Scientia Iranica D (2019) 26(6), 3515–3539

Sharif University of Technology
Scientia Iranica
Transactions D: Computer Science & Engineering and Electrical Engineering
http://scientiairanica.sharif.edu

Joint distribution adaptation via feature and model matching

M. Mardani and J. Tahmoresnezhad[*]

Faculty of IT & Computer Engineering, Urmia University of Technology, Urmia, Iran.

Received 31 October 2017; received in revised form 25 May 2018; accepted 8 December 2018

KEYWORDS
Pattern recognition;
Domain adaptation;
Transfer learning;
Feature transformation;
Model matching.

Abstract. It is usually supposed that the training (source domain) and test (target domain) data follow similar distributions and feature spaces in most pattern recognition tasks. However, in many real-world applications, particularly in visual recognition, this hypothesis has frequently been violated. Thus, the trained classifier for the source domain performs poorly in the target domain. This problem is known as domain shift problem. Domain adaptation and transfer learning are promising techniques towards an effective and robust classifier to tackle the shift problem. In this paper, a novel scheme is proposed for domain adaptation, named Joint Distribution Adaptation via Feature and Model Matching (JDAFMM), in which feature transform and model matching are jointly optimized. By introducing regularization performed between the marginal and conditional distribution shifts across the domains, data drift can be successfully adapted as much as possible along with empirical risk minimization and rate of consistency maximization between manifold and prediction functions. Extensive experiments were conducted to evaluate the performance of the proposed model against other machine learning and domain adaptation methods in three types of visual benchmark datasets. Our experiments illustrated that our JDAFMM significantly outperformed other baseline and state-of-the-art methods.

1. Introduction

The main assumption of the machine learning and pattern recognition tasks is that the training and test data should be sampled in similar distribution patterns [1]. However, this assumption has frequently been violated in many real-world applications. For example, in computer vision task, imagine that we are to learn a classifier in order to recognize objects in images captured by a mobile phone camera while we have no labeled images. At first, we train a classifier on a related labeled dataset, e.g., flicker images. Next, we evaluate it on the target dataset, i.e., mobile phone images. A simple review proved that the performance of the trained model was significantly low since the flicker and mobile phone images, even with similar objects, had a considerable distribution difference as a result of many factors, including poses, illuminations, and expressions.

The distribution difference between the training and test sets gives rise to an issue denominated as the domain shift problem. This problem occurs not only in image recognition task, but also in other machine learning tasks [2,3] such as speech and language processing [4,5], statistics, and computer vision [6,7]. However, in this work, we are to find a solution so as to deal with the domain shift problem to improve the performance of the model.

The domain shift problem can be solved by

[*]. *Corresponding author. Tel.: +98 44 31980236;*
Fax: +98 44 31980236
Email addresses: mehri.mardani@it.uut.ac.ir (M. Mardani); j.tahmores@it.uut.ac.ir (J. Tahmoresnezhad)

doi: 10.24200/sci.2018.5487.1304

Domain Adaptation (DA) and *Transfer Learning (TL)* techniques via training a robust classifier against any distribution mismatch across the domains. The general assumption of DA is that the marginal distributions of source domain (X_s) and target domain (X_t) are different, i.e., $P(X_s) \neq P(X_t)$; however, the conditional distribution across domains is the same, i.e., $P(Y_s \mid X_s) \simeq P(Y_t \mid X_t)$ [8]. Further, in TL, it is assumed that $P(X_s) \simeq P(X_t)$ and $P(Y_s \mid X_s) \neq P(Y_t \mid X_t)$ [8].

DA approaches, depending on the available information in the target domain, are divided into two general categories: (1) unsupervised domain adaptation approaches in which there are no labeled data in the target domain [4,9,10], and (2) semi-supervised domain adaptation approaches, in which a small part of the target domain is labeled [11,12]. Recent studies have shown that the unsupervised domain adaptation tasks are more practical and challenging in real-world applications [13].

In most available researches, the origin of domain shift is investigated only from the marginal or conditional distribution mismatch perspectives, while, in some real-world applications (e.g., visual domains), distribution mismatch across the source and target domains originates from the difference of domains in both marginal and conditional distributions. Recently, some researchers provided approaches in which both the marginal and conditional distributions matched using kernel density estimation [14], sample selection [15], or two-stage reweighting [16]; however, the main drawback to the proposed approaches was the requirement of labeled data in the target domain.

In this paper, we address unsupervised domain shift problem where the difference between the marginal and conditional distributions across domains is too much. We put forward a two-phase framework named "Joint Distribution Adaptation via Feature and Model Matching (JDAFMM)" in which the difference between both the marginal and conditional distributions in a principal dimensionality reduction procedure is reduced. In the first phase, JDAFMM projects the source and target domains in a shared low-dimensional subspace based on Principal Component Analyses (PCA) [17] and then, employs the nonparametric Maximum Mean Discrepancy (MMD) [18] to minimize the difference between the marginal and conditional distributions across the domains. In the second phase, since the source and target domains are similar in terms of distribution (in a new feature space), JDAFMM benefits from the source domain as labeled instances and the target domain as unlabeled instances; then, it learns an adaptive classifier using both of them. Specifically, unlike the other unsupervised DA approaches, we not only employ unlabeled target instances to find a unified feature transformation, but also utilize them to learn an adaptive classifier. The learned adaptive classifier aims at minimizing the empirical risk of prediction function in the source domain and maximizing the rate of consistency between the prediction function and the geometric data structure.

The performance of JDAFMM is evaluated with respect to three types of benchmark domain adaptation datasets. Our comprehensive experiments demonstrate that JDAFMM outperforms other state-of-the-art DA and dimensionality reduction methods in most cases. In addition, JDAFMM achieves a significant improvement in terms of the average classification accuracy (10.55%) compared to the best available method.

The rest of the paper is organized as follows. In Section 2, a short review of the related studies is presented. The proposed method is introduced in Section 3. The experimental setup and comparisons are provided in Sections 4 and 5. Finally, our conclusion and suggestions for future research are presented in Section 6.

2. Related work

In recent years, DA has attracted considerable attention as one of the promising solutions to the domain shift problem. The focus of DA approaches is on reducing the distribution mismatch between the source and target domains via three various frameworks, namely: (1) instance-based, (2) feature-based, and (3) model-based methods.

Instance-based methods [16,19,20] reweight the samples of the source data in order to adapt the source and target domains. Indeed, the main inspiration for the instance-based methods is to learn an optimal model for the reweighted source data to apply to unlabeled target data. Landmark selection [21] exploits MMD to discover a subset of labeled samples in the source domain with the highest similarity to the target domain in terms of distribution, i.e., landmarks. Indeed, landmarks are used as a bridge across the source and target domains. Kernel-based feature Mapping with Ensemble (KMapEnsemble) [15] is an effective method that benefits from both adaptive kernel- and sample-based methods. KMapEnsemble projects the marginal distribution of the source and target data in a common subspace and employs a sample selection method to reduce the conditional distribution mismatch between the source and target domains.

Feature-based methods [5,11,22-25] minimize the distribution mismatch across the source and target domains, typically by constructing a common feature space. In fact, the feature-based methods transfer the source and target data into a common feature space based on the shared features of the source and target domains, train a model for the embedded source data, and apply it to the unlabeled target data. There are several feature-based approaches, e.g., Maximum

Mean Discrepancy Embedding (MMDE) [26], Transfer Component Analysis (TCA) [3], Geodesic Flow Kernel (GFK) [9], Joint Distribution Adaptation (JDA) [27], and Visual Domain Adaptation (VDA) [8], integrating PCA with DA approaches to construct a new feature representation.

MMDE measures the divergence between the source and target domains using MMD and discovers invariant features across them along with variance preservation of input data. TCA benefits from MMD as a distance measure across domains and projects the source and target domains in a latent subspace based on transfer components. GFK considers an infinite number of subspaces along the geodesic path on a Grassmann manifold. GFK adapts the domain shift problem via integrating an infinite number of intermediate subspaces and then, models statistical properties of data drift. JDA projects source and target data in a common feature space, such that the differences of both the marginal and conditional distributions across domains are minimized. VDA minimizes the mismatch between joint marginal and conditional distributions across domains and maximizes the discrimination margin among various classes.

Model-based methods [28,29] learn an adaptive classifier for the target data via joint parameters or priors derived from the source model. In fact, the model-based approaches facilitate domain adaptation in a semi-supervised manner and exploit Support Vector Machine (SVM) to find an adaptive classifier [28,30]. Yang et al. [30] proposed the Adaptive Support Vector Machine (ASVM) to exploit decision boundaries of the source data in determining boundaries of target data. Bruzzone and Marconcini [31] proposed a Domain Adaptation Support Vector Machine (DASVM) to learn a classifier in an iterative manner. In each iteration, DASVM predicts the labels of unlabeled target data and removes some labeled source data, which are not fruitful in obtaining a classifier for target label prediction. Long et al. [32] proposed Adaptive Regularization-based Transfer Learning (ARTL) to learn an adaptive classifier in an unsupervised manner. ARTL tends to optimize the following three objectives: (1) minimizing the structural risk functional, (2) minimizing the joint distribution mismatch between domains, and (3) maximizing the manifold consistency underlying the marginal distribution.

This paper introduces a novel framework for an unsupervised DA problem, which benefits from both feature- and model-based approaches. JDAFMM discovers a common feature subspace in which the mismatch between marginal and conditional distributions across the source and target domains is reduced. Next, JDAFMM learns an adaptive classifier using labeled source data and unlabeled target data to build a robust model against data drift across the source and target domains. Unlike other available DA methods, JDAFMM utilizes the unlabeled target data simultaneously to adapt the domains and build an adaptive classifier.

3. Definition

This section introduces the basic notations and definitions for the domain adaptation problem.

3.1. Domain

Domain $\mathcal{D}$ is composed of two principal elements: an m-dimensional feature space $\mathcal{X}$ and a marginal probability $P(x)$, i.e., $\mathcal{D} = \{\mathcal{X}, P(x)\}$, where $x \in \mathcal{X}$.

The input data include two domains: a source domain (S) and a target domain (T). $\mathcal{D}_s = \{(x_1, y_1), \ldots, (x_{n_s}, y_{n_s})\}$ denotes the labeled source domain with n_s samples and $\mathcal{D}_t = \{x_{n_s+1}, \ldots, x_{n_s+n_t}\}$ denotes unlabeled target data with n_t samples.

Overall, the two domains are different when they possess either different feature spaces or marginal probability distributions, i.e., $\mathcal{X}_s \neq \mathcal{X}_t$ or $P_s(x_s) \neq P_t(x_t)$ [32].

3.2. Task

Given a specific domain $\mathcal{D}$, task $\mathcal{T}$ is comprised of pairs $\{\mathcal{Y}, f(x)\}$, where $\mathcal{Y}$ is a label space and $f(x)$ is a prediction function. From a probabilistic viewpoint, $f(x)$ can be interpreted as the conditional probability distribution, i.e., $f(x) = Q(y \mid x)$, where $y \in \mathcal{Y}$.

In general, two tasks are different when they possess either different label spaces or conditional probability distributions, i.e., $\mathcal{Y}_s \neq \mathcal{Y}_t$ or $Q_s(y_s \mid x_s) \neq Q_t(y_t \mid x_t)$ [32].

For a specific labeled source domain $\mathcal{D}_s$ and unlabeled target domain $\mathcal{D}_t$, under the following assumptions: $\mathcal{X}_s = \mathcal{X}_t$, $\mathcal{Y}_s = \mathcal{Y}_t$, $P_s(x_s) \neq P_t(x_t)$ and $Q_s(y_s \mid x_s) \neq Q_t(y_t \mid x_t)$, our problem is to obtain a low-dimensional feature space in which two major criteria are satisfied:

1. Difference minimization between $P_s(x_s)$ and $P_t(x_t)$;
2. Difference minimization between $Q_s(y_s \mid x_s)$ and $Q_t(y_t \mid x_t)$.

3.3. PCA

Let $X = [x_1, \ldots, x_n] \in \mathbb{R}^{(m \times n)}$ be the input data matrix and $H = I - \frac{1}{n}\overrightarrow{1}\overrightarrow{1}^T$ be the centering matrix where $I \in \mathbb{R}^{(n \times n)}$ is the identity matrix, $\overrightarrow{1}$ is a $n \times 1$ vector of ones, and n is equal to $n_s + n_t$. XHX^T computes the covariance matrix of data. PCA attempts to learn an orthogonal transformation matrix $A \in \mathbb{R}^{m \times k}$ besides maximum variance preservation in an embedded subspace, according to the following relation:

$$\max_{A^T A = I} tr(A^T X H X^T A), \tag{1}$$

where $tr(.)$ shows the trace of matrix. The optimization

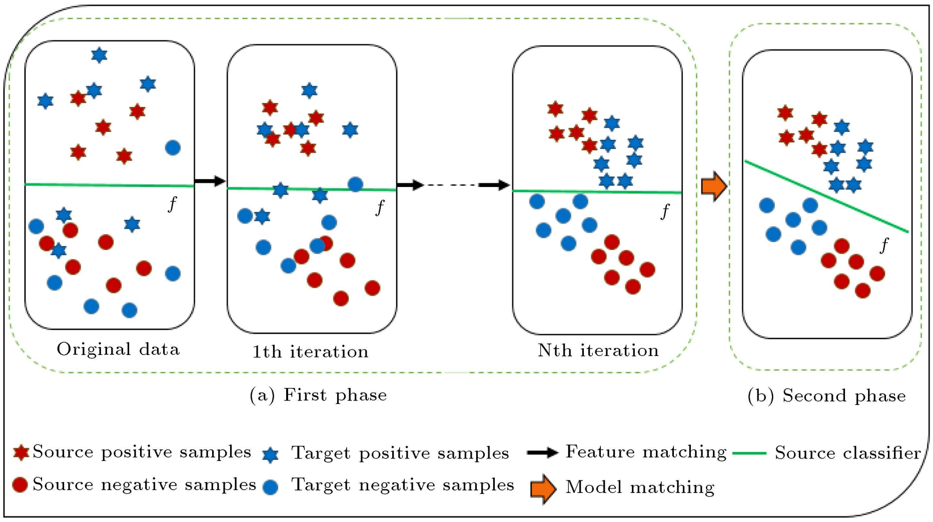


Figure 1. Illustration of our Joint Distribution Adaptation via Feature and Model Matching (JDAFMM): (a) In the first phase, JDAFMM projects the source and target samples in a shared feature space that minimizes both the marginal and conditional distribution mismatches, simultaneously and (b) in the second phase, JDAFMM constructs an adaptive classifier to match f with the manifold underlying marginal distributions.

problem can be solved via an eigenvalue decomposition of XHX^T, where matrix A is composed of top k eigenvectors corresponding to k largest eigenvalues. Then, the input data are projected in a k-dimensional representation by $Z = [z_1, \dots, z_n] = A^T X$.

