

# Visual domain adaptation via transfer feature learning

Jafar Tahmoresnezhad<sup>1</sup> · Sattar Hashemi<sup>1</sup>

Received: 4 September 2015 / Revised: 22 March 2016 / Accepted: 8 April 2016 /  
Published online: 7 May 2016  
© Springer-Verlag London 2016

**Abstract** One of the serious challenges in computer vision and image classification is learning an accurate classifier for a new unlabeled image dataset, considering that there is no available labeled training data. Transfer learning and domain adaptation are two outstanding solutions that tackle this challenge by employing available datasets, even with significant difference in distribution and properties, and transfer the knowledge from a related domain to the target domain. The main difference between these two solutions is their primary assumption about change in marginal and conditional distributions where transfer learning emphasizes on problems with same marginal distribution and different conditional distribution, and domain adaptation deals with opposite conditions. Most prior works have exploited these two learning strategies separately for domain shift problem where training and test sets are drawn from different distributions. In this paper, we exploit joint transfer learning and domain adaptation to cope with domain shift problem in which the distribution difference is significantly large, particularly vision datasets. We therefore put forward a novel transfer learning and domain adaptation approach, referred to as visual domain adaptation (VDA). Specifically, VDA reduces the joint marginal and conditional distributions across domains in an unsupervised manner where no label is available in test set. Moreover, VDA constructs condensed domain invariant clusters in the embedding representation to separate various classes alongside the domain transfer. In this work, we employ pseudo target labels refinement to iteratively converge to final solution. Employing an iterative procedure along with a novel optimization problem creates a robust and effective representation for adaptation across domains. Extensive experiments on 16 real vision datasets with different difficulties verify that VDA can significantly outperform state-of-the-art methods in image classification problem.

**Keywords** Transfer learning · Unsupervised domain adaptation · Domain invariant clustering · Domain shift · Invariant feature representation

---

✉ Jafar Tahmoresnezhad  
tahmores@cse.shirazu.ac.ir

<sup>1</sup> CSE and IT Department, Shiraz University, Shiraz, Iran

# 1 Introduction

In traditional machine learning and image processing, it is often assumed that the training and testing datasets follow the same distribution. However, many real-world applications disregard this assumption so that data from the same classes but various domains may show different characteristics. Thus, the accuracy of machines and systems that are trained in the laboratory and deployed in the wild are significantly damaged. For example, imagine that we are to train a robot to detect objects in surroundings. Can we teach robot with indoor appliances and expect it to still recognize objects well in the outdoor environment?

Our initial impression says no. The significant difference between the indoor and outdoor objects will cripple robot and confuse it when dealing with new objects. In visual recognition applications, a number of factors such as pose, lighting, blur, and resolution create a substantial difference across source domain on which classifiers are trained and the target domain on which the classifiers are tested [1–3]. Indeed, the classifiers often behave poorly on the target domain because of significant difference across domains [4–8].

In object recognition systems, labeled images are often scarce in new domains and learning a novel model without rich labeled source domain is very complex, or in some circumstances impossible [9, 10]. However, a challenging problem in computer vision that still remains is to learn an accurate classifier for a new domain using labeled images from an old domain [11, 12]. Of the several available approaches for addressing this problem, *transfer learning (TL)* and *domain adaptation (DA)* are two outstanding strategies. The main margin between these two solutions is their assumptions about the drift condition among the training and test domains [13]. In particular, DA highlights the case where the marginal distribution of the source domain  $X_s$  and the target domain  $X_t$  has basic differences, i.e.,  $P(X_s) \neq P(X_t)$ , but the conditional distributions of labels,  $P(Y_s | X_s)$  and  $P(Y_t | X_t)$ , are similar across domains. On the other hand, TL engages the problem where the marginal distribution of the source and target domains is similar, while  $P(Y_s | X_s)$  and  $P(Y_t | X_t)$  have significant difference.

Most of the available approaches decrease the distribution difference across domains based on either marginal distribution or conditional distribution [14–16]. While in some real-world applications when the domain difference is substantially large, such as image classification, both marginal and conditional distributions vary highly across domains. Recently, some approaches have started to match both the marginal and conditional distributions using kernel density estimation [17], sample selection [18], or two-stage reweighting [19], but they need to adapt in a semi-supervised manner where the target domain contains a few labeled data. In addition, lately a joint distribution adaptation approach [20] has been proposed to extract a shared subspace between the source and target domains by simultaneously reducing the marginal and conditional distributions. Despite its success, it is important to note that knowledge transfer alone without considering separability across various classes reduces classification accuracy of model on the target domain [21].

## 1.1 Contributions

In this paper, we attempt to discover a shared feature representation by reducing the distribution difference between the source and target domains. We introduce *visual domain adaptation (VDA)*, which is a novel joint transfer learning and domain adaptation approach that simultaneously adapts both the marginal and conditional distributions. VDA proceeds

to transfer the knowledge from the source to target domain alongside the preservation of discrimination across different classes. Indeed, VDA exploits domain invariant clusters to discriminate between different classes in the shared representation. In general, VDA employs a principled dimensionality reduction that transfers knowledge across domains and discriminates various classes.

In this work, we make use a nonparametric two-sample test method [22], referred to as *maximum mean discrepancy (MMD)*, to measure the dissimilarity across the empirical distributions of the source and target domains. Moreover, *principal component analysis (PCA)* [23] is exploited to extract a shared representation that is robust for distribution difference. We conduct extensive experiments on real-world vision datasets under various difficulties in knowledge transference. Our results show a noteworthy improvement in terms of average classification accuracy, where VDA outperforms the state-of-the-art transfer learning methods on most datasets.

## 1.2 Organization of the paper

Section 2 reviews related work. Section 3 presents the proposed method. Sections 4 and 5 provide experimental details and comparisons with TL and dimensionality reduction approaches for object recognition, and the paper is concluded in Sect. 6.

## 2 Related work

Transfer learning [24,25] is one of the challenging research areas studied in recent years and has been extensively researched from various perspectives [26–33]. For example, transfer learning has been employed beside genetic programming and gradient descent to tackle shift problem in unseen data [34,35].

The existing transfer learning methods are divided into three main categories: (1) instance-based methods, (2) model-based methods and (3) feature-based methods. Instance-based methods [26,36,37] engage in reweighting or sample selection of source domain based on its discrepancy from target domain. Indeed, the main strategy of instance-based methods is to incorporate instance-dependant weights into the loss function to find an optimal model. Landmark selection [2] is one of the successful approaches that incorporates MMD to reweight the source examples, where landmarks are a subset of source domain instances that are similar to the target domain in terms of the distribution. Since some of the features may only be relevant to one specific domain, landmark selection suffers from original domain comparison. Kernel-based feature Mapping with Ensemble (KMapEnsemble) [18] is another adaptive kernel- and sample-based method that maps the marginal distribution of the source and target data into a shared space and exploits a sample selection method to reduce conditional distribution across domains. The main drawback of KMapEnsemble is the increase in entropy of labels due to data mapping into a common representation.

