannot achieve promising results on this dataset, and they are worse than SVM in many cases. The recently proposed method SA [14] and DIP [2] achieves relatively better results, which are better than SVM on two out of three cases.

We further investigate the latent domain discovering methods [20,15]. We use their methods to divide the source domain into several latent domains. Then, we follow [15] to perform the GFK [16] method between each discovered latent domain and the target domain to learn a new kernel for reducing the domain distribution mismatch, and train SVM classifiers using the learnt kernels. Then we also use the two strategies (*i.e.*, ensemble and match) to fuse the SVM classifiers learnt from different latent domains. Moreover, as the SA method achieves better results than GFK on the combined source domain, we further use the SA method to replace the GFK method for reducing the domain distribution mismatch between each latent domain and the target domain. The other steps are as the same as those when using the GFK method. We report the results using latent domain discovering methods [20,15] combined with GFK and SA in Table 2, which are denoted as *GFK(latent)* and *SA(latent)*, respectively. As our method is also related to the DAM method [13], we also report the results of the

DAM method by treating the discovered latent domains with [20,15] as multiple source domains, which is referred to as *DAM(latent)*.

From Table 2, we observe GFK(latent) using the latent domains discovered by [15] is generally better when compared with GFK(latent) using the latent domains discovered by [20]. By using the latent domains discovered by [15], the results of GFK(latent) using both match and ensemble strategies are better than those of GFK on the combined source domain. However, most results from GFK(latent) are still worse than SVM, possibly because the GFK method cannot effectively handle the domain distribution mismatch between each discovered latent domain and the target domain. When using the SA method to replace GFK, we observe the results from SA(latent) in all three cases are improved when compared with their corresponding results from GFK(latent) by using the latent domains discovered by [15]. Moreover, we also observe DAM(latent) outperforms SVM in all cases or most cases when using the latent source domains discovered by [15] or [20].

Our method achieves the best results in all three cases, which again demonstrates the effectiveness of our proposed LRE-SVMs-DA for exploiting the low-rank structure in the source domain. Moreover, our method LRE-SVMs-DA outperforms LRE-SVMs on three cases (see Table 1). Note LRE-SVMs does not use the target domain unlabeled samples during the training process. The results further demonstrate the effectiveness of our domain adaptation approach LRE-SVMs-DA for coping with the domain distribution mismatch in the domain adaptation task.

6 Conclusions

In this paper, we have proposed a new method called Low-rank Exemplar-SVMs (LRE-SVMs) for domain generalization by exploiting the low-rank structure of positive training samples from multiple latent source domains. Based on the recent work on exemplar-SVMs, we propose to exploit the low-rank structure in the source domain by introducing a nuclear-norm based regularizer on the likelihood matrix consisting of the likelihoods of all positive samples from all exemplar classifiers. To further handle the domain distribution mismatch between the training and test data, we further develop an effective method to re-weight the selected set of exemplar classifiers based on the Maximum Mean Discrepancy (MMD) criterion, and extend the Domain Adaptation Machine (DAM) method to learn a better target classifier. The comprehensive experiments have demonstrated the effectiveness of our approach for domain generalization and domain adaptation.

Acknowledgement. This work is supported by the Singapore MoE Tier 2 Grant (ARC42/13).

References

1. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. NIPS (2007)
2. Baktashmotlagh, M., Harandi, M., Brian Lovell, M.S.: Unsupervised domain adaptation by domain invariant projection. In: ICCV (2013)

3. Bruzzone, L., Marconcini, M.: Domain adaptation problems: A DASVM classification technique and a circular validation strategy. T-PAMI 32(5), 770–787 (2010)
4. Cai, J., Cands, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization 20(4), 1956–1982 (2010)
5. Chen, J., Zhou, J., Ye, J.: Integrating low-rank and group-sparse structures for robust multi-task learning. In: SIGKDD (2011)
6. Chen, L., Duan, L., Xu, D.: Event recognition in videos by learning from heterogeneous web sources. In: CVPR, pp. 2666–2673 (2013)
7. Chen, L., Li, W., Xu, D.: Recognizing RGB images by learning from RGB-D data. In: CVPR, pp. 1418–1425 (2014)
8. Chu, W.S., la Torre, F.D., Cohn, J.F.: Selective transfer machine for personalized facial action unit detection. In: CVPR (2013)
9. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: DeCAF: A deep convolutional activation feature for generic visual recognition. In: ICML (2014)
10. Duan, L., Xu, D., Tsang, I.W., Luo, J.: Visual event recognition in videos by learning from web data. T-PAMI 34(9), 1667–1680 (2012)
11. Duan, L., Tsang, I.W., Xu, D.: Domain transfer multiple kernel learning. T-PAMI 34(3), 465–479 (2012)
12. Duan, L., Xu, D., Chang, S.F.: Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In: CVPR, pp. 1338–1345 (2012)
13. Duan, L., Xu, D., Tsang, I.W.: Domain adaptation from multiple sources: A domain-dependent regularization approach. T-NNLS 23(3), 504–518 (2012)
14. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: ICCV (2013)
15. Gong, B., Grauman, K., Sha, F.: Reshaping visual datasets for domain adaptation. In: NIPS (2013)
16. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: CVPR (2012)
17. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: An unsupervised approach. In: ICCV (2011)
18. Gretton, A., Gorgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. JMLR 23, 723–773 (2012)
19. Hoai, M., Zisserman, A.: Discriminative sub-categorization. In: CVPR (2013)
20. Hoffman, J., Kulis, B., Darrell, T., Saenko, K.: Discovering latent domains for multisource domain adaptation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 702–715. Springer, Heidelberg (2012)
21. Huang, J., Smola, A., Gretton, A., Borgwardt, K., Scholkopf, B.: Correcting sample selection bias by unlabeled data. In: NIPS (2007)
22. Khosla, A., Zhou, T., Malisiewicz, T., Efros, A.A., Torralba, A.: Undoing the damage of dataset bias. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 158–171. Springer, Heidelberg (2012)
23. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: CVPR (2011)
24. Li, W., Duan, L., Xu, D., Tsang, I.W.: Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. T-PAMI 36(6), 1134–1148 (2014)

25. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-SVMs for object detection and beyond. In: ICCV (2011)
26. Muandet, K., Balduzzi, D., Schölkopf, B.: Domain generalization via invariant feature representation. In: ICML (2013)
27. Recht, B., Fazel, M., Parrilo, P.A.: Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization. SIAM Review 52(3), 471–501 (2010)
28. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010)
29. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. International Journal of Computer Vision 103(1), 60–79 (2013)
30. Weinland, D., Boyer, E., Ronfard, R.: Action recognition from arbitrary views using 3d exemplars. In: ICCV (2007)
31. Xiao, S., Tan, M., Xu, D.: Weighted block-sparse low rank representation for face clustering in videos. In: ECCV (2014)
32. Xu, X., Tsang, I.W., Xu, D.: Soft margin multiple kernel learning. T-NN 24(5), 749–761 (2013)