3.4. MMD

There are many criteria to measure the distribution difference of various domains; however, most of them are parametric (e.g., Kallback-Leibler divergence). The parametric methods measure an intermediate density estimation, which may be a nontrivial problem. Therefore, non-parametric Maximum Mean Discrepancy (MMD) criteria are employed to measure distribution difference in Reproducing Kernel Hilbert Space (RKHS) [33]. According to MMD theory [33] the distance of mean elements in RKHS is equal to the distance of source and target domains in the original space.

4. The proposed method

This section begins with introducing the proposed approach to addressing unsupervised domain adaptation. Section 4.1 describes a joint feature- and model-based framework to learn an unsupervised domain adaptation model. Then, it presents feature- and model-based learning is presented in detail in Sections 4.2 and 4.3, respectively.

4.1. General framework

Most of the current feature-based methods for domain adaptation are aimed at obtaining a new feature representation for the source and target domains such that only the marginal distribution difference between domains is reduced. A standard classifier (e.g., K-Nearest Neighbor (KNN)) only on the labeled source data in the embedded subspace is employed, which is applied to unlabeled target data. This paper puts forward JDAFMM as a generic two-phase framework inspired by both feature- and model-based methods. Figure 1 demonstrates the main concept of our proposed approach. In the first phase, JDAFMM reduces the divergence between both the marginal and conditional distributions of the source and target domains via constructing a shared feature representation. The second phase consists in adaptive classifier learning via both labeled source and unlabeled target data. In this phase, JDAFMM adapts the prediction function with geometric data structure underlying the marginal distribution in the new feature space. In the next section, our feature- and model-based learning methods are explained in more details.

4.2. Feature matching

In this paper, JDAFMM is proposed as a particular approach to utilizing Joint Distribution Adaptation (JDA) [27] in the feature-based learning phase to find a new feature representation of data.

4.2.1. Representation learning via JDA

The main objective of dimensionality reduction methods is to find a transformed feature representation besides the reconstruction error minimization of the input data. To find a new feature representation, PCA is employed to extract the principal components.

Unfortunately, PCA is not capable of reducing the distribution difference between the source and target domains, since it assumes that the source and target data are drawn from the same probability distribution. Thus, the main issue is how to minimize the distribution difference between the source and target domains in the embedded subspace.

Marginal distribution adaptation

We seek to minimize the marginal distribution difference to adapt distribution mismatch between the source and target domains. MMD computes the distance among the instance means of domains in k-dimensional embedding:

$$Mrg(X_s, X_t) = \| \frac{1}{n_s}\sum_{i=1}^{n_s} Ax_i - \frac{1}{n_t}\sum_{j=n_s+1}^{n_s+n_t} A^T x_j \|^2$$

$$= tr(A^T X M_0 X^T A), \qquad (2)$$

where Mrg computes the distance of marginal distributions across domains and $M_0 \in \mathbb{R}^{((n_s+n_t)\times(n_s+n_t))}$ is an MMD coefficient matrix computed as follows:

$$(M_0)_{ij} = \begin{cases} \frac{1}{n_s n_s} & \text{if } x_i, x_j \in \mathcal{D}_s \\ \frac{1}{n_t n_t} & \text{if } x_i, x_j \in \mathcal{D}_t \\ \frac{-1}{n_s n_t} & \text{otherwise} \end{cases} \qquad (3)$$

Conditional distribution adaptation

To learn a model with maximum prediction accuracy, only the marginal distribution adaptation is not adequate. Thus, we resort to MMD to minimize the conditional distribution difference between domains. In this way, direct matching of conditional distribution is impossible, since the target data are completely unlabeled. That is $Q_t(y_t \mid x_t)$ cannot be estimated instantly, because the target data are completely unlabeled. To solve this problem, pseudo target labels are determined via some standard classifiers (e.g., KNN) and a trained model for the labeled samples in the source domain is employed. Since the estimation of the posterior probabilities is fully complex, class-conditional distribution as an appropriate alternative is utilized. Hence, MMD is modified to estimate the class-conditional distributions as follows:

$$Cnd(X_s, X_t) = \| \frac{1}{n_s^c}\sum_{x_i \in X_s^c} Ax_i - \frac{1}{n_t^c}\sum_{x_j \in X_t^c} A^T x_j \|^2$$

$$= tr(A^T X M_c X^T A), \qquad (4)$$

where Cnd computes the distance of class-conditional distributions across domains, and n_s^c and n_t^c are adjusted as the total numbers of instances belonging to class c in the source and target domains, respectively. In addition, D_s^c and D_t^c demonstrate the set of instances that belong to class c in the source and target domains, respectively. Moreover, M_c is the MMD coefficient matrix that contains class labels computed as follows:

$$(M_c)_{ij} = \begin{cases} \frac{1}{n_s^c n_s^c} & \text{if } x_i, x_j \in \mathcal{D}_s^c \\ \frac{1}{n_t^c n_t^c} & \text{if } x_i, x_j \in \mathcal{D}_t^c \\ \frac{-1}{n_s^c n_t^c} & \text{if } x_i \in \mathcal{D}_s^c, x_j \in \mathcal{D}_t^c \\ & \| \; x_j \in \mathcal{D}_s^c, x_i \in \mathcal{D}_t^c \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

It is noteworthy that although target labels may be imprecise due to the considerable distribution difference across domains, we suppose that the pseudo target labels may not be residing far apart from the true target labels.

Optimization problem

JDA jointly attempts to minimize differences between the marginal and conditional distributions across the source and target domains in order to learn an efficient and robust model. Thus, Eqs. (2) and (4) are integrated into Eq. (1) to form the optimization problem of JDA:

$$\min_{A^T X H X^T A = I} \sum_{c=0}^{C} tr(A^T X M_c X^T A) + \lambda \| A \|_F^2, \quad (6)$$

where λ denotes the regularization parameter, and $||.||_F$ is the Frobenius norm to ensure that the optimization problem is well defined. According to Rayleigh quotient, minimizing Eqs. (2) and (4) together with Eq. (1) is equivalent to Eqs. (2) and (4) minimized when Eq. (1) is supposed to be fixed.

Kernelization

For solving nonlinear problems, mapping function ϕ is denoted by $X \rightarrow \phi(X)$ when $\phi(X) = \{\phi(x_1), \ldots, \phi(x_n)\}$, and kernel matrix K is considered as $K = \phi(X)^T \phi(X) \in \mathbb{R}^{(n\times n)}$. Moreover, V is defined as $A = V^T \phi$ where $V \in \mathbb{R}^{((n_s+n_t)\times k)}$. Thus, according to the representation theorem, the optimization problem is redefined as follows:

$$\min_{V^T X H X^T V = I} \sum_{c=0}^{C} tr(V^T X M_c X^T V) + \lambda(\| V \|_F^2), \, (7)$$

where V is the transformation matrix for Kernel-JDA.

Learning algorithm

The Lagrange function for Eq. (6) is derived as in the following relation:

$$L = tr(A^T(X \sum_{c=0}^{C} M_c X^T + \lambda I)A)$$

$$+ \, tr((I - A^T X H X^T A)\phi), \qquad (8)$$

where $\phi = diag(\phi_1, \ldots, \phi_k) \in \mathbb{R}^{k\times k}$ is the Lagrange multiplier. Regulating $\frac{\mathrm{d}L}{\mathrm{d}A} = 0$, the generalized form of eigen decomposition is attained as follows:

$$\left(X\sum_{c=0}^{C} M_c X^T + \lambda I\right) A = XHX^T A\phi. \tag{9}$$

Finally, by solving Eq. (9), k eigenvectors corresponding to k smallest eigenvalues are chosen as transformation matrix A. JDA benefits from pseudo target labeling in an iterative manner to minimize conditional distribution mismatch. Our experiments demonstrate that the accuracy of pseudo labels in each iteration increases until convergence is reached. The refinement of target labels is done in an Expectation Maximization-like (EM) procedure.

4.3. Model matching

In the second phase, we are to learn an adaptive classifier to optimize the following two supplementary objectives: (1) structural risk minimization of prediction function in the source labeled data and (2) manifold consistency maximization underlying the marginal distributions. In the rest, the adaptive classifier and its objectives are presented in more details.

- **Learning based on structural risk minimization.** The first goal of an adaptive classifier is to find the prediction function with minimum empirical risk in the labeled source data. To achieve this goal, the empirical risk/loss function is defined as follows:

$$l(f(g(x_i)), y_i) = \sum_{i=1}^{n_s+n_t} R_{ii}(y_i - f(g(x_i)))^2, \tag{10}$$

where l is considered as the squared loss and each instance of the source domain is demonstrated as a feature vector x. Moreover, $g(.)$ indicates the mapping function to map source domain data onto the embedded feature space, f is the prediction function to determine the labels of the source data in the shared subspace, and R is a diagonal matrix computed as:

$$R_{ii} = \begin{cases} 1 & \text{if } x_i \in X_s \\ 0 & \text{otherwise} \end{cases}.$$

Indeed, by employing l, the sum squared error of the actual and predicted labels in the source domain is minimized.

- **Learning based on manifold consistency maximization.** The second goal of our adaptive classifier is to find a prediction function with maximum consistency with geometric data structure. To realize this goal, manifold assumption is utilized, which supposes that the conditional distributions of two data points x_s and x_t are similar if they are close together in the underlying geometry of the marginal distributions [34]. Thus, a prediction function is built with good performance for the target data according to the obtained knowledge from the marginal distribution.

Generally, the nearest neighbor graph is exploited to model the inherent structure of input data. In this graph, there are $n_s + n_t$ vertices with each vertex representing a data point. In addition, each data point is connected to its P nearest neighbors by edges. In order to determine the weight of each edge on the graph, the following weight function is defined:

$$W_{i,j} = e^{-\| \frac{(x_i - x_j)^2}{\delta} \|}, \tag{11}$$

where δ is considered as a normalization parameter. Then, function M_f is adjusted to learn a prediction function with maximum consistency with the manifold underlying the marginal distributions:

$$M_f(P_s, P_t) = \sum_{i,j=1}^{n_s+n_t} (f(x_i) - f(x_j))^2 W_{ij}$$

$$= \sum_{i,j=1}^{n_s+n_t} f(x_i)\overline{L}_{i,j} f(x_j), \tag{12}$$

where $\overline{L}$ specifies the normalized Laplacian matrix, and P_s and P_t represent the marginal distribution of the source and target domains, respectively. Let D be a diagonal matrix $D_{ii} = \sum_{j=1}^{n_s+n_t} W_{ij}$, measuring the weighted sum of node i with other nodes; thus, $L = D - W$ denotes the un-normalized Laplacian matrix, which measures the sum of weights of node i and other nodes except itself. Also, $\overline{L} = I - D^{-\frac{1}{2}} W D^{\frac{1}{2}}$ is defined as the normalized form of matrix L [35]. Thus, incorporating Eqs. (10) and (12) can lead to the formation of our adaptive classifier optimization problem [34]:

$$\min_{f \in F} \sum_{i=1}^{n_s} l(f(g(x_i)), y_i) + \sigma f^2 + \gamma M_f(P_s, P_t), \tag{13}$$

where σ and γ are the regularization parameters, and F is a collection of classifiers.

In order to solve Eq. (13) with kernel, kernel trick form is utilized. Thus, the prediction function is redefined as $f(g(x_i)) = w^T \varphi(g(x_i))$, where φ demonstrates the mapping function to embed feature vector x in a Hilbert space and w is the classifier parameter. Moreover, $k(g(x_i), g(x_j)) = \varphi(g(x_i))^T \varphi(g(x_j))$ is considered as the kernel function. Therefore, the prediction function is modified based on representation theorem as follows [36]:

$$f(g(x)) = \sum_{i=1}^{n_s+n_t} \alpha_i k(g(x_i), g(x)). \tag{14}$$

In addition, Eq. (13) is reformulated as:

$$\alpha = argmin_{\alpha \in R^{n_s+n_t}} \parallel (Y - \alpha^T K)R \parallel_F^2$$

$$+ tr(\gamma \alpha^T K \overline{L} K \alpha + \sigma \alpha^T K \alpha), \tag{15}$$

where K is the kernel matrix and α denotes the optimal classifier parameters. Moreover, when the derivation of the objective function in Eq. (15) is set to 0, the optimal classifier parameters are achieved as in the following relation:

$$\alpha = (\sigma I + (R + \gamma \overline{L})K)^{-1} R Y^T. \tag{16}$$

Herein, the optimal parameters (α) of an efficient and robust classifier have been achieved to find prediction function f via Eq. (14). Therefore, classifier f is constructed in a new projected subspace by employing labeled source and unlabeled target samples. Now, this classifier determines the labels of unlabeled target samples with higher accuracy in the new subspace. Algorithm 1 demonstrates the complete process of JDAFMM. In the first phase, JDAFMM projects the source and target domains in a shared low-dimensional subspace and achieves difference minimization of marginal and conditional distributions between domains. JDAFMM predicts pseudo target labels using a trained model for the source domain in the embedded subspace. Moreover, JDAFMM refines the pseudo target labels in an iterative manner in order to predict more accurate labels for target data. In the second phase, JDAFMM achieves an adaptive classifier, which facilities adapting prediction function with the manifold alongside marginal distribution adaptation.

4.4. Computational complexity

In this section, computational complexity of JDAFMM is computed. According to Algorithm 1, we consider the number of iterations to be constant (e.g., 10), i.e., $O(1)$. The computational complexity is detailed as follows: $O((n_s + n_t)^2)$ for computing MMD matrix M_0, i.e., Line 3; $O(m^2)$ for solving eigen-value decomposition, i.e., Line 5; $O(m(n_s + n_t))$ and $O(C(n_s + n_t)^2)$ for classifying and updating MMD matrix M_c, respectively, i.e., Lines 7 and 9; $O((n_s + n_t)^2)$ for computing kernel matrix K and coefficients α, i.e., Lines 13 and 14; and $O((n_s + n_t)^2)$ for construction of the adaptive classifier, i.e., Line 15. Since $m << (n_s + n_t)$, the total computational complexity of JDAFMM is considered as $O(C(n_s + n_t)^2)$.

5. Experimental setup

This section presents the evaluation data and implementation details of our proposed method.

5.1. Data description

In order to assess the performance of JDAFMM approach, we conduct a variety of experiments on three types of visual benchmark datasets. Table 1 summarizes the benchmark image datasets.

```
1:  **Input:** source and target data $X$; source domain labels $y_s$; #subspace bases $k$;
    regularization parameter $\gamma$, $\sigma$, $\lambda$
2:  **Output:** target domain labels $y_t$
3:  compute $M_0$ by Equation (3)
4:  **repeat until convergence or maximum iteration reached**
5:  decompose eigenvalues of Equation (9)
6:  $A$ = choose $k$ eigenvectors corresponding to $k$ smallest eigenvalues
7:  $f$ = train a standard classifier on projected source data $\{A^T X_s, y_s\}$
8:  $y_t$ = update pseudo target labels using standard classifier $f$
9:  $M_c$ = update coefficients by Equation (5)
10: **end repeat**
11: $\{A^T X_s, A^T X_t\}$
12: $k(x_i, x_j)$ = select a kernel function
13: $K$ = compute kernel matrix by $K_{ij} = k(x_i, x_j)$ on projected data $\{A^T X_s, A^T X_t\}$
14: $\alpha$ = compute coefficients by Equation (16)
15: $f$ = learn an adaptive classifier using $f(x) = \sum_{i=1}^{n_s+n_t} \alpha_i k(x_i, x), x \in D_t$
16: $y_t$ = return target domain labels predicted by the adaptive classifier $f$
```

Algorithm 1. Joint Distribution Adaptation via Feature and Model Matching (JDAFMM).