Model-based domain adaptation methods [38,39] find adaptive classifiers which transfer the model parameters learned by the source domain into the target domain. The main focus in this area is on the semi-supervised domain adaptation problem where support vector machine (SVM) is used to find an adaptive classifier [31,40].

Our work belongs to the feature-based category [29,41,42], which can be divided into property preservation [43,44] and distribution adaptation [32,45] subcategories. In the former,

shared representation across domains is extracted by preserving important properties such as geometric structure and statistical properties. But in the latter, the difference in either marginal distribution or conditional distribution is minimized to reduce discrepancy. Transfer component analysis (TCA) [29] is a dimensionality reduction feature-based approach that exploits MMD to measure the distance across domains. It reduces the distance between domains in reproducing kernel Hilbert space by learning transfer components. Downside to TCA is that the method does not reduce conditional distribution difference explicitly and only concentrates to reduce the marginal distribution difference across domains. Also, TCA is an unsupervised method on which the assumption is that no label available in the source and target domains; however, source data contain label and it could be exploited to transfer knowledge across domains.

Geodesic flow kernel (GFK) [32] is another dimensionality reduction approach that integrates an infinite number of subspaces in a geodesic from the source to the target subspace. The incremental changes of domains are reflected along a flow as geometric and statistical properties of domains. The main disadvantage of GFK is that the constructed subspaces do not represent the original data accurately due to selecting small dimension for smooth transit across flow. Transfer joint matching (TJM) [46] is an alternative state-of-the-art joint feature- and instance-based domain adaptation method that learns a new space in which the distance across the source and target data is minimized. TJM assigns less importance to the source instances that are irrelevant to the target data. Moreover, it exploits a kernel mapping of samples by a nonlinear transformation into a low-dimensional space. The downside to TJM is that the optimization problem is quite complex, and it uses an iterative alternative to update adaptation matrix.

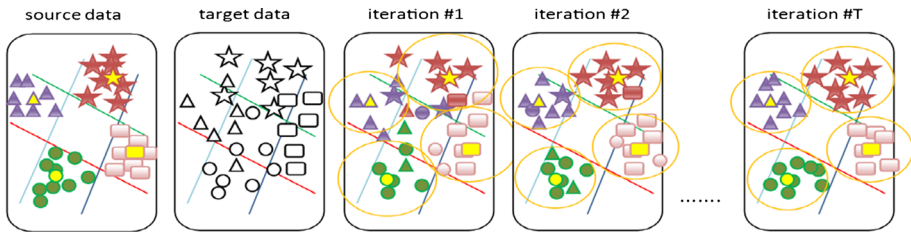
In this paper, we propose a joint marginal and conditional distribution adaptation method that exploits domain invariant clustering to discriminate between various classes. VDA transfers knowledge from the source to target domain by preserving statistical and geometric structure of domains in the shared representation. Moreover, VDA constructs compact clusters in the new representation that are domain invariant and discriminative for target data classification. Thus, VDA benefits from distance reduction and within-class scatter minimization concurrently.

### 3 Proposed method

In this section, VDA approach for effectively tackling the problem of domain shift is presented in detail.

#### 3.1 Motivation

Unlike most existing methods that solve domain shift problem with TL or DA, our methodology of addressing the problem is inspired by joint distribution adaptation (that illustrates the benefits of adapting between domains by reducing the distance between them). VDA tries to discover 'potential' clusters between the source and target domains and learns the discriminative information. Figure 1 represents the main idea of our proposed method. However, in search of the new representation, (1) we assume that we are given an  $m$ -dimensional representation of data from  $X_s$  and  $X_t$ , and (2) we learn the domain invariant representation between these two domains by preserving the geometric and statistics of their underlying space. A formal problem statement is given below.



**Fig. 1** Main idea of our kernel-based visual domain adaptation approach (Best viewed in *color*). VDA embeds source and target data into a latent space on which minimizes marginal and conditional distribution differences and clusters same label instances. VDA employs an iterative procedure to predict target data label to compare conditional distribution of domains. Surprisingly, VDA tends to converge based on increasing the amount of true labels and number of repetitions completed

### 3.2 Problem description

We begin with the definitions of domain and task and thereafter express our problem description.

**Definition 1 (Domain)** A domain  $\mathcal{D} = \{\mathcal{X}, P(x)\}$  composed of an  $m$ -dimensional feature space  $\mathcal{X}$  and a marginal probability distribution  $P(x)$ , where  $x \in \mathcal{X}$ .  $\mathcal{D}_s = \{(x_1, y_1), \dots, (x_{n_s}, y_{n_s})\}$  and  $\mathcal{D}_t = \{x_{n_s+1}, \dots, x_{n_s+n_t}\}$  are defined as labeled source and unlabeled target domains. Moreover,  $X_s$  and  $X_t$  denote the input source and target matrices.

**Definition 2 (Task)** Given domain  $\mathcal{D}$ , a task  $\mathcal{T} = \{\mathcal{Y}, f(x)\}$  is composed of label set  $\mathcal{Y}$  that pertains to  $C$  categories or classes, and a model  $f(x)$ , which can be interpreted as the conditional probability distribution, i.e.,  $f(x) = Q(y | x)$  where  $y \in \mathcal{Y}$ .

Our problem is to learn a feature representation that explicitly reduces distribution difference between joint marginal and conditional distributions, i.e.,  $P_s(x_s) \approx P_t(x_t)$  and  $Q_s(y_s | x_s) \approx Q_t(y_t | x_t)$ , respectively, where  $\mathcal{X}_s = \mathcal{X}_t$  and  $\mathcal{Y}_s = \mathcal{Y}_t$ . In fact, VDA attempts to find a new representation on which the marginal and conditional distributions are drawn from similar distributions.

### 3.3 Generating domain invariant representation

In this paper, a joint adaptation methodology is proposed that extracts a low-dimensional transformed representation which concurrently improves a classifier  $f$  based on the extracted features on refined labels. Since there are no labeled data in the target domain, i.e.,  $Q_t(y_t|x_t)$  cannot be estimated exactly, VDA seems to be a nontrivial problem. However, we exploit an EM-like (Expectation-Maximization) method to iteratively refine the adaptation matrix and classifier  $f$ . In our approximation, we estimate the  $Q_t(y_t|x_t) \approx Q_s(y_t|x_t)$  to acquire accurate target labels. In the next section, we will discuss the iterative structure of VDA in detail.