Table 1. Description of benchmark image datasets.

Dataset	#Type	#Examples	#Features	#Classes	Subsets
USPS	Digit	1,800	256	10	U
MNIST	Digit	2,000	256	10	M
PIE	Face	11,554	1,024	68	P1,..., P5
Office	Object	1,410	800	10	A, W, D
Caltech	Object	1,123	800	10	C

Office and Caltech [9,11,21,37] datasets are well-known benchmark sets for visual DA. The datasets are comprised of four distinct object domains: Amazon (A), DSLR (D), Webcam (W), and Caltech-256 (C). Images in Amazon domain are collected from online merchants; images in DSLR domain are captured via high-resolution cameras; images in Webcam domain are taken by low-resolution cameras; and images in Caltech-256 domain are collected from Google images. In our experiments, the common Office dataset is employed, which was released by Gong et al. [9].

The images of Office and Caltech datasets follow various distributions; however, the following 10 common classes are considered in our experiments: headphones, touring-bike, computer-monitor, computer-mouse, computer-keyboard, laptop-101, calculator, video projector, backpack, and coffee-mug. By utilizing a subdivision of images from Amazon as the codebook, the images of all domains are encoded into 800-bin histograms and standardized with z-score. In general, 12 domain adaptation experiments are designed based on four available domains by considering two different datasets as the source and target domains, i.e., $C \longrightarrow A, C \longrightarrow W, \ldots, D \longrightarrow W$.

USPS (U) and MNIST (M) domains are popular handwritten digit datasets with different distributions and statistics. USPS dataset has 7,291 training and 2007 test images of size 16 × 16 scanned from envelops of the US Postal Service. MNIST dataset has 60,000 training and 10,000 test images of size 28 × 28 scanned from mixed American Census Bureau employees and American high school students. All images of USPS and MNIST datasets are resized to 16 × 16 at a grayscale level. Thus, two domain adaptation experiments are designed, namely: $U \longrightarrow M$, and $M \longrightarrow U$.

PIE is a well-known benchmark face dataset, which contains 41,368 images of size 32 × 32 from 68 individuals. All images are captured by 13 synchronized cameras and 21 flashes with different poses, illuminations, and expressions. PIE dataset, depending on the position of images, is divided into 5 different subsets: PIE1(C05, left pose), PIE2(C07, upward pose), PIE3(C09, downward pose), PIE4(C27, frontal pose), and PIE5(C29, right pose). Thus, 20 domain adaptation experiments are designed as follows: $P1 \longrightarrow P2, P1 \longrightarrow P3, \ldots, P5 \longrightarrow P4$.

5.2. *Method evaluation*

In this section, our JDAFMM results are compared with the results of two baseline machine learning approaches (TCA [3], GFK [9], JDA [27], TJM [38], and VDA [8]). Since all the mentioned methods are presented as dimensionality reduction approaches, another model is trained for the labeled source data using NN classifier for predicting the primitive labels of the unlabeled target data. All approaches are evaluated based on their reported best results.

5.3. *Implementation details*

In order to assess the effectiveness of JDAFMM versus other approaches, classification accuracy is employed as the evaluation criterion. The number of iterations for convergence of JDAFMM is set to 10. JDAFMM approach contains four different parameters, namely λ (the regularization parameter in Eq. (13), k (the size of subspace), γ (the regularization parameter in Eq. (7), and σ (the regularization parameter of Eq. (13), of which the optimal values are reported in Table 2. Moreover, the impact of parameter setting is evaluated in the next section.

6. Experimental results and discussion

In order to assess the effectiveness of our JDAFMM approach, it is compared with six related baseline methods for benchmark visual domain adaptation datasets.

6.1. *Result evaluation*

6.1.1. *Object and digit recognition*

Table 3 presents the classification accuracy of JDAFMM and six baseline methods for object (Office+Caltech) and digit datasets. For more details, experimental results are visualized in Figure 2. The experimental results demonstrate that JDAFMM leads to considerable improvement in classification accuracy (2.19%) in comparison with the best approach, i.e., VDA, and outperforms it in 8 out of 14 experiments.

Table 2. Optimal values of JDAFMM for three visual datasets.

Dataset	k	λ	γ	σ
Office+Caltech	140	1	0.01	1
Digit	180	0.01	10	0.001
PIE	180	0.1	0.01	0.001

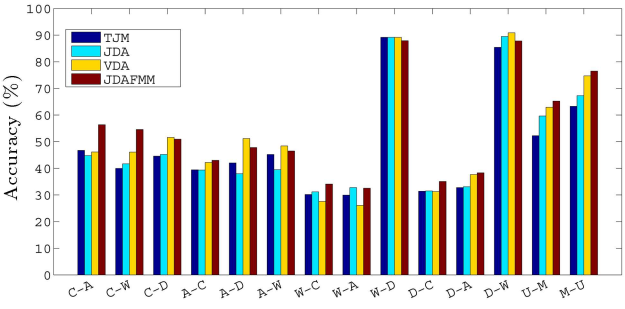


Figure 2. Classification accuracy (%) for Office+Caltech and Digits datasets. JDAFMM outperforms other dimensionality reduction and DA approaches in 8 out of 14 experiments.

Table 3. Classification accuracy (%) of the proposed method for Office+Caltech and digits datasets (our approach outperforms other dimensionality reduction and DA approaches in 8 out of 14 experiments).

Dataset	**NN**	**PCA**	**TCA**	**GFK**	**JDA**	**TJM**	**VDA**	**JDAFMM**
$C \longrightarrow A$	23.70	36.95	45.82	41.02	44.78	46.76	46.14	**56.37**
$C \longrightarrow W$	25.76	32.54	30.51	40.68	41.69	39.98	46.10	**54.58**
$C \longrightarrow D$	25.48	38.22	35.67	38.85	45.22	44.59	**51.59**	50.96
$A \longrightarrow C$	26.00	34.73	40.07	40.25	39.36	39.45	42.21	**43.01**
$A \longrightarrow W$	29.83	35.59	35.25	38.98	37.97	42.03	**51.19**	47.80
$A \longrightarrow D$	25.48	27.39	34.39	36.31	39.49	45.22	**48.41**	46.50
$W \longrightarrow C$	19.86	26.36	29.92	30.72	31.17	30.19	27.60	**34.11**
$W \longrightarrow A$	22.96	29.35	28.81	29.75	**32.78**	29.96	26.10	32.57
$W \longrightarrow D$	59.24	77.07	85.99	80.89	89.17	89.17	**89.18**	87.9
$D \longrightarrow C$	26.27	29.65	32.06	30.28	31.52	31.43	31.26	**35.08**
$D \longrightarrow A$	28.50	32.05	31.42	32.05	33.09	32.78	37.68	**38.31**
$D \longrightarrow W$	63.39	75.93	86.44	75.59	89.49	85.42	**90.85**	87.80
$U \longrightarrow M$	44.70	44.95	51.05	46.45	59.65	52.25	62.95	**65.25**
$M \longrightarrow U$	65.94	66.22	56.28	67.22	67.28	63.28	74.72	**76.50**
Average	34.79	41.93	44.93	44.55	48.04	48.76	51.86	**54.05**

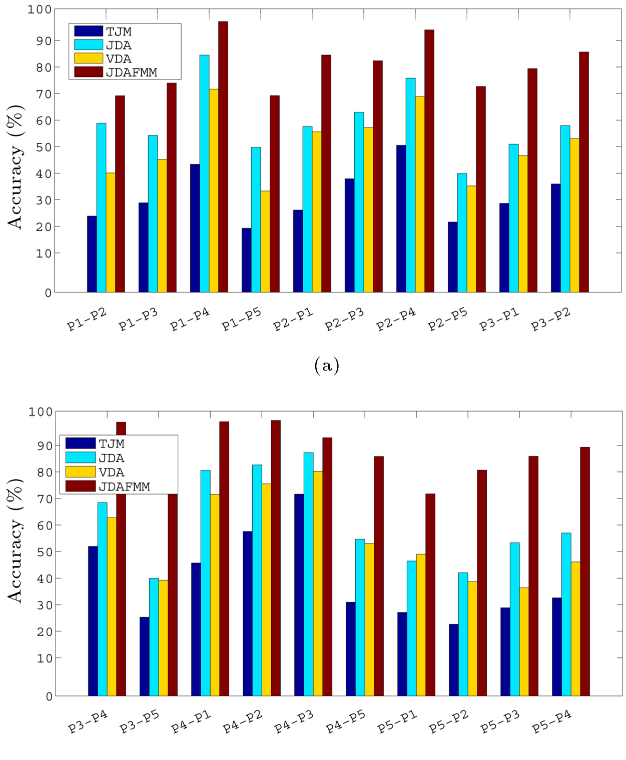
In addition, JDAFMM achieves 19.26% performance improvement in comparison with NN. This proves JDAFMM to be a promising and satisfactory solution in the case of domain shift problem.

6.1.2. *Face recognition*

Table 4 reports the classification accuracy of JDAFMM and six baseline methods for PIE dataset. For more details, the experimental results are visualized as barplots in Figure 3. From the reported results, it is observed that JDAFMM obtains significant improvement (13.07%) in classification accuracy compared with the best approach, i.e., VDA, and outperforms VDA in 19 out of 20 experiments. In the rest, we compare JDAFMM with each of the considered methods.

PCA is a significant approach in the literature on dimensionality reduction, which aims to transfer source and target data into a shared subspace alongside maximum variance preservation in the embedded subspace. Since it is supposed that the source and target samples are drawn from a similar distribution, PCA does not reach a considerably better performance than other domain adaptation methods do. The performance improvement with JDAFMM in comparison with PCA is 12.12 and 50.23 for face and object+digit datasets, respectively.

TCA is one of the domain adaptation benchmark methods that exploits transfer components to project source and target data in a new subspace. The



(a)

(b)

Figure 3. Classification accuracy (%) for PIE dataset. Our approach outperforms other dimensionality reduction and DA approaches in 19 out of 20 experiments: (a) The first 10 experiments and (b) the second 10 experiments.

Table 4. Classification accuracy (%) of the proposed method for PIE dataset (our approach outperforms other dimensionality reduction and DA approaches in 19 out of 20 experiments).

Dataset	**NN**	**PCA**	**TCA**	**GFK**	**TJM**	**JDA**	**VDA**	**JDAFMM**
$P1 \longrightarrow P2$	26.09	24.8	40.76	26.15	23.87	58.81	**72.99**	69.18
$P1 \longrightarrow P3$	26.59	25.18	41.79	27.27	28.86	54.23	61.64	**73.96**
$P1 \longrightarrow P4$	30.67	29.26	59.63	31.15	43.37	84.59	90.12	**97.12**
$P1 \longrightarrow P5$	16.67	16.3	29.35	17.59	19.3	49.75	42.40	**69.24**
$P2 \longrightarrow P1$	24.49	24.22	41.81	25.24	26.14	57.62	72.87	**84.51**
$P2 \longrightarrow P3$	46.63	45.53	51.47	47.37	37.93	62.93	75.61	**82.35**
$P2 \longrightarrow P4$	54.07	53.35	64.73	54.25	50.53	75.82	83.60	**93.99**
$P2 \longrightarrow P5$	26.53	25.43	33.7	27.08	21.63	39.89	57.72	**72.67**
$P3 \longrightarrow P1$	21.37	20.95	34.69	21.82	28.66	50.96	58.76	**79.41**
$P3 \longrightarrow P2$	41.01	40.45	47.7	43.16	35.97	57.95	74.65	**85.67**
$P3 \longrightarrow P4$	46.53	46.14	56.23	46.41	51.97	68.45	87.53	**96.31**
$P3 \longrightarrow P5$	26.23	25.31	33.15	26.78	25.31	39.95	52.63	**77.27**
$P4 \longrightarrow P1$	32.95	31.96	55.64	34.24	45.71	80.58	92.35	**96.49**
$P4 \longrightarrow P2$	62.68	60.96	67.83	62.92	57.58	82.63	92.27	**96.99**
$P4 \longrightarrow P3$	73.22	72.18	75.86	73.35	71.63	87.25	90.38	**92.89**
$P4 \longrightarrow P5$	37.19	35.11	40.26	37.38	30.94	54.66	69.98	**85.85**
$P5 \longrightarrow P1$	18.49	18.85	26.98	20.35	27.13	46.46	49.91	**71.73**
$P5 \longrightarrow P2$	24.19	23.39	29.9	24.62	22.65	42.05	62.31	**80.66**
$P5 \longrightarrow P3$	28.31	27.21	29.9	28.49	28.86	53.31	61.27	**85.91**
$P5 \longrightarrow P4$	31.24	30.34	33.64	31.33	32.59	57.01	71.19	**89.31**
Average	34.76	33.85	44.75	35.35	35.53	60.24	71.01	**84.08**

following major limitations have considerable impact on the performance of TCA: (1) It maps the source and target data in an unsupervised procedure and does not exploit label information of the source domain and (2) It only adapts the marginal distribution of the source and target domains and obviously does not reduce the conditional distribution difference across domains. JDAFMM benefits from the source domain labels to construct a new subspace and adapts the differences in both marginal and conditional distributions of the source and target domains. Improvement in performance by JDAFMM in comparison with TCA is 9.12 and 39.33 for face and object+digit datasets, respectively.

GFK learns a low-dimensional subspace by integrating an infinite number of subspaces to distinguish drifts in geometric and statistical properties of the source and target data. Due to the low-dimension of the embedded subspace, GFK represents the original data inaccurately in the embedded subspace. However, JDAFMM finds a common subspace that accurately reflects the original data. Improvement in performance by JDAFMM compared with GFK is 9.5 and 48.73 for face and object+digit datasets, respectively.

JDA, TJM, and VDA are well-known approaches that attempt to learn a common feature space by reducing distribution difference across the source and target domains. TJM suffers from the following two restrictions: (1) It needs to solve a complex optimization problem; and (2) It only adapts the marginal distribution differences between domains. JDA reduces differences of both the marginal and conditional distributions between the source and target domains; however, it does not benefit from label information of source data. In addition to reducing the mismatch between the joint marginal and conditional distributions, VDA maximizes the discrimination margin across various classes. JDAFMM outperforms JDA and VDA in 14 out of 14 and 9 out of 14 object+digit datasets, respectively, and 20 out of 20 and 19 out of 20 face datasets, respectively.

6.2. Effectiveness evaluation

A targeted series of experiments are conducted on all datasets to verify the effectiveness of JDAFMM and three baseline methods by comparing their performances in 10 iterations. TJM, JDA, VDA, and JDAFMM are repeated 10 times with their optimal parameters for Office+Caltech, Digits, and PIE datasets, and the results are illustrated in Figures 4-6. Later on, in this section, the convergence property of JDAFMM will be investigated.

Figure 4 demonstrates the average classification accuracy of JDAFMM and three baseline methods for the Office+Caltech dataset. As it is clear from the figures, TJM reduces the marginal distribution mismatch between domains via integrating feature matching and instance reweighting; however, it performs poorly in comparison with other baseline methods. JDA obtains desirable performance and outperforms TJM in 7 out of 12 experiments. VDA reduces the mismatch between joint marginal and conditional distributions in the source and target domains and employs domain invariant clustering in the embedded subspace. VDA outperforms TJM and JDA in most cases. However, JDAFMM incorporates transfer learning and domain adaptation concurrently and reduces the distribution mismatch between domains. Moreover, JDAFMM exploits an adaptive classifier in the embedded subspace to adapt source and target domains. JDAFMM outperforms VDA in 7 out of 12 experiments for Office+Caltech dataset.

It should be noticed that the classification accuracy of JDAFMM increases sharply in the 11th iteration. This is due to the use of an adaptive classifier in the embedded subspace in the second phase (iteration 11). Indeed, in the first phase, JDAFMM is repeated 10 times in order to discover a suitable shared feature representation by reducing joint marginal and conditional distributions mismatch between the domains. In most cases, JDAFMM shows similar results to those of other DA approaches; however, in the 11th iteration, JDAFMM lunges considerably because of applying the adaptive classifier. In fact, JDAFMM adapts the model along with the data in the last iteration. In this case, the model resists data drifts in source and target domains.

Figure 5 displays the performance of JDAFMM and three baseline methods for digits dataset. As it is clear from the sub-figures, JDAFMM makes remarkable improvement with digits dataset in comparison with other DA methods, particularly JDA (7.23% improvement).

Figure 6 shows the average classification accuracy of JDAFMM and three baseline methods for PIE dataset. As can be seen in the subfigures, JDAFMM performs worse than other methods in the starting steps; however, it achieves extraordinary progress from the 6th iteration onwards. Performance improvement in JDAFMM in comparison with JDA and VDA is 23.84 and 13.07, respectively.

6.3. Impact of objective function factors

In order to assess our contributions regarding the performance of JDAFMM, we conduct a serious of experiments on two benchmark datasets. Table 5 demonstrates the obtained results of JDAFMM for Office+Caltech and Digits datasets in 10 iterations.

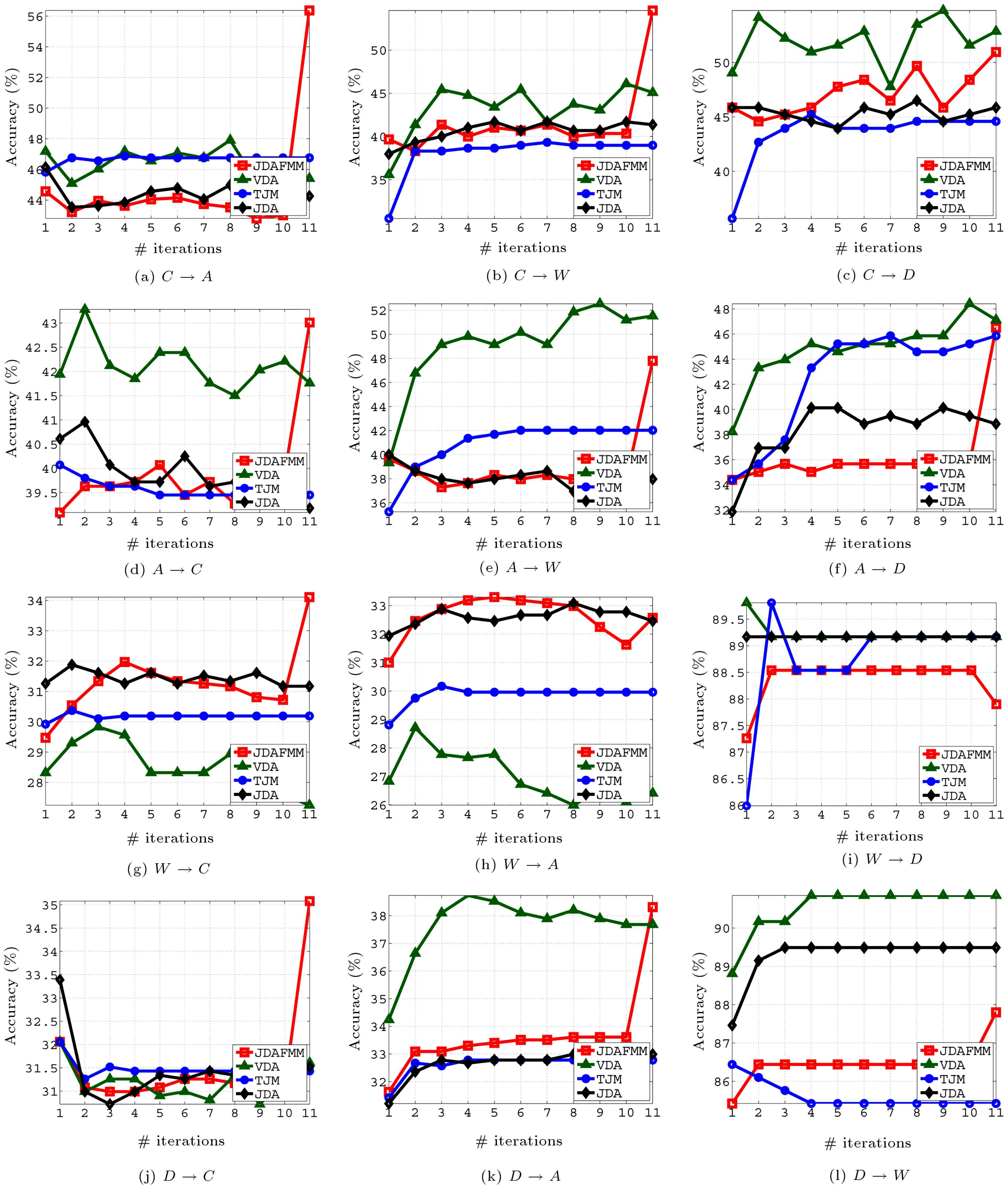
Ignoring the second phase (model matching) of the proposed approach results in 5% accuracy reduction for Office+Caltech and Digits datasets. In such situation, there are two principal reasons that propel the accuracy reduction in JDAFMM: (1) The learned model does not have minimum error for the labeled source data, and (2) The learned model does not consistent with geometric data structure. Therefore, by learning an adaptive classifier in the embedded subspace, the trained model possesses high accuracy for target samples due to the model matching with the manifold underlying the marginal distributions.

Eliminating the marginal distribution adaptation from the first phase leads to 6.54% accuracy reduction for Office+Caltech and Digits datasets. This is due to the substantial marginal distribution difference between the source and target domains. Thus, the learned model predicts the labels of target samples with low accuracy.

Ignoring the conditional distribution adaptation from the first phase yields 5.03% accuracy reduction for Office+Caltech and Digits datasets. This considerable reduction is due to the mismatch between the conditional distributions of the source and target domains. Therefore, by minimizing the conditional distribution mismatch between domains, the trained model predicts the labels of target data with high accuracy.

Table 5. Impact of objective function factors.

Dataset	JDAFMM	Ignoring model matching	Ignoring $cnd(X_s, X_t)$	Ignoring $mrg(X_s, X_t)$
Office+Caltech, Digits	**54.05**	48.04	49.02	47.51



(a) $C \rightarrow A$

(b) $C \rightarrow W$

(c) $C \rightarrow D$

(d) $A \rightarrow C$

(e) $A \rightarrow W$

(f) $A \rightarrow D$

(g) $W \rightarrow C$

(h) $W \rightarrow A$

(i) $W \rightarrow D$

(j) $D \rightarrow C$

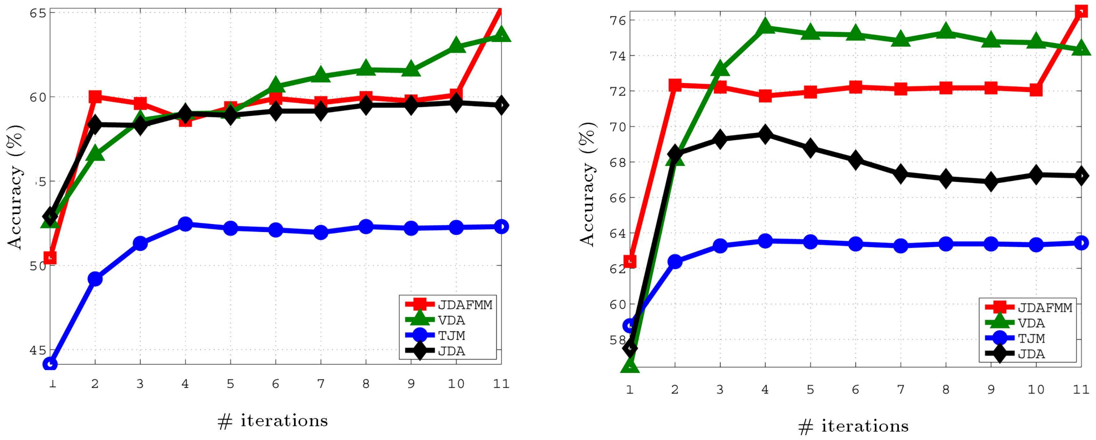
(k) $D \rightarrow A$

(l) $D \rightarrow W$

Figure 4. Classification accuracy (%) with respect to the number of iterations for Office+Caltech dataset.

6.4. Impact of parameter settings

The proposed experiments are conducted with respect to different values of parameters in order to evaluate the performance of JDAFMM in various situations. In general, four important parameters are adjusted for JDAFMM in different datasets, namely the size of subspaces, k, and the regularization parameters γ, σ, and λ. JDAFMM, VDA, and JDA are run with different values of parameters for Office+Caltech, Digits, and PIE datasets.



(a) USPS vs MNIST

(b) MNIST vs USPS

Figure 5. Classification accuracy (%) with respect to the number of iterations for digits dataset.

Figure 7 demonstrates the empirical results for parameter k for the Office+Caltech dataset. We investigate the classification accuracy of JDAFMM with $k \in [20\ 220]$ for 12 Office+Caltech datasets. The value of k characterizes the feature representation accuracy for data reconstruction. The plots show that, in most cases, JDAFMM has maximum performance with $k = 140$ for Office+Caltech dataset. Figure 8 shows the experimental results of JDAFMM, VDA, and JDA with respect to $\lambda \in [0.00001\ \ 10]$ for the Office+Caltech dataset. As it is clear from the plots, JDAFMM obtains considerable results with large values of λ. However, $\lambda = 1$ is considered for Office+Caltech dataset. In general, larger values of λ increase the importance of the regularization term. Moreover, smaller values of λ make the optimization problem ill-defined and the eigenvalue decomposition complex.

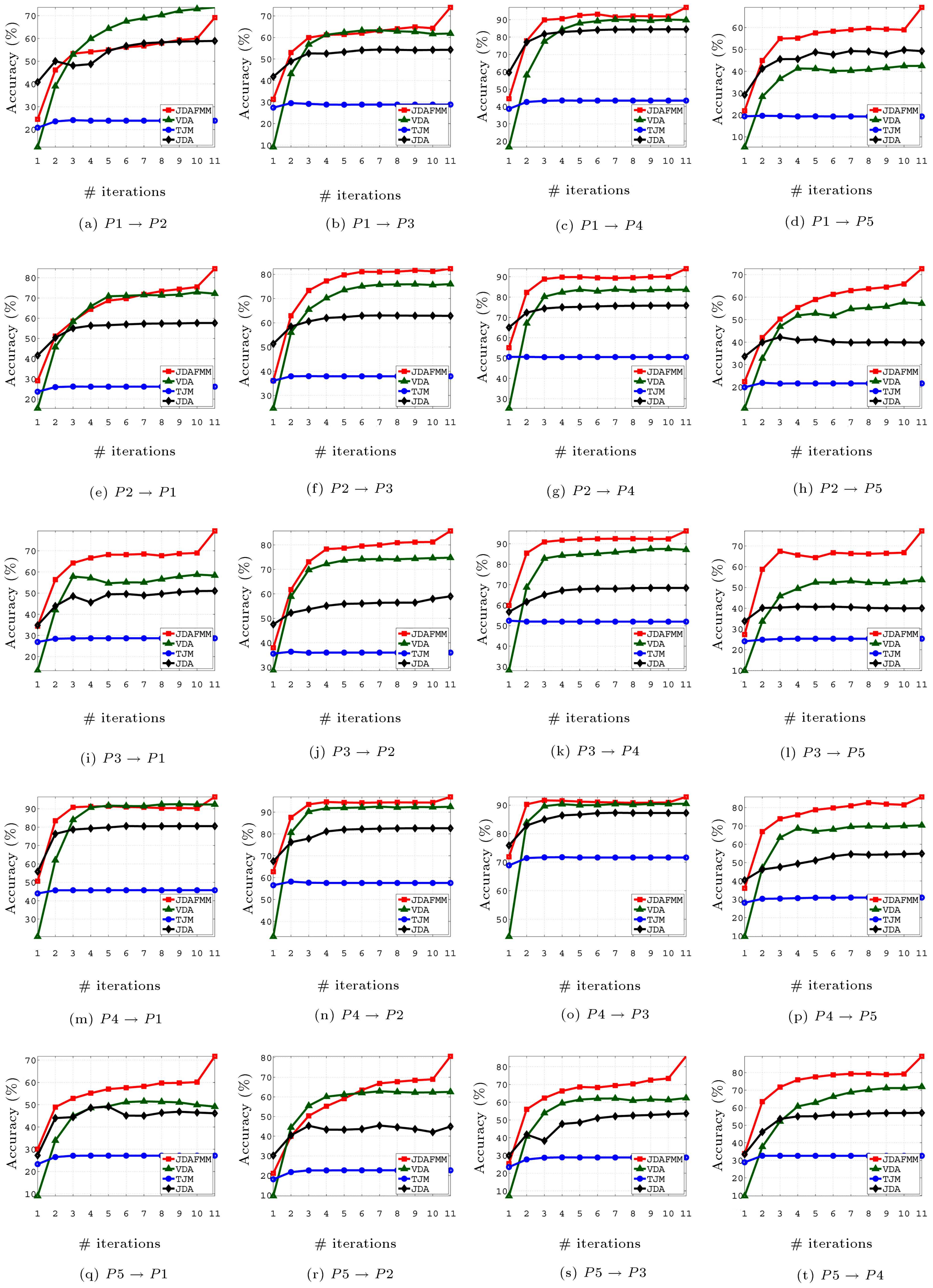
Figure 9 represents to parameter evaluation of JDAFMM regarding the classification accuracy and parameter $\gamma \in [0.0000110]$ for Office+Caltech dataset. As it is observable in the subfigures, JDAFMM performs poorly with large values of parameter γ. The value $\gamma = 0.01$ is chosen for Office+Caltech dataset. In general, large values of parameter γ neglect the label information of source domain in constructing the adaptive classifier. Moreover, small values of parameter γ neglect the extracted information from unlabeled target data.