#### 3.3.1 Visual domain adaptation

**Domain adaptation** Based on the source data  $\{x_{s_i}\}$  and  $\{y_{s_i}\}$ , and the target data  $\{x_{t_i}\}$ , our task is to predict unlabeled  $\{y_{t_i}\}$  in the target domain. The general assumption in real-world domain adaptation data is that the marginal distribution of the source and target domains is very different, i.e.,  $P(X_s) \neq P(X_t)$ . Our goal is to find a low-dimensional invariant feature representation for both  $X_s$  and  $X_t$  that preserves the data properties of two domains after

adaptation. Let  $A \in \mathbb{R}^{m \times k}$  be the adaptation matrix, and  $Z_s = A^T X_s$  and  $Z_t = A^T X_t$  be the projected source and target data. The major issue is to reduce the distribution difference across domains by explicitly minimizing the empirical distance measure. Since the parametric criteria to measure distance between two distributions require expensive distribution calculation [15], we employ a nonparametric distance measure, referred to as maximum mean discrepancy (MMD). MMD computes the distance between the sample means of the source and target domains in the  $k$ -dimensional embedding:

$$D_1(X_s, X_t) = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} A^T x_i - \frac{1}{n_t} \sum_{j=n_s+1}^{n_s+n_t} A^T x_j \right\|^2 \tag{1}$$

where  $D_1$  is the distance of marginal distributions across domains, and  $n_s$  and  $n_t$  denote the number of instances in the source and target domains, respectively. It has been shown that MMD can be estimated empirically [20, 29, 46] as

$$D_1(X_s, X_t) = \operatorname{argmin}_A \operatorname{tr}(A^T X W_0 X^T A) \tag{2}$$

where  $X = \{X_s, X_t\} \in \mathbb{R}^{m \times (n_s+n_t)}$  and  $W_0 = \begin{bmatrix} (W_0)_{s,s} & (W_0)_{s,t} \\ (W_0)_{t,s} & (W_0)_{t,t} \end{bmatrix} \in \mathbb{R}^{(n+m) \times (n+m)}$  is MMD coefficient matrix where  $(W_0)_{ss}$ ,  $(W_0)_{tt}$  and  $(W_0)_{st}$  are calculated by  $\frac{1}{n_s n_s}$ ,  $\frac{1}{n_t n_t}$  and  $\frac{-1}{n_s n_t}$ , respectively. Moreover  $\operatorname{tr}(\cdot)$  denotes the trace of a matrix.

*Transfer learning* Although domain adaptation reduces the difference in marginal distribution across domains, it cannot ensure that the difference between conditional distributions is reduced as well. TL assumes that the marginal distribution of data in the source and target domains is similar, while the conditional distributions  $Q_s(y_s | x_s)$  and  $Q_t(y_t | x_t)$  are different. In this paper, we propose a robust distribution adaptation by minimizing the difference between the class-conditional distributions alongside the marginal distributions. Here, MMD is modified to measure the class-conditional distributions:

$$D_2(X_s, X_t) = \left\| \frac{1}{n_s^c} \sum_{x_i \in X_s^c} A^T x_i - \frac{1}{n_t^c} \sum_{x_j \in X_t^c} A^T x_j \right\|^2 \tag{3}$$

where  $D_2$  is the distance of class-conditional distributions between the source and target domains;  $n_s^c$  and  $n_t^c$  denote the number of examples in the source and target domains that belong to the class  $c$ , respectively. Also,  $X_s^c$  and  $X_t^c$  are defined to be the set of instances from class  $c$  belonging to the source and target data in turn. According to Eq. 2, MMD for measuring class-conditional distribution between the source and target domains can be estimated empirically as

$$D_2(X_s, X_t) = \operatorname{argmin}_A \operatorname{tr}(A^T X W_c X^T A) \tag{4}$$

where  $W_c = \begin{bmatrix} (W_c)_{s,s} & (W_c)_{s,t} \\ (W_c)_{t,s} & (W_c)_{t,t} \end{bmatrix} \in \mathbb{R}^{(n+m) \times (n+m)}$  is MMD coefficient matrix that involves the class labels, and it is computed with  $(W_c)_{ss} = \frac{1}{n_s^c n_s^c}$ ,  $(W_c)_{tt} = \frac{1}{n_t^c n_t^c}$  and  $(W_c)_{st} = \frac{-1}{n_s^c n_t^c}$ . However, the computation of  $W_c$  is nontrivial where no labeled data in the target domain are available, i.e.,  $n_t^c$  is unknown.

In this paper, we exploit the pseudo-labels of the target domain [20], which can be acquired by training a base classifier on the labeled source data to predict the unlabeled target data. The base classifier can be a standard machine learning algorithm such as nearest neighbor (NN). However, our justification is that although most of the predicted labels at first glance are inaccurate, we can still exploit them to calculate Eq. 4 to adapt conditional distribution

(as shown in Fig. 1). The iterative structure of VDA (EM-like) refines the label assignment of target domain during the learning process. Thus, the unlabeled target domain problem is resolved by leveraging the auxiliary classifier. In the experiment section, we will show how iterative pseudo-labeling aids in determining target domain labels with better accuracy.

*Domain invariant clustering* In domain shift setting, often the within-class frequencies are unequal according to significant difference among the source and target datasets; naturally the performance is challenging with respect to input data properties [6, 33]. Therefore, we are to minimize the within-class variance in any particular shifted data so that the maximal separability is guaranteed across different classes [12]. VDA endeavors to provide more separability and decision region across given classes. In fact, VDA finds a linear combination of features which jointly separates and transfers two or more classes. In this way, VDA tries to model the difference between the classes and also preserve the statistics and geometry of the original data. To this end, VDA clusters the samples with the same labels in the shared representation. Indeed, VDA minimizes the distance of each projected sample from its mean.  $S_w \in \mathbb{R}^{m \times m}$  denotes the within-class scatter matrix, which measures the distance of samples in the shared representation from their class mean. Here we formulize the optimization problem:

$$D_3(X_s, \mu^c) = \operatorname{argmin}_A (A^T S_w A) \tag{5}$$

where  $D_3$  is the distance of each instance from its mean in the projected domain;  $S_w = \sum_{\forall c \in C} \sum_{x_i \in X_s^c} (x_i - \mu_c)^T (x_i - \mu_c)$  and  $\mu_c$  denotes the mean of samples in class  $c$ . In real-world applications, the problem data  $\mu_c$  are not known and are estimated from the sample data. Hence, VDA can be sensitive to the problem data and may provide poor discrimination for some sets of problem data. This argument will be thoroughly verified in the experiments.