Figure 10 reports the classification accuracy of JDAFMM for evaluating parameter $\sigma \in [0.00001\ 10]$ for the Office+Caltech dataset. As it is clear from the subfigures, in most cases, the performance of JDAFMM is degraded with larger values of σ. We choose $\sigma = 1$ for Office+Caltech dataset. In fact, larger values of σ may increase complexity of the model and decrease the effect of the other parameters in constructing the adaptive classifier.

Figure 11 illustrates the experimental results for parameter k for Digits dataset. The subfigures represent the performance of JDAFMM against other baseline methods with $k \in [20\ \ 220]$. JDAFMM obtains satisfactory results with large values of k. We set $k = 180$ for digits dataset. Figure 12 presents the parameter evaluation with respect to classification accuracy and parameter $\lambda \in [0.00001\ 10]$ for Digits dataset. The reported results illustrate that JDAFMM performs well for digits dataset with small values of λ. In other words, for large values of λ, JDAFMM cannot learn an adaptive model across the source and target domains. Thus, we set $\lambda = 0.01$ for Digits dataset.

Figure 13(a) displays the performance of JDAFMM regarding $\gamma \in [0.00001\ 10]$ for digits dataset. It is clear from Figure 13(a) and (b) that JDAFMM has an ascending order of large values of γ. As a result, $\gamma = 10$ is considered for Digits dataset. Figure 13(b) represents the experimental results of JDAFMM in analyzing the effect of parameter $\sigma \in [0.00001\ 10]$ on the accuracy of model. The figure shows that JDAFMM gains poor classification accuracy with large values of σ. We consider $\sigma = 0.001$ for Digits dataset.

Figure 14 shows the classification accuracy of JDAFMM, VDA, and JDA with respect to $k \in [20\ \ 220]$ for PIE dataset. The experimental results show that JDAFMM outperforms other DA methods with various values of k. In other words, the results indicate the effectiveness and robustness of JDAFMM in knowledge transfer from the source domain to the target one. The optimal value of k is set to 180 for PIE dataset. Figure 15 demonstrates parameter evaluation with respect to classification accuracy and parameter $\lambda \in [0.00001\ \ 10]$ for PIE dataset. As it is clear from the subfigures, in most cases, JDAFMM has better performance with $\lambda \in [0.01\ 1]$. The optimal value of λ is 0.1 for PIE dataset.



(a) $P1 \rightarrow P2$

(b) $P1 \rightarrow P3$

(c) $P1 \rightarrow P4$

(d) $P1 \rightarrow P5$

(e) $P2 \rightarrow P1$

(f) $P2 \rightarrow P3$

(g) $P2 \rightarrow P4$

(h) $P2 \rightarrow P5$

(i) $P3 \rightarrow P1$

(j) $P3 \rightarrow P2$

(k) $P3 \rightarrow P4$

(l) $P3 \rightarrow P5$

(m) $P4 \rightarrow P1$

(n) $P4 \rightarrow P2$

(o) $P4 \rightarrow P3$

(p) $P4 \rightarrow P5$

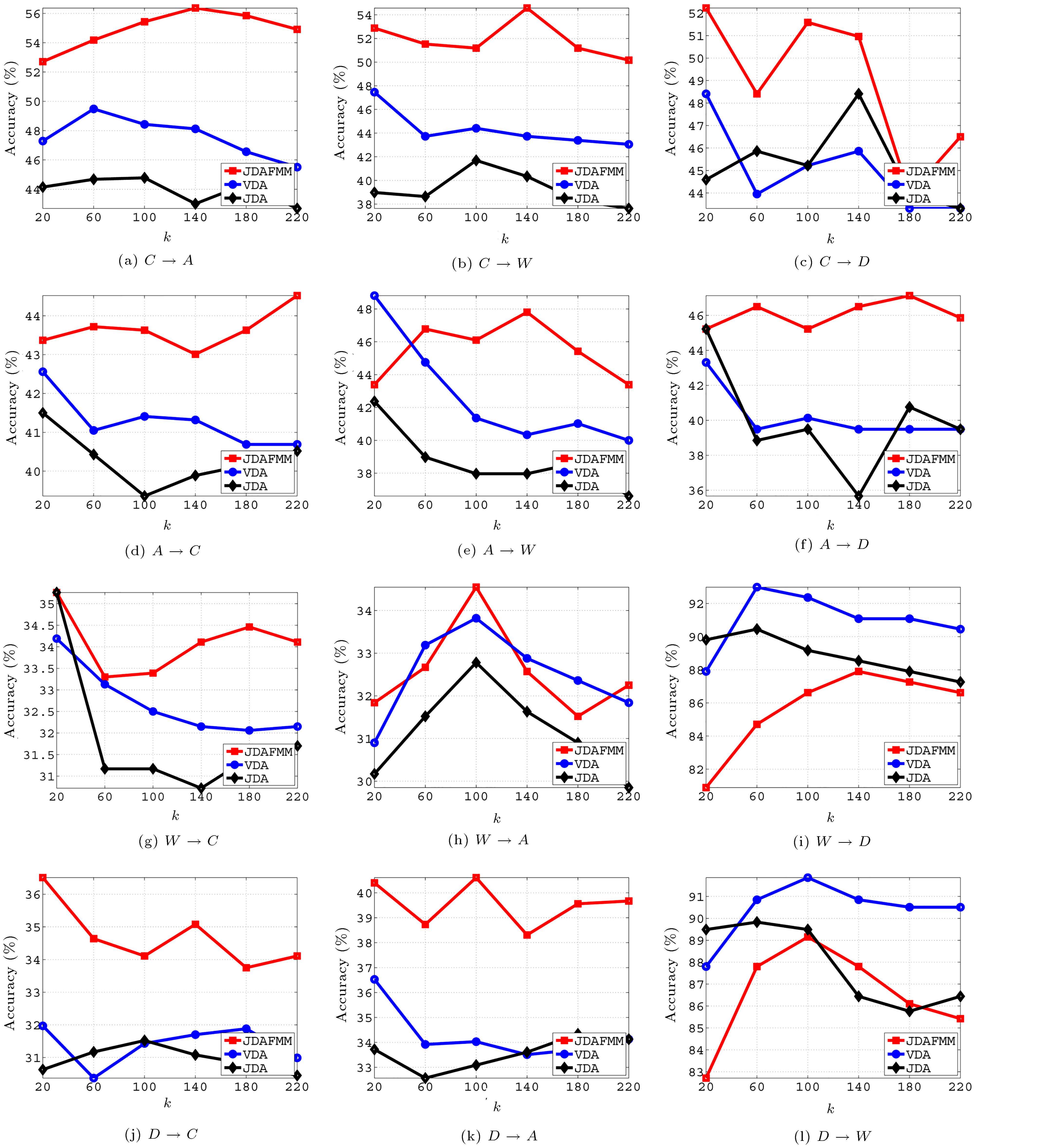
(q) $P5 \rightarrow P1$

(r) $P5 \rightarrow P2$

(s) $P5 \rightarrow P3$

(t) $P5 \rightarrow P4$

Figure 6. Classification accuracy (%) with respect to the number of iterations for PIE dataset.



(a) $C \rightarrow A$

(b) $C \rightarrow W$

(c) $C \rightarrow D$

(d) $A \rightarrow C$

(e) $A \rightarrow W$

(f) $A \rightarrow D$

(g) $W \rightarrow C$

(h) $W \rightarrow A$

(i) $W \rightarrow D$

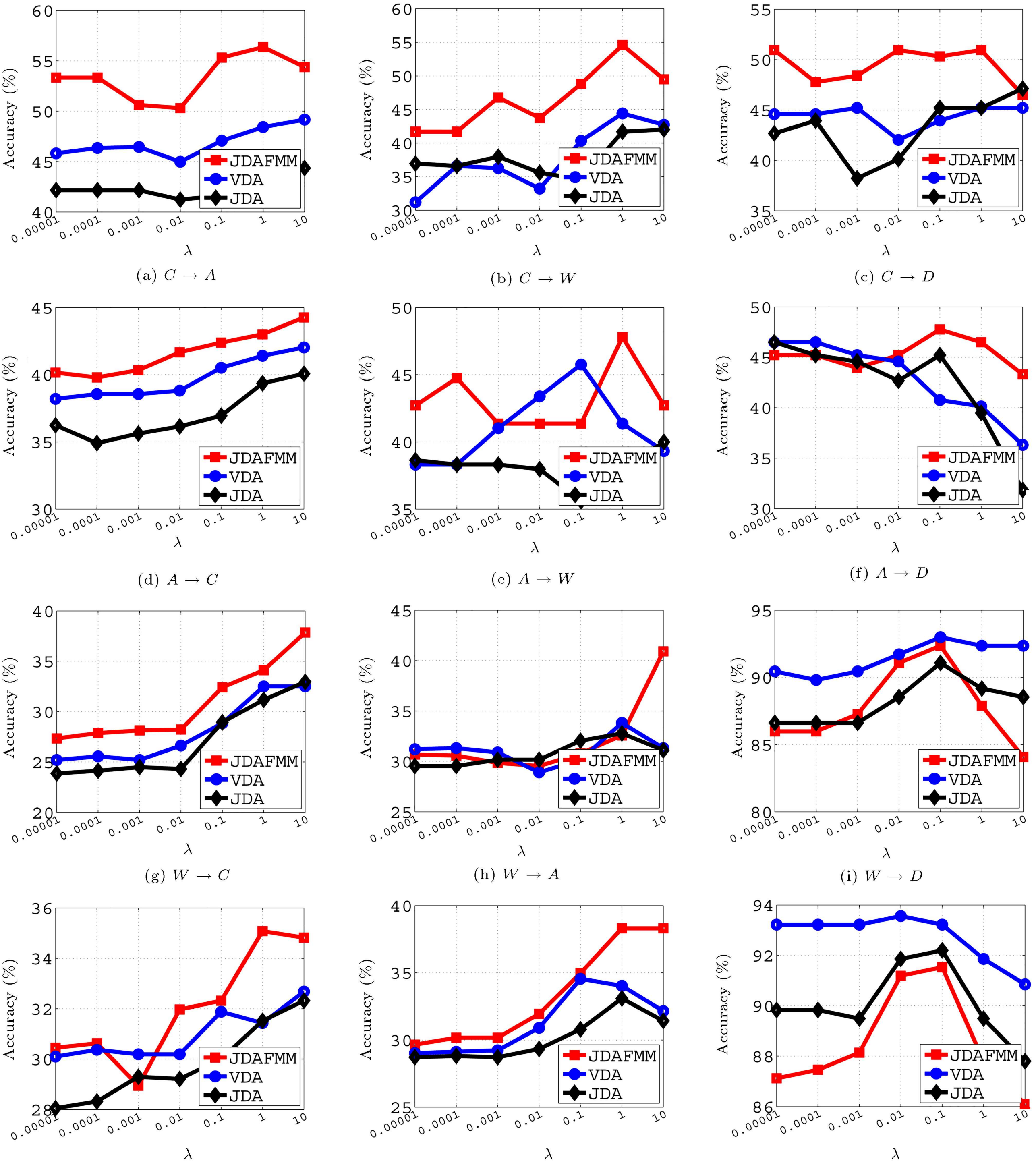
(j) $D \rightarrow C$

(k) $D \rightarrow A$

(l) $D \rightarrow W$

Figure 7. Parameter evaluation with respect to classification accuracy (%) and the number of subspace bases, k, for Office+Caltech dataset. In most cases, JDAFMM has the best performance with $k = 140$ for Office+Caltech dataset.

Figure 16 reports the experimental results of JDAFMM in evaluating parameter $\gamma \in [0.00001 \quad 10]$ for PIE dataset. It is observed in subfigures that, in most cases, JDAFMM has high classification accuracy with $\gamma \in [0.001 \quad 0.01]$. The optimal value of γ is 0.01 for PIE dataset. Figure 17 illustrates the performance of JDAFMM regarding $\sigma \in [0.00001 \ 10]$ for PIE dataset. JDAFMM demonstrates an descending order in large values of σ. Thus, we choose $\sigma = 0.001$ for PIE dataset.



(a) $C \rightarrow A$

(b) $C \rightarrow W$

(c) $C \rightarrow D$

(d) $A \rightarrow C$

(e) $A \rightarrow W$

(f) $A \rightarrow D$

(g) $W \rightarrow C$

(h) $W \rightarrow A$

(i) $W \rightarrow D$

(j) $D \rightarrow C$

(k) $D \rightarrow A$

(l) $D \rightarrow W$

Figure 8. Parameter evaluation with respect to classification accuracy (%) and parameter, λ, for Office+Caltech dataset. JDAFMM obtains considerable results with large values of λ. We consider $\lambda = 1$ for Office+Caltech dataset.

6.5. Convergence evaluation

Comprehensive experiments on Office+Caltech, digits, and PIE datasets are conducted to validate JDAFMM from a convergence standpoint and the performances of JDAFMM in comparison with VDA and JDA presented. Figures 18, 19, and 20 show the experimental results of JDAFMM, VDA, and JDA in 20 iterations for Office+Caltech, Digits, and PIE datasets, respectively; they demonstrate that there is no significant improvement after the 10th iterations. As it is clear from the

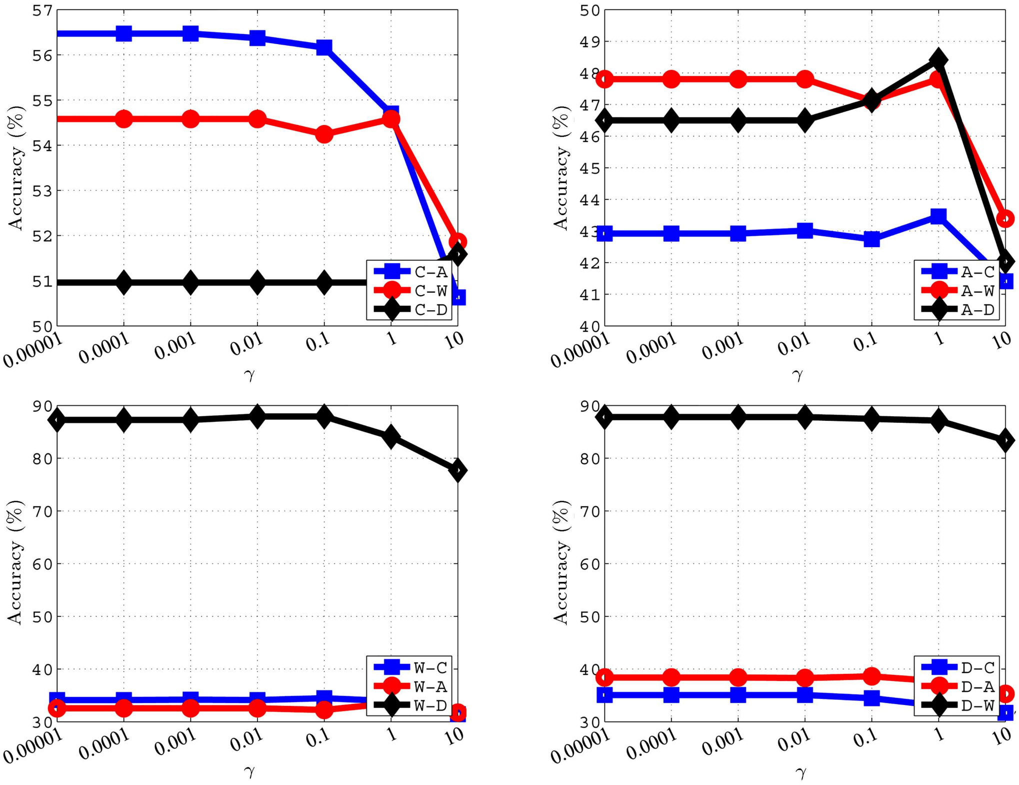


Figure 9. Parameter evaluation with respect to the classification accuracy (%) and parameter, γ, for Office+Caltech dataset. JDAFMM performs poorly with large values of parameter γ. The value $\gamma = 0.01$ is chosen for Office+Caltech dataset.

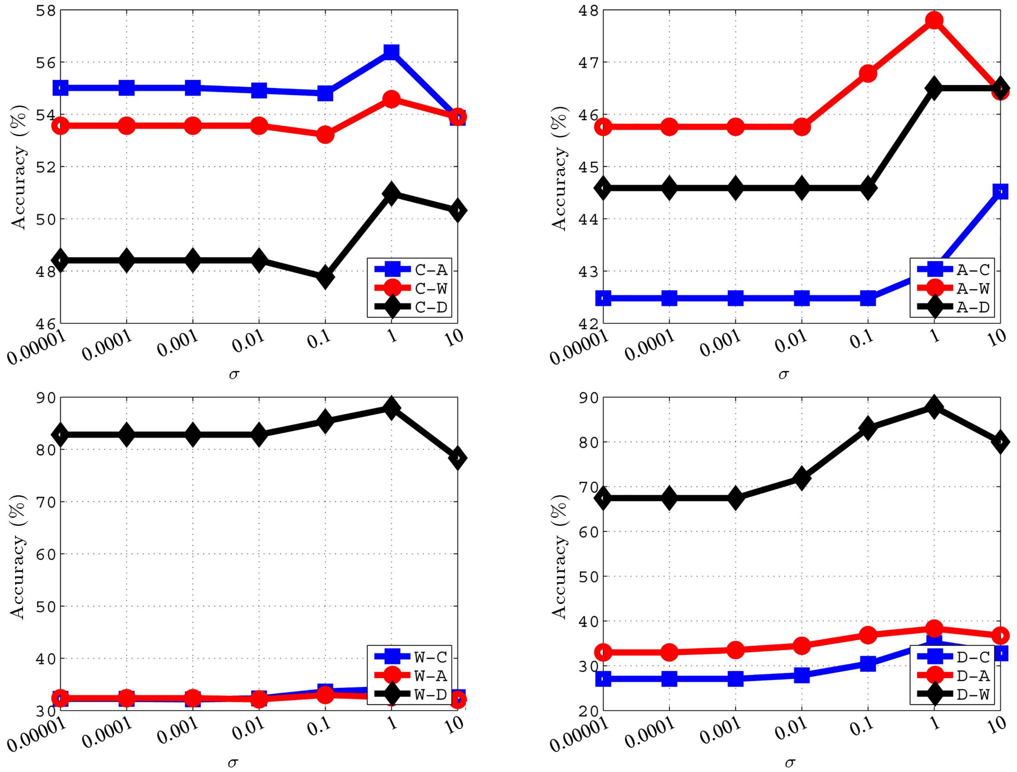
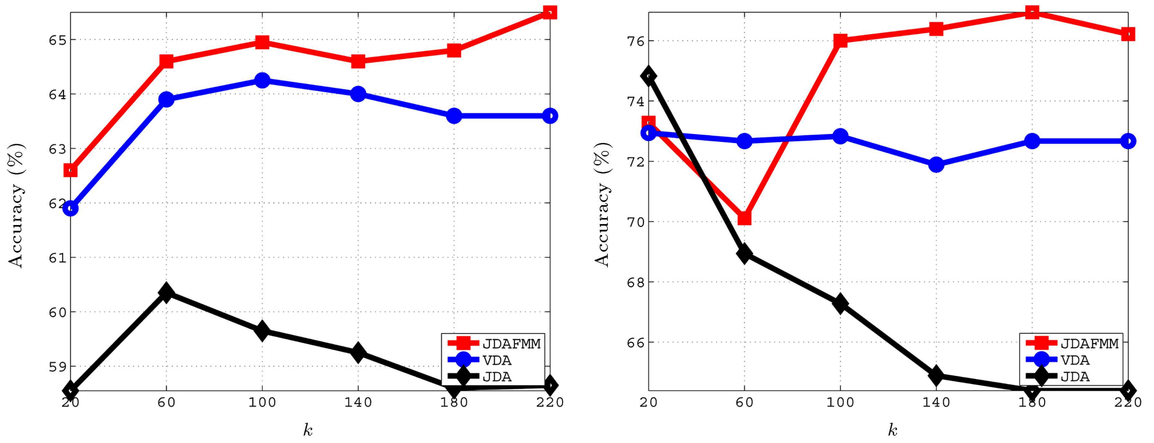


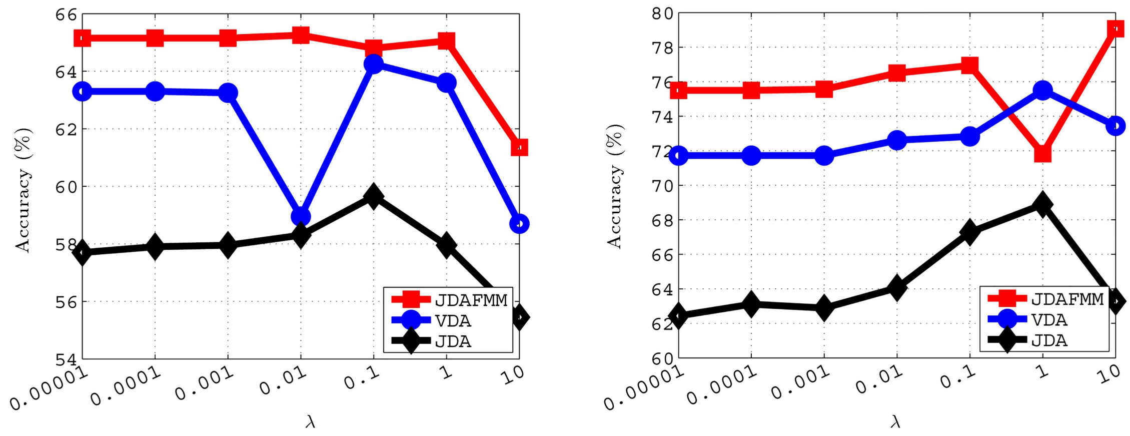
Figure 10. Parameter evaluation with respect to the classification accuracy (%) and parameter, σ, for Office+Caltech dataset. In most cases, the performance of JDAFMM is degraded with large values of σ. We choose $\sigma = 1$ for Office+Caltech dataset.



(a) USPS vs MNIST

(b) MNIST vs USPS

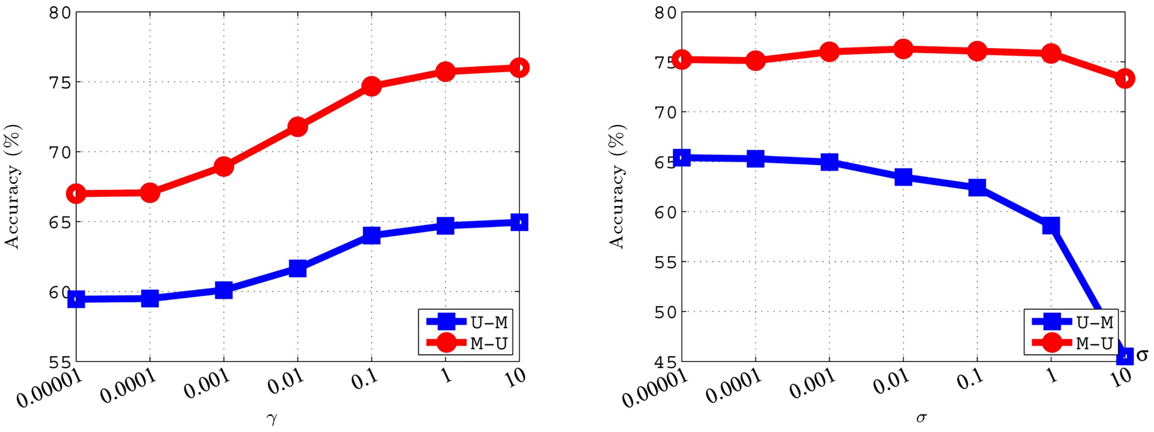
Figure 11. Parameter evaluation with respect to the classification accuracy (%) and the number of subspace bases, k, for digits dataset. JDAFMM obtains satisfactory results with large values of k. We set $k = 180$ for digits dataset.



(a) USPS vs MNIST

(b) MNIST vs USPS

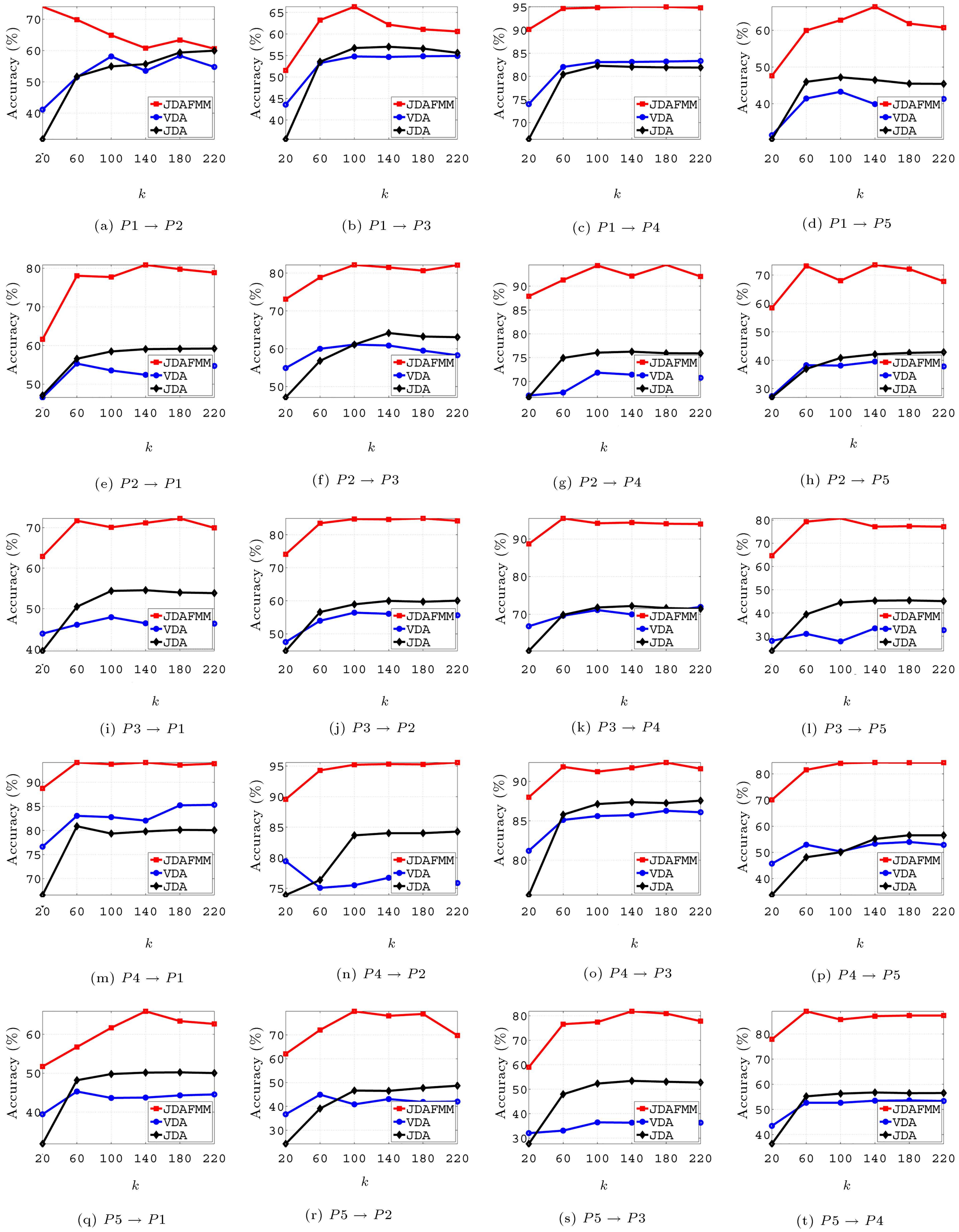
Figure 12. Parameter evaluation with respect to the classification accuracy (%) and the regularization parameter, λ, for digits dataset. JDAFMM performs well for digits dataset with small values of λ. We adjust $\lambda = 0.01$ for digits dataset.



(a)

(b)

Figure 13. Parameter evaluation with respect to the classification accuracy (%) and parameters γ and σ for digits dataset. JDAFMM shows an ascending manner with large values of γ. Moreover, JDAFMM has poor classification accuracy with large values of σ. As a result, we consider $\sigma = 0.001$ and $\gamma = 10$ for digits dataset in evaluating (a) parameter γ and (b) parameter σ.