### 3.3.2 Optimization problem and optimal adaptation matrix

In VDA, to have an effective and robust learning, we aim to simultaneously do transfer learning and domain adaptation besides domain invariant clustering across domains. Thus, our optimization problem is comprised from Eqs. 2, 4 and 5:

$$\min_{A^T X H X^T A = I} \sum_{c=0}^C \operatorname{tr}(A^T (X W_c X^T + S_w) A) + \lambda \|A\|_F^2 \tag{6}$$

where  $\|\cdot\|_F$  is the Frobenius norm that guarantees the optimization problem to be well defined, and  $\lambda$  denotes the regularization parameter. The optimization problem achieves an adaptation matrix where it can learn a transformed feature representation by minimizing the reconstruction error. We exploit PCA for data reconstruction so that  $A^T X H X^T A$  is maximized subject to  $A^T A = I$ , where  $H = I - \frac{1}{n} \mathbf{1} \mathbf{1}^T$  denotes the centering matrix.  $I$  is considered as the identity matrix and  $\mathbf{1}$  as the ones vector. Moreover,  $X H X^T$  denotes the covariance matrix of the input data. The goal is to find an orthogonal transformation matrix  $A \in \mathbb{R}^{m \times k}$  where the variance of data in the latent space is maximized.

We derive the Lagrange function for Eq. 6 such that  $\phi = \operatorname{diag}(\phi_1, \dots, \phi_k) \in \mathbb{R}^{k \times k}$  is the Lagrange multiplier.

$$L = \operatorname{tr}(A^T (X \sum_{c=0}^C W_c X^T + S_w + \lambda I) A) + \operatorname{tr}((I - A^T X H X^T A) \phi) \tag{7}$$

Considering  $\frac{dL}{dA} = 0$ , the generalized eigen decomposition is achieved as follows:

$$\left(X \sum_{c=0}^C W_c X^T + S_w + \lambda I\right)A = X H X^T A \phi \tag{8}$$

The adaptation matrix  $A$  is obtained from solving Eq. 8 for  $k$  smallest eigenvectors. Algorithm 1 presents the complete flow of VDA. In each iteration, VDA exploits pseudo-labeling besides optimization problem (EM-like) to refine the predicted labels. In general, VDA finds the labels of target data in an iterative manner. We will verify this argument thoroughly in the experiments.

---

**Algorithm 1** Visual domain adaptation (VDA)

- 1: **Input:** source and target data  $X$ ; source domain labels  $y_s$ ; regularization parameter  $\lambda$ ; #subspace bases  $k$
  - 2: **Output:** target domain labels  $y_t$
  - 3:  $(W_0)_{ij} = \begin{cases} \frac{1}{n_s n_s} & \text{if } x_i, x_j \in \mathcal{D}_s \\ \frac{1}{n_t n_t} & \text{if } x_i, x_j \in \mathcal{D}_t \\ \frac{-1}{n_s n_t} & \text{otherwise} \end{cases}$
  - 4:  $S_w = (x_{i^*}^{s^c} - \mu^c)(x_{i^*}^{s^c} - \mu^c)^T \quad \forall i = 1 \dots n_s, c = 1 \dots C$
  - 5:  $v = \text{ones}(n, n)$ ; an  $n \times n$  matrix of ones
  - 6:  $H = I - \frac{1}{n} v v^T$
  - 7: **repeat until convergence**
  - 8: solve eigendecomposition  $(X \sum_{c=0}^C W_c X^T + S_w + \lambda I)A = X H X^T A \phi$  and select  $k$  smallest eigenvectors as adaptation matrix  $A$
  - 9: update pseudo target labels using a standard classifier  $f$  trained on projected source data  $\{A^T X_s, y_s\}$
  - 10: update  $(W_c)_{ij} = \begin{cases} \frac{1}{n_s^c n_s^c} & \text{if } x_i, x_j \in \mathcal{D}_s^c \\ \frac{1}{n_t^c n_t^c} & \text{if } x_i, x_j \in \mathcal{D}_t^c \\ \frac{-1}{n_s^c n_t^c} & \text{if } x_i \in \mathcal{D}_s^c, x_j \in \mathcal{D}_t^c \parallel x_j \in \mathcal{D}_s^c, x_i \in \mathcal{D}_t^c \\ 0 & \text{otherwise} \end{cases}$
  - 11: **end repeat**
  - 12: return target domain labels  $y_t$  determined by classifier  $f$
- 

**3.4 Computational complexity**

In this section, the computational complexity of VDA is investigated. According to Algorithm 1, the number of iterations (e.g., 10) and subspaces (e.g., 20) is considered constant, i.e.,  $O(1)$ . In this way, the computational complexity of VDA is achieved as  $O(mn + m^2 + Cn^2)$ , where  $mn$  belongs to the matrix calculation,  $m^2$  denotes the eigen decomposition and  $Cn^2$  is for the optimization problem construction. In more details, Line 3 needs  $O(n^2)$  to compute matrix  $W_0$ . Line 4 runs in  $O(n_s m)$  to compute within-class scatter matrix. Line 5 defines matrix  $v$  in  $O(n^2)$ . Line 6 computes matrix  $H$  in  $O(n^2)$ . The eigenvalue decomposition is done in  $O(m^2)$ , i.e., Line 8. The classification and update of matrix  $W_c$  need  $O(mn)$  and  $O(Cn^2)$ , respectively.

**4 Experimental setup**

In this section, we present the setup of our experiments on visual datasets for our proposed approach.



### 4.1 Data description

VDA is evaluated on three types of visual datasets that are benchmark in domain shift area (Table 1). Figure 2 shows a sample view from Office+Caltech, Digits and COIL20 datasets. Office dataset contains four domains which were studied in [2,32,33,47]: Webcam (W: low-resolution images captured by webcam), Amazon (A: images downloaded from online merchants), DSLR (D: high-resolution images captured by digital SLR camera), and Caltech-256 (C) [48]. In our experiments, the public Office dataset exploited by Gong et al. [32] is adopted to directly compare the published results.