(a) $P1 \rightarrow P2$

(b) $P1 \rightarrow P3$

(c) $P1 \rightarrow P4$

(d) $P1 \rightarrow P5$

(e) $P2 \rightarrow P1$

(f) $P2 \rightarrow P3$

(g) $P2 \rightarrow P4$

(h) $P2 \rightarrow P5$

(i) $P3 \rightarrow P1$

(j) $P3 \rightarrow P2$

(k) $P3 \rightarrow P4$

(l) $P3 \rightarrow P5$

(m) $P4 \rightarrow P1$

(n) $P4 \rightarrow P2$

(o) $P4 \rightarrow P3$

(p) $P4 \rightarrow P5$

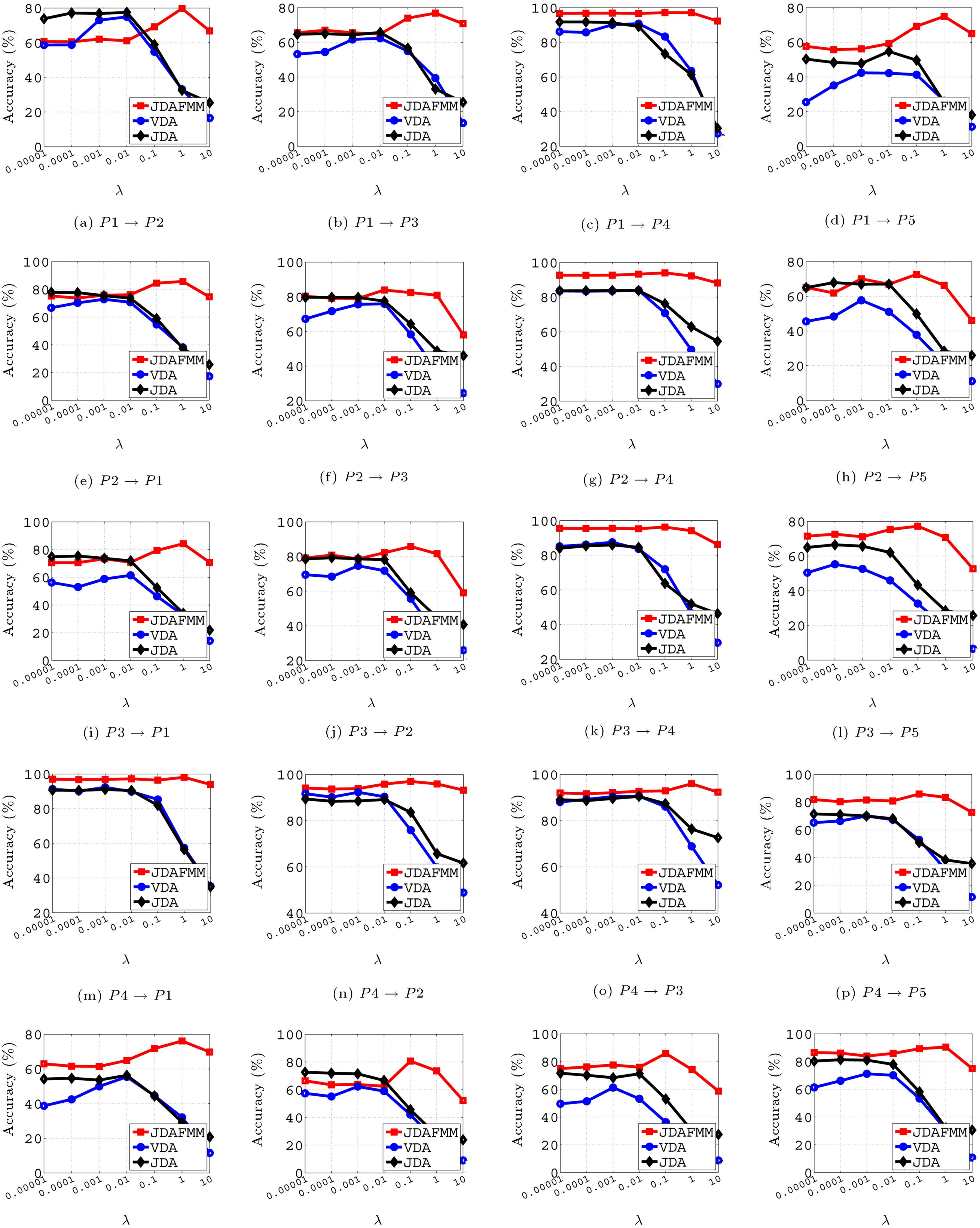
(q) $P5 \rightarrow P1$

(r) $P5 \rightarrow P2$

(s) $P5 \rightarrow P3$

(t) $P5 \rightarrow P4$

Figure 14. Parameter evaluation with respect to the classification accuracy (%) and the number of subspace bases, k, for PIE dataset. The experimental results show that JDAFMM outperforms other DA methods with varying values of k. The optimal value of k is set to 180 for PIE dataset.



(a) $P1 \rightarrow P2$

(b) $P1 \rightarrow P3$

(c) $P1 \rightarrow P4$

(d) $P1 \rightarrow P5$

(e) $P2 \rightarrow P1$

(f) $P2 \rightarrow P3$

(g) $P2 \rightarrow P4$

(h) $P2 \rightarrow P5$

(i) $P3 \rightarrow P1$

(j) $P3 \rightarrow P2$

(k) $P3 \rightarrow P4$

(l) $P3 \rightarrow P5$

(m) $P4 \rightarrow P1$

(n) $P4 \rightarrow P2$

(o) $P4 \rightarrow P3$

(p) $P4 \rightarrow P5$

(q) $P5 \rightarrow P1$

(r) $P5 \rightarrow P2$

(s) $P5 \rightarrow P3$

(t) $P5 \rightarrow P4$

Figure 15. Parameter evaluation with respect to the classification accuracy (%) and the number of subspace bases, λ, for PIE dataset. In most cases, JDAFMM has better performance with $\lambda \in [0.01\ 1]$. The optimal value of λ is 0.1 for PIE dataset.

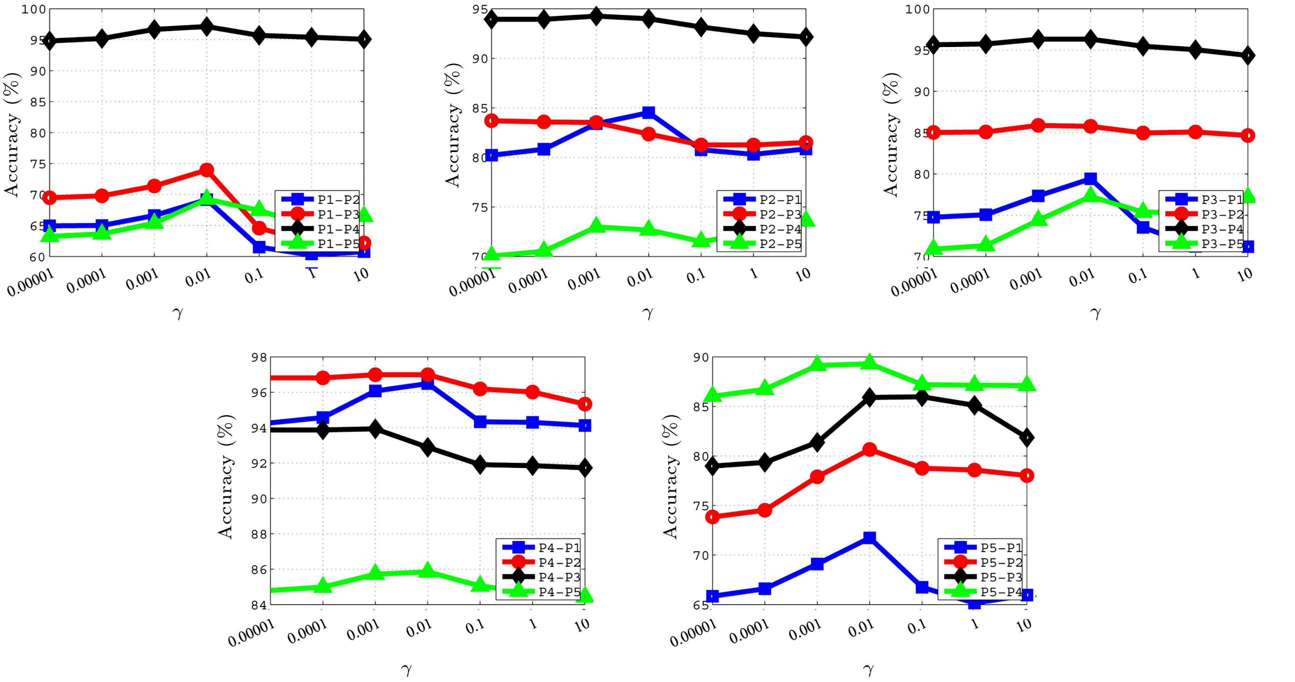


Figure 16. Parameter evaluation with respect to the classification accuracy (%) and parameter γ for PIE dataset. In most cases, JDAFMM reaches high classification accuracy with $\gamma \in [0.001\ 0.01]$. The optimal value of γ is 0.01 for PIE dataset.

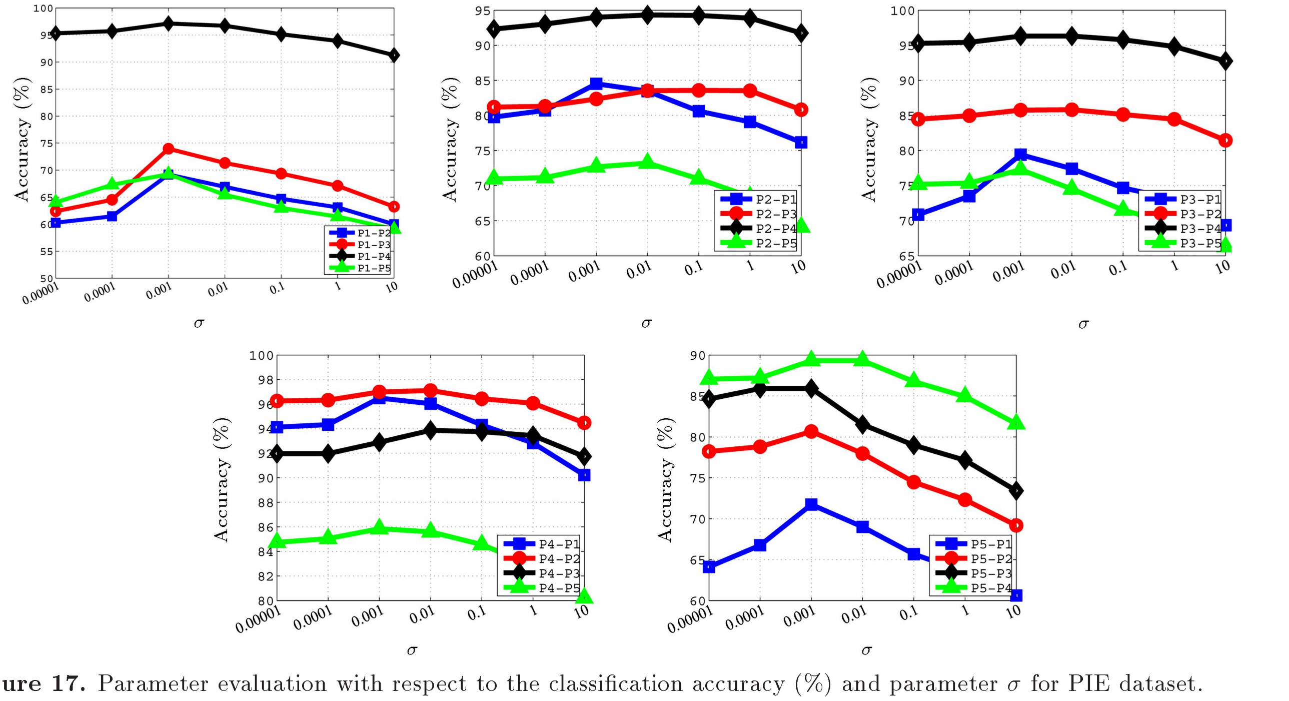
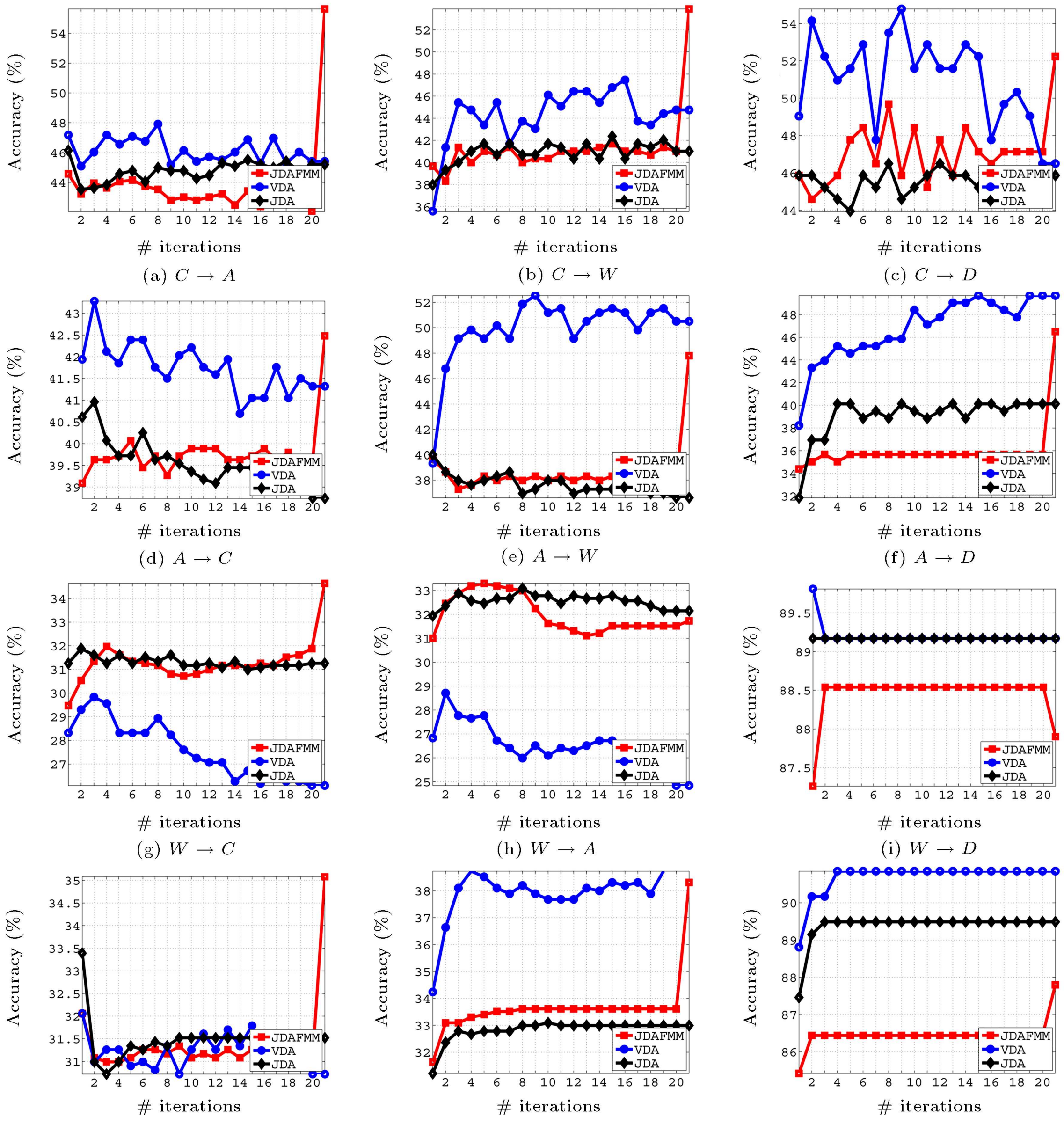


Figure 17. Parameter evaluation with respect to the classification accuracy (%) and parameter σ for PIE dataset. JDAFMM demonstrates a descending manner for large values of σ. We choose $\sigma = 0.001$ for PIE dataset.

figures, in most cases, the performance of JDAFMM gradually increases during early iterations and becomes stable after about 10 iterations. In addition, JDAFMM achieves considerable improvement whenever the adaptive classifier is applied after either the 10th or the 20th iteration.