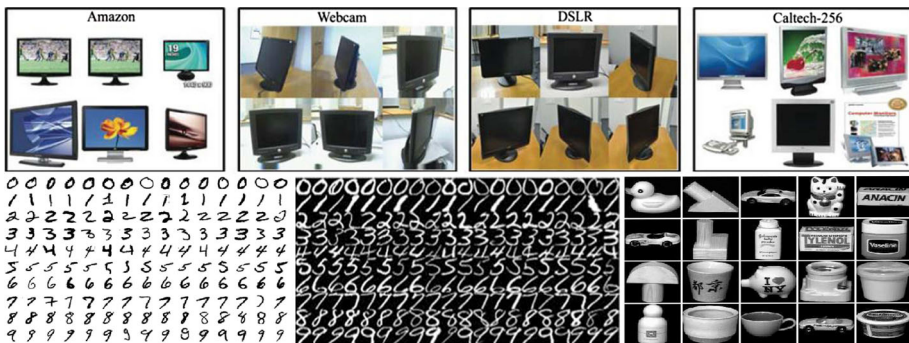
10 common classes among all office domains were employed, i.e., *calculator*, *laptop-101*, *computer-keyboard*, *computer-mouse*, *computer-monitor*, *video-projector*, *head-phones*, *backpack*, *coffee-mug*, and *touring-bike*. The SURF features [49] are exploited by considering 800-bin histograms from the trained codebooks on Amazon images. The histograms are standardized by *z*-score. Finally, two different domains are selected as the source and target domains, i.e.,  $C \rightarrow A$ ,  $C \rightarrow W, \dots, D \rightarrow W$ . Thus, the number of cross-domain office datasets will be 12.

COIL20 is a database of 20 grayscale objects with 1440 images [50]. Each object was placed in the center of a turntable stable configuration with a black background. The 72 images were taken per object through 360 degrees rotation (one for every 5 degrees of rotation). A rectangular bounding box is used to clip out from the background. The achieved box is resized to  $32 \times 32$  pixels, and then, the size of images is normalized.

We follow similar experiment protocols used in previous works to compare results directly by the reported experiments. In general, COIL20 dataset is divided into two different subsets

**Table 1** Five benchmark datasets for evaluating domain shift problem in computer vision and image processing domains

Dataset	#Instances	#Features	#Classes	Data	Domains
Office	1410	800	10	A, W, D	Object
Caltech	1123	800	10	C	Object
USPS	1800	256	10	USPS	Digit
MNIST	2000	256	10	MNIST	Digit
COIL20	1440	1024	20	COIL1, COIL2	Object



**Fig. 2** The first row illustrates the Office+Caltech-256 datasets, and the second row shows MNIST, USPS and COIL20 datasets (from left to right)

**Table 2** Recognition accuracies on 16 pairs of source/target domains

Dataset	NN	PCA	TCA	GFK	JDA	TJM	VDA
$C \rightarrow A$	23.70	36.95	45.82	41.02	44.78	<b>46.76</b>	46.14
$C \rightarrow W$	25.76	32.54	30.51	40.68	41.69	39.98	<b>46.10</b>
$C \rightarrow D$	25.48	38.22	35.67	38.85	45.22	44.59	<b>51.59</b>
$A \rightarrow C$	26.00	34.73	40.07	40.25	39.36	39.45	<b>42.21</b>
$A \rightarrow W$	29.83	35.59	35.25	38.98	37.97	42.03	<b>51.19</b>
$A \rightarrow D$	25.48	27.39	34.39	36.31	39.49	45.22	<b>48.41</b>
$W \rightarrow C$	19.86	26.36	29.92	30.72	31.17	<b>30.19</b>	27.60
$W \rightarrow A$	22.96	29.35	28.81	29.75	<b>32.78</b>	29.96	26.10
$W \rightarrow D$	59.24	77.07	85.99	80.89	89.17	89.17	<b>89.18</b>
$D \rightarrow C$	26.27	29.65	32.06	30.28	<b>31.52</b>	31.43	31.26
$D \rightarrow A$	28.50	32.05	31.42	32.05	33.09	32.78	<b>37.68</b>
$D \rightarrow W$	63.39	75.93	86.44	75.59	89.49	85.42	<b>90.85</b>
COIL1 versus COIL2	83.61	84.72	88.47	72.50	89.31	91.67	<b>99.31</b>
COIL2 versus COIL1	82.78	84.03	85.83	74.17	88.47	91.53	<b>97.92</b>
USPS versus MNIST	44.70	44.95	51.05	46.45	59.65	52.25	<b>62.95</b>
MNIST versus USPS	65.94	66.22	56.28	67.22	67.28	63.28	<b>74.72</b>
Average	40.84	47.23	49.87	48.48	53.77	53.48	<b>57.70</b>

In most cases (12 out of 16), VDA outperforms other dimensionality reduction and domain adaptation approaches

Bold values indicate the best results

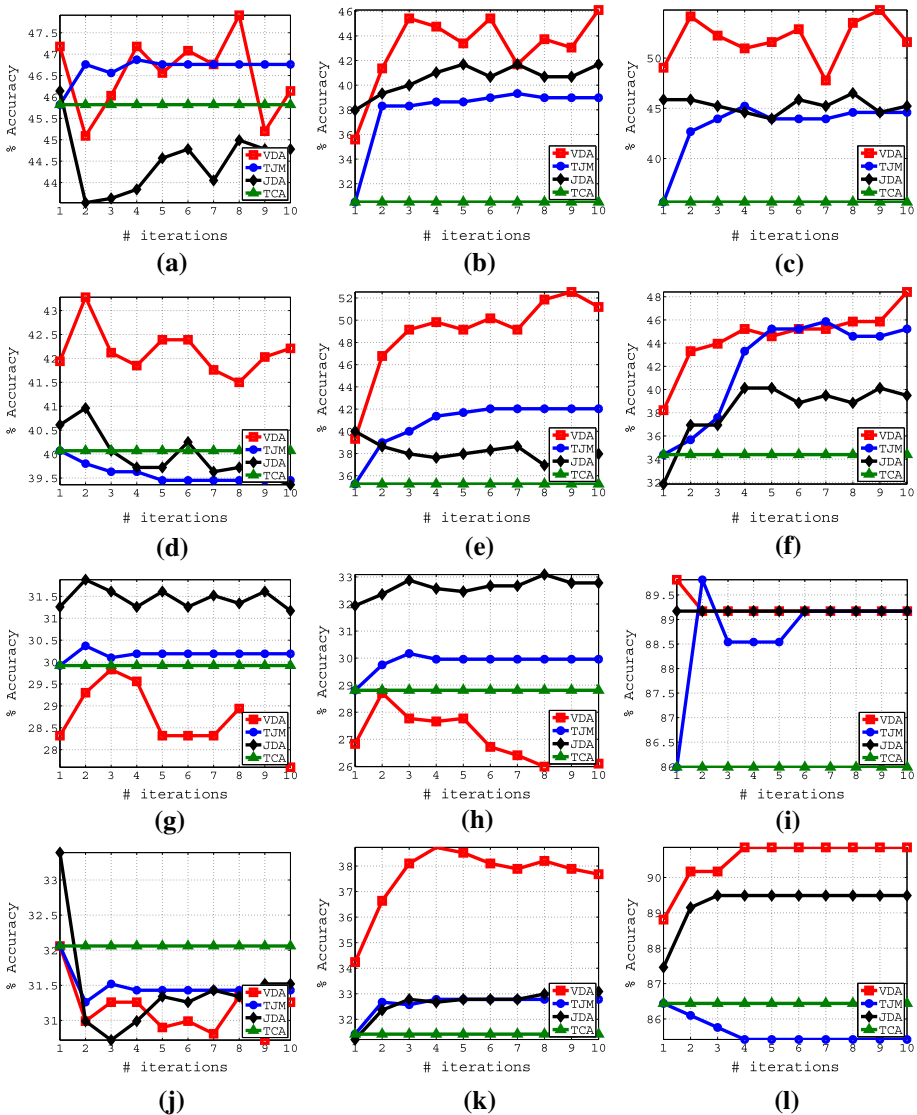
with the same number of images (720 images in each subset): COIL1 and COIL2. COIL1 contains all images taken in directions  $[0^\circ, 85^\circ]$  and  $[180^\circ, 265^\circ]$ ; COIL2 contains all images taken in directions  $[90^\circ, 175^\circ]$  and  $[270^\circ, 355^\circ]$ . We conduct two experiments with COIL1 and COIL2 as the source and target domains, respectively, and vice versa.