7. Conclusion

This research introduced JDAFMM as an unsuperviseddomain adaptation approach that benefited from both model- and feature-based techniques to cope with the domain shift problem. JDAFMM was a two-



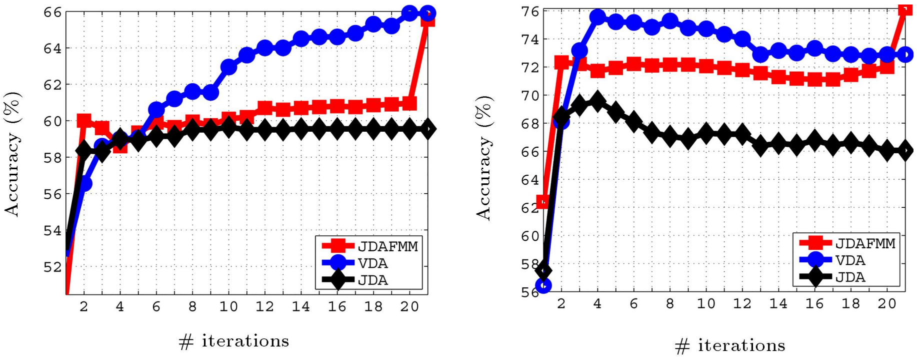
(a) $C \rightarrow A$ (b) $C \rightarrow W$ (c) $C \rightarrow D$

(d) $A \rightarrow C$ (e) $A \rightarrow W$ (f) $A \rightarrow D$

(g) $W \rightarrow C$ (h) $W \rightarrow A$ (i) $W \rightarrow D$

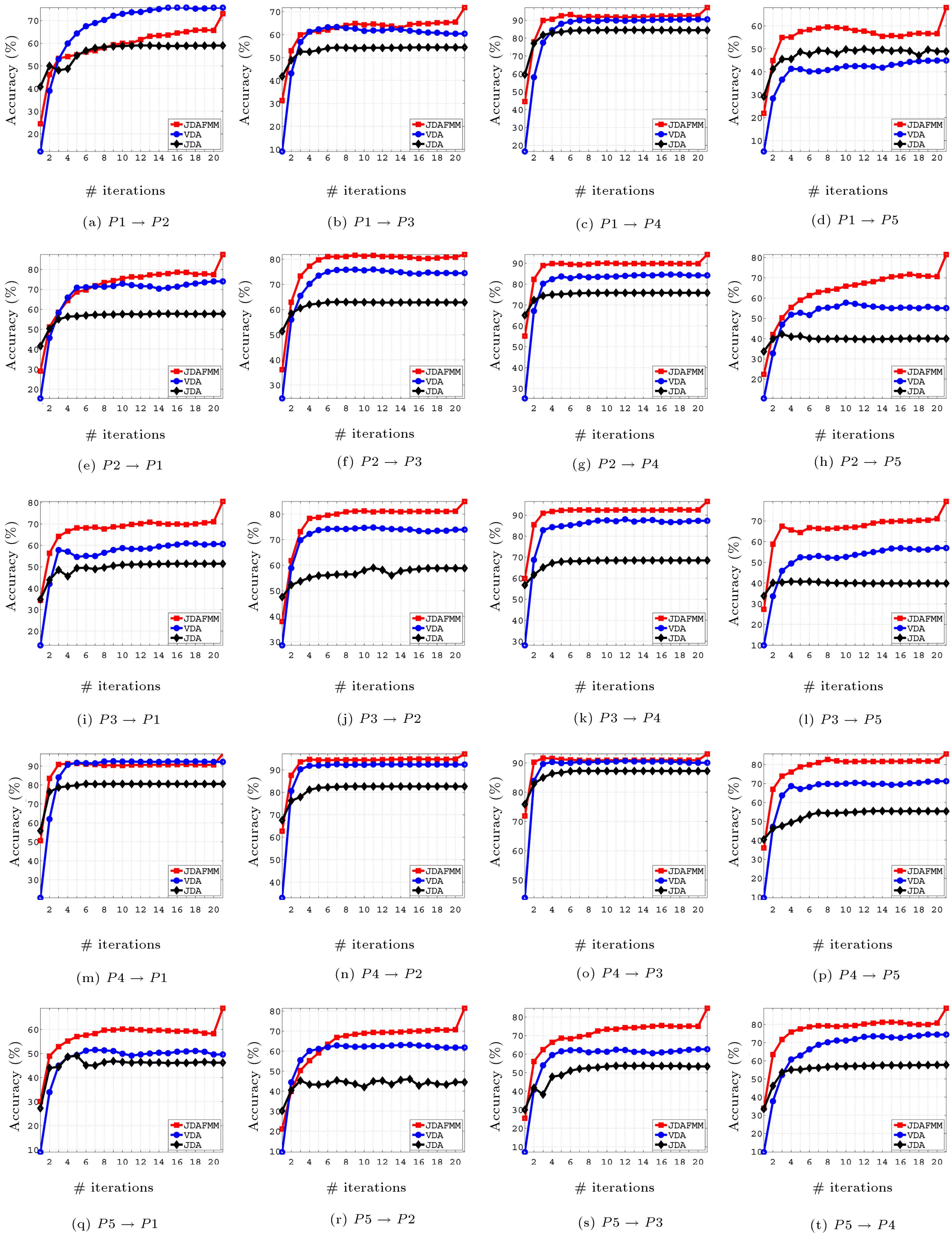
(j) $D \rightarrow C$ (k) $D \rightarrow A$ (l) $D \rightarrow W$

Figure 18. Convergence evaluation with respect to the classification accuracy (%) in 20 iterations for Office+Caltech dataset. In most cases, all three methods converge in the first 10 iterations.



(a) USPS vs MNIST (b) MNIST vs USPS

Figure 19. Convergence evaluation with respect to the classification accuracy (%) in 20 iterations for digits dataset. In most cases, all three methods converge in the first 10 iterations.



(a) $P1 \rightarrow P2$

(b) $P1 \rightarrow P3$

(c) $P1 \rightarrow P4$

(d) $P1 \rightarrow P5$

(e) $P2 \rightarrow P1$

(f) $P2 \rightarrow P3$

(g) $P2 \rightarrow P4$

(h) $P2 \rightarrow P5$

(i) $P3 \rightarrow P1$

(j) $P3 \rightarrow P2$

(k) $P3 \rightarrow P4$

(l) $P3 \rightarrow P5$

(m) $P4 \rightarrow P1$

(n) $P4 \rightarrow P2$

(o) $P4 \rightarrow P3$

(p) $P4 \rightarrow P5$

(q) $P5 \rightarrow P1$

(r) $P5 \rightarrow P2$

(s) $P5 \rightarrow P3$

(t) $P5 \rightarrow P4$

Figure 20. Convergence evaluation with respect to the classification accuracy (%) in 20 iterations for PIE dataset. In most cases, all three methods converge in the first 10 iterations.

phase solution with the following characteristics. In the first phase, JDAFMM projected the source and target data into a shared feature subspace where the differences of joint marginal and conditional distributions between domains were minimized simultaneously. In the second phase, an adaptive classifier was trained in the embedded subspace based on the joint labeled source and unlabeled target data. The goal of this adaptive classifier was to find a prediction function with minimum empirical risk for the labeled source data and maximum consistency with geometric data structure. Comprehensive experiments were conducted to validate the performance of JDAFMM from different standpoints. Our experimental results demonstrated that JDAFMM significantly outperformed other state-of-the-art domain adaptation methods for various visual benchmark datasets.

References

1. Shi, Y. and Sha, F. "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation", arXiv preprint arXiv:1206.6438 (2012).

2. Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. "Correcting sample selection bias by unlabeled data", *Neural Information Processing Systems (NIPS)*, pp. 601-608 (2007).

3. Pan, S.J., Tsang, I.W., Kwok, J.T., and Yang, Q. "Domain adaptation via transfer component analysis", *IEEE Transactions on Neural Network*, **22**(2), pp. 199-210 (2011).

4. Blitzer, J., Dredze, M., and Pereira, F. "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification", *ACL*, **7**, pp. 440-447 (2007).

5. Blitzer, J., McDonald, R., and Pereira, F. "Domain adaptation with structural correspondence learning", *Conf. on Emp. Meth. in Natu. Lang. Proc.*, Sydney, Australia, pp. 120-128 (2006).

6. Duan, L., Xu, D., Tsang, I.W., and Luo, J. "Visual event recognition in videos by learning from web data", *IEEE Trans. Pattern Anal. Mach. Intell*, **34**(9), pp. 1667-1680 (2012).

7. Jain, V. and Learned-Miller, E. "Online domain adaptation of a pre-trained cascade of classifiers", *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 577-584 (2011).

8. Tahmoresnezhad, J. and Hashemi, S. "Visual domain adaptation via transfer feature learning", *Knowledge and Information Systems (KAIS)*, **50**(2), pp. 588-605 (2017).

9. Gong, B., Shi, Y., Sha, F., and Grauman, K. "Geodesic flow kernel for unsupervised domain adaptation", *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2066-2073 (2012).

10. Tahmoresnezhad, J. and Hashemi, S. "Transductive transfer learning via maximum margin criterion", *Scientia Iranica (SCI)*, **23**(3), p. 1239 (2016).

11. Saenko, K., Kulis, B., Fritz, M., and Darrell, T. "Adapting visual category models to new domains", *11th European Conference on Computer Vision (ECCV)*, pp. 213-226 (2010).

12. Kumar, A., Saha, A., and Daume, H. "Co-regularization based semi-supervised domain adaptation", *Neural Information Processing Systems (NIPS)*, pp. 478-486 (2010).

13. Zhang, J., Li, W., and Ogunbona, P. "Joint geometrical and statistical alignment for visual domain adaptation", arXiv preprint arXiv:1705.05498 (2017).

14. Quanz, B., Huan, J., and Mishra, M. "Knowledge transfer with low-quality data: A feature extraction issue", *IEEE Transactions on Knowledge and Data Engineering*, **24**(10), pp. 1789-1802 (2012).

15. Zhong, E., Fan, W., Peng, J., Zhang, K., Ren, J., Turaga, D., and Verscheure, O. "Cross domain distribution adaptation via kernel mapping", *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1027-1036 (2009).

16. Sun, Q., Chattopadhyay, R., Panchanathan, S., and Ye, J. "A two-stage weighting framework for multi-source domain adaptation", *Neural Information Processing Systems (NIPS)*, pp. 505-513 (2011).

17. Jolliffe, I.T. "Principal component analysis and factor analysis", *PCA*, pp. 150-166 (2002).

18. Gretton, A., Borgwardt, K.M., Rasch, M., Schölkopf, B., and Smola, A.J. "A kernel method for the two-sample-problem", *Neural Information Processing Systems (NIPS)*, pp. 513-520 (2007).

19. Aljundi, R., Emonet, R., Muselet, D., and Sebban, M. "Landmarks-based kernelized subspace alignment for unsupervised domain adaptation", *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 56-63 (2015).

20. Ishii, M., and Sato, A. "Joint optimization of feature transform and instance weighting for domain adaptation", *International Joint Conference on Neural Networks (IJCNN)*, pp. 3793-3799 (2017).

21. Gong, B., Grauman, K., and Sha, F. "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation", *Int. Conf. on Mach. Learn.*, pp. 222-230 (2013).

22. Tahmoresnezhad, J. and Hashemi, S. "DiReT: An effective discriminative dimensionality reduction approach for multi-source transfer learning", *Scientia Iranica (SCI)*, **24**(3), pp. 1303-1311 (2017).

23. Liu, J., Li, J., and Lu, K. "Coupled local-global adaptation for multi-source transfer learning", *Neurocomputing*, **275**, pp. 247-254 (2018).

24. Luo, L., Wang, X., Hu, S., Wang, C., Tang, Y., and Chen, L. "Close yet distinctive domain adaptation", arXiv preprint arXiv:1704.04235 (2017).

25. Luo, L., Wang, X., Hu, S., and Chen, L. "Robust data geometric structure aligned close yet discriminative domain adaptation", arXiv preprint arXiv:1705.08620 (2017).

26. Pan, S.J., Kwok, J.T., and Yang, Q. "Transfer learning via dimensionality reduction", *23d Conf. on Artif. Intel.*, Chicago, USA, pp. 677-682 (2008).

27. Long, M., Wang, J., Ding, G., Sun, J., and Yu, P.S. "Transfer feature learning with joint distribution adaptation", *IEEE Int. Conf. on Computer Vision*, pp. 2200-2207 (2013).

28. Aytar, Y. and Zisserman, A. "Tabula rasa: Model transfer for object category detection",*Int. Conf. on Computer Vision (ICCV)*, pp. 2252-2259 (2011).

29. Gheisari, M. and Baghshah, M.S. "Joint predictive model and representation learning for visual domain adaptation", *Engineering Applications of Artif. Intel.*, **58**, pp. 157-170 (2017).

30. Yang, J., Yan, R., and Hauptmann, A.G. "Adapting SVM classifiers to data with shifted distributions", *7th IEEE Int. Conf. on Data Mining Workshops*, pp. 69-76 (2007).

31. Bruzzone, L. and Marconcini, M. "Domain adaptation problems: A DASVM classification technique and a circular validation strategy", *IEEE Trans. Pattern Anal. Mach. Intell*, **32**(5), pp. 770-787 (2010).

32. Long, M., Wang, J., Ding, G., Pan, S.J., and Yu, P.S. "Adaptation regularization: A general framework for transfer learning", *IEEE Transactions on Knowledge and Data Engineering*, **26**(5), pp. 1076-1089 (2014).

33. Borgwardt, K.M., Gretton, A., Rasch, M.J., Kriegel, H.P., Schölkopf, B., and Smola, A.J. "Integrating structured biological data by kernel maximum mean discrepancy", *Bioinformatics*, **22**(14), pp. e49-e57 (2006).

34. Belkin, M., Niyogi, P., and Sindhwani, V. "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples", *J. Mach. Learn. Res*, **7**(Nov), pp. 2399-2434 (2006).

35. Von Luxburg, U. "A tutorial on spectral clustering", *Statistics and Computing (SC)*, **17**(4), pp. 395-416 (2007).

36. Schölkopf, B., Herbrich, R., and Smola, A. "A generalized representer theorem", *Computational Learning Theory*, **11**, pp. 416-426 (2001).

37. Long, M., Wang, J., Sun, J., and Philip, S.Y. "Domain invariant transfer kernel learning", *IEEE T KNOWL DATA EN*, **27**(6), pp. 1519-1532 (2015).

38. Long, M., Wang, J., Ding, G., Sun, J., and Yu, P.S. "Transfer joint matching for unsupervised domain adaptation", *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1410-1417 (2014).

Biographies

Mehri Mardani received her BS degree in Information Technology (IT) Engineering from Foulad Institute of Technology, Isfahan, Iran, in 2015 and her MS degree in the same field from Urmia University of Technology, Urmia, Iran, in 2017.

Jafar Tahmoresnezhad received his PhD degree in Computer Science from Shiraz University, Shiraz, Iran, in 2015. He is currently an Assistant Professor in the Faculty of IT and Computer Engineering at Urmia University of Technology, Urmia, Iran. His research interests include pattern recognition, transfer learning, deep learning, data mining, and computer security.