Digits dataset contains USPS and MNIST handwritten digits. USPS dataset refers to the handwritten digits scanned from envelopes of the US Postal Service. The number of training and test samples is 7291 and 2007, respectively, and the objects have been normalized in  $16 \times 16$  grayscale images. MNIST is a large dataset of handwritten digits that were taken from mixed American Census Bureau employees and American high school students. The images were normalized to fit into  $20 \times 20$  pixel bounding box with grayscale level. MNIST has a training set of 60000 samples and a test set with 10000 samples. USPS and MNIST share 10 classes of digits, but they have been drawn from very different distributions.

To speed up tests and have similar conditions with respect to published works, the experiments follow from [20]. We uniformly rescale all images in USPS and MNIST to size  $16 \times 16$  and fit them into grayscale pixel feature vectors. Thus, the feature space of the source and target data is unified. Also, we randomly select 1800 images from USPS as source data, and 2000 images from MNIST as target data. Next, source and target data are switched to form another dataset.

## 4.2 Method evaluation

We compare our VDA results with two machine learning baseline methods (NN and PCA) and four state-of-the-art domain adaptation approaches (TCA [29], GFK [32], JDA [20] and TJM

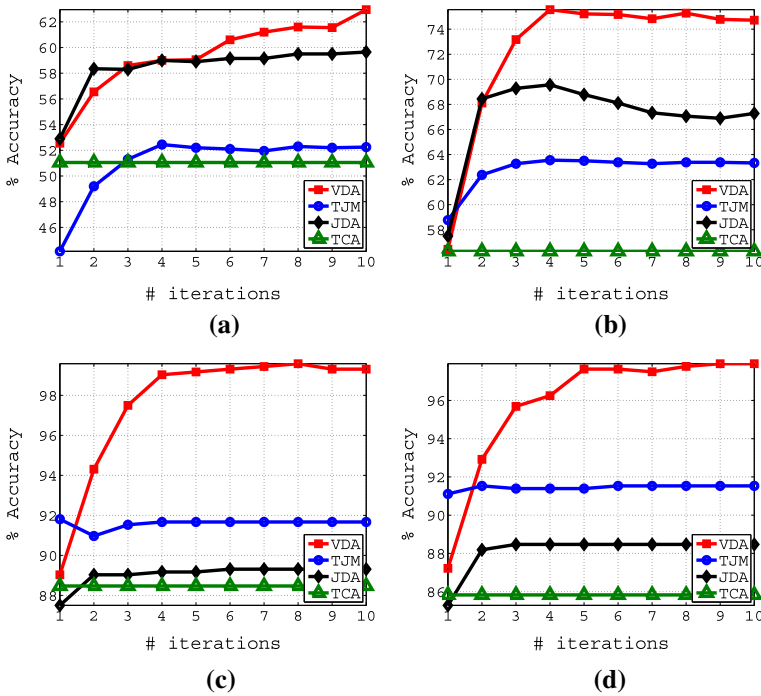


**Fig. 3** Classification accuracy (%) with respect to the number of iterations for Office+Caltech datasets. In each iteration, labels of target data are determined by an auxiliary classifier trained on the projected source data. **a**  $C \rightarrow A$ , **b**  $C \rightarrow W$ , **c**  $D \rightarrow D$ , **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**  $D \rightarrow W$

[46]). Since PCA, VDA and other domain adaptation methods are dimensionality reduction approaches, NN classifier is trained on the labeled source data for classifying unlabeled target data. All methods are evaluated by their reported best results.

### 4.3 Implementation details

The performance of VDA against other methods is evaluated by classification accuracy, which is widely used in most DA and TL methods. Moreover, since there is no random initialization,



**Fig. 4** Classification accuracy (%) with respect to the number of iterations for Digits and COIL20 datasets. The performance of VDA shows growing improvement with increasing the number of iterations. **a** USPS versus MNIST, **b** MNIST versus USPS, **c** COIL1 versus COIL2, **d** COIL2 versus COIL1

VDA does not run repeatedly. The number of iterations to VDA convergence is set to 10. We set  $k = 20$ , number of subspaces, for Office+Caltech and COIL20 datasets and  $k = 120$  for Digits datasets. Also, we consider  $\lambda = 0.05$  for Office+Caltech datasets,  $\lambda = 1.0$  for Digits datasets and  $\lambda = 0.001$  for COIL20 datasets. In the next section, the parameter settings will be discussed.

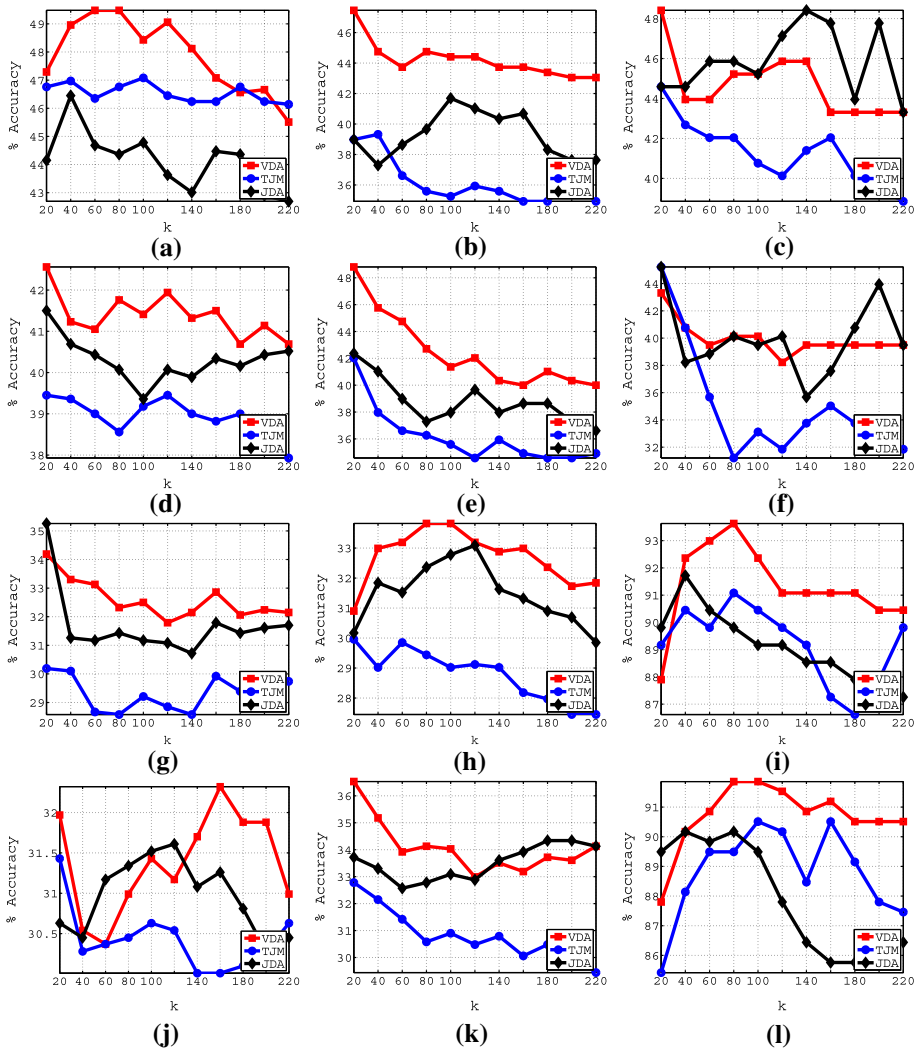
## 5 Experimental results and discussion

In this section, the performance of VDA and six baseline methods on visual benchmark domain adaptation datasets are compared.

### 5.1 Results evaluation

In Table 2, we illustrate the classification accuracy of VDA and six baseline methods on 16 visual datasets. Our results show a significant improvement in classification accuracy (3.93 %) where VDA outperforms the state-of-the-art adaptation methods on most of the datasets (12 out of 16). It is worth noting that VDA has (16.86 %) improvement compared to NN, which illustrates the adaptation difficulty in the examined datasets. As is clear from the results, VDA adapts robustly and effectively across different domains.

PCA is an effective dimensionality reduction approach that induces a  $k$ -dimensional representation. Although PCA tries to find a shared representation across domains, the distribution

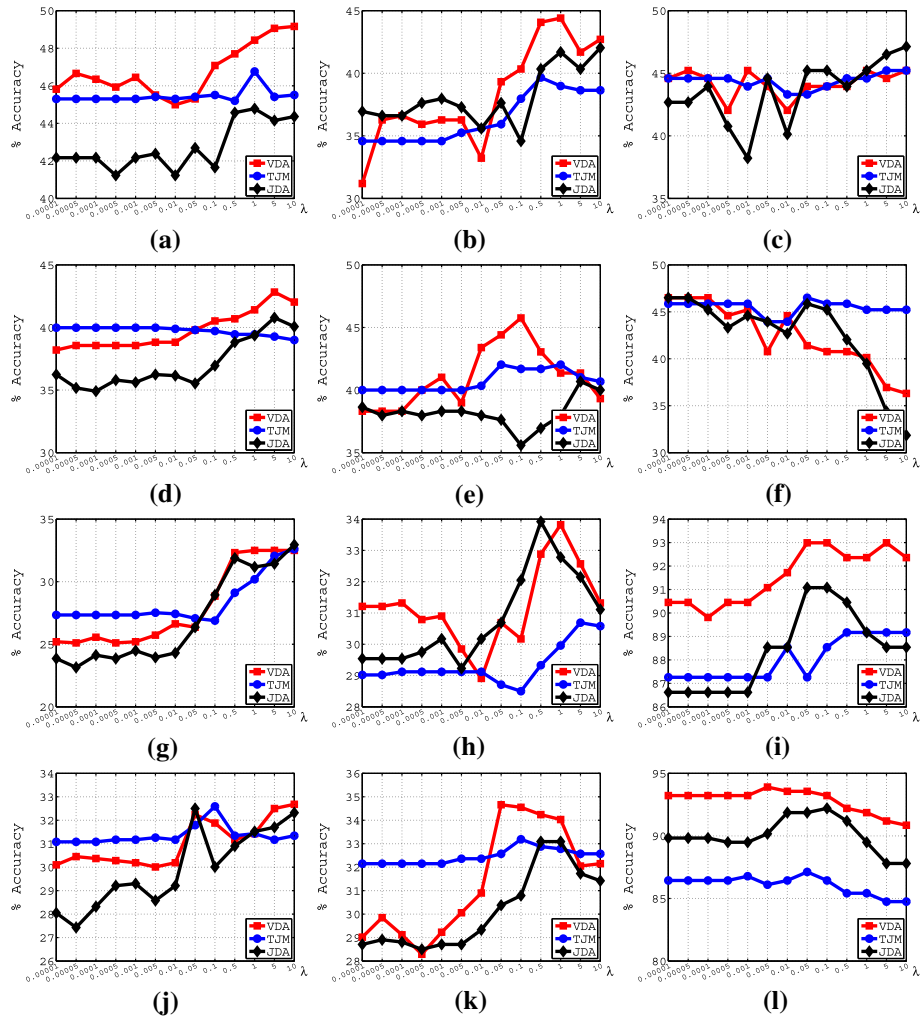


**Fig. 5** Parameter evaluation with respect to the classification accuracy (%) and the number of subspace bases,  $k$ , for Office+Caltech datasets. VDA shows better performance in the interval of  $k \in [20\ 100]$ . **a**  $C \rightarrow A$ , **b**  $C \rightarrow W$ , **c**  $D \rightarrow D$ , **d**  $A \rightarrow C$ , **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**  $D \rightarrow W$

difference between domains will still be significantly large. However, PCA shows better performance compared to NN, but it performs poorly against domain adaptation baseline methods.

TCA is a state-of-the-art domain adaptation method that maps original data onto the transfer components. TCA suffers from two major limitations: (1) It transfers domains fully unsupervised and does not exploit source domain labels, and (2) it does not reduce conditional distribution explicitly. VDA exploits source domain labels in composing domain invariant clusters and reduces jointly conditional and marginal distributions.

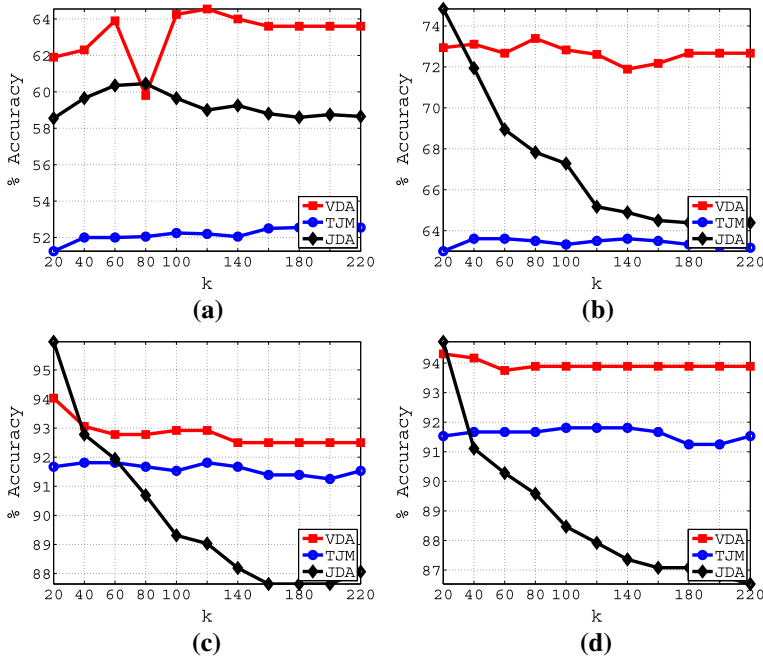
In GFK, since the dimension of subspaces should be small enough (because of smooth transit along flow), the subspaces do not represent original data accurately. Therefore, the



**Fig. 6** Parameter evaluation with respect to classification accuracy (%) and parameter,  $\lambda$ , for Office+Caltech datasets. In general, VDA shows acceptable results with larger values of  $\lambda$ . **a**  $C \rightarrow A$ , **b**  $C \rightarrow W$ , **c**  $D \rightarrow D$ , **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**  $D \rightarrow W$

classification accuracy is affected with respect to geodesic between source and target domains. However, VDA learns a precise shared subspace across domains that exactly reflects input data.

JDA and TJM are very noticeable approaches among current state-of-the-art methods that learn a shared representation by minimizing empirical means of domains. The optimization problem of TJM is very complex, and it uses an iterative alternative to update adaptation matrix, since it optimizes two different criteria simultaneously. JDA performs well, but it adapts in an unsupervised manner, similar to TCA. VDA outperforms both TJM and JDA in 14 out of 16 experiments.



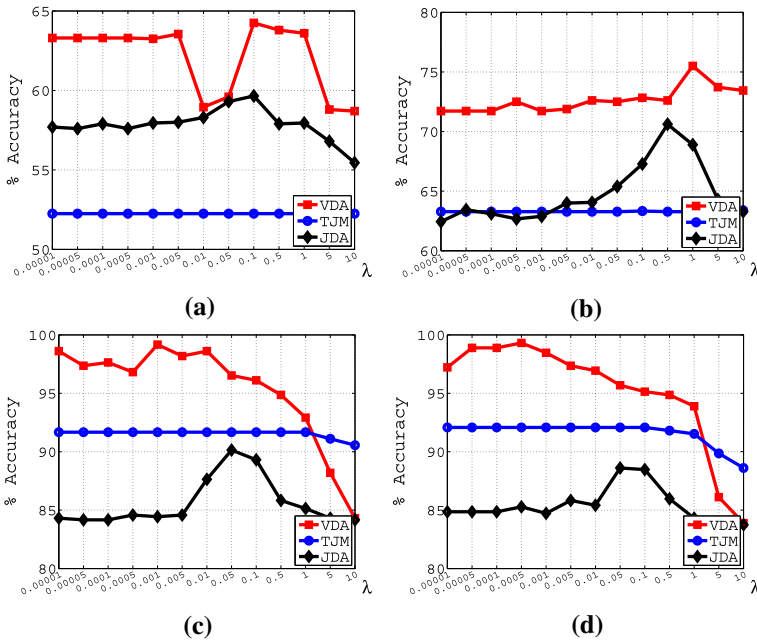
**Fig. 7** Parameter evaluation with respect to the classification accuracy (%) and the number of subspace bases,  $k$ , for Digits and COIL20 datasets. VDA indicates that is not sensitive to the number of subspace bases. **a** USPS versus MNIST, **b** MNIST versus USPS, **c** COIL1 versus COIL2, **d** COIL2 versus COIL1

### 5.2 Effectiveness evaluation

The effectiveness of VDA and three baseline methods is evaluated by comparing their performance in 10 iterations. We run TCA, JDA, TJM and VDA on Office+Caltech, Digits and COIL20 datasets and illustrate the results in Figs. 3 and 4. Later it will be shown that our algorithm converges in 10 iterations.

Figure 3 shows the classification accuracy computed for each method in 10 iterations on Office+Caltech datasets. As is clear from the figures, TCA reduces substantial distribution difference in marginal distributions, but in most cases it shows poor performance against other baseline methods. JDA has very good classification accuracy and outperforms TJM in most cases. However, VDA can perform transfer learning and domain adaptation simultaneously and reduce the distribution difference across domains. Moreover, VDA composes compact domain invariant clusters in embedding and extracts a robust and effective representation for domain shift problem. However, VDA has a sensitivity to problem data where the estimated mean value for clustering is not accurate. In this way, VDA illustrates a slight fluctuation on some datasets, e.g.,  $C \rightarrow A$ .

Figure 4a, b illustrates the performance of VDA and three baseline methods on Digits datasets. As is clear from the figures, despite the close performance of JDA on *USPS vs MNIST* dataset to our approach (3.3 % superiority of ours), VDA outperforms all baseline methods, particularly JDA, on *MNIST vs USPS* dataset significantly (7.44 % superiority). Indeed, VDA shows superior performance on Digits dataset compared to other DA state-of-the-art methods. In Figure 4c, d, the results are very different. Although VDA illustrates poor performance in the starting steps, it shows unexpected progress in the last iterations (from iteration 4



**Fig. 8** Parameter evaluation with respect to the classification accuracy (%) and the regularization parameter,  $\lambda$ , for Digits and COIL20 datasets. We choose  $\lambda \in [0.00001 \ 1.0]$  where the sensitivity of VDA is low for dealing with Digits datasets, and also choose  $\lambda \in [0.00001 \ 0.005]$  on COIL20 datasets. **a** USPS versus MNIST, **b** MNIST versus USPS, **c** COIL1 versus COIL2, **d** COIL2 versus COIL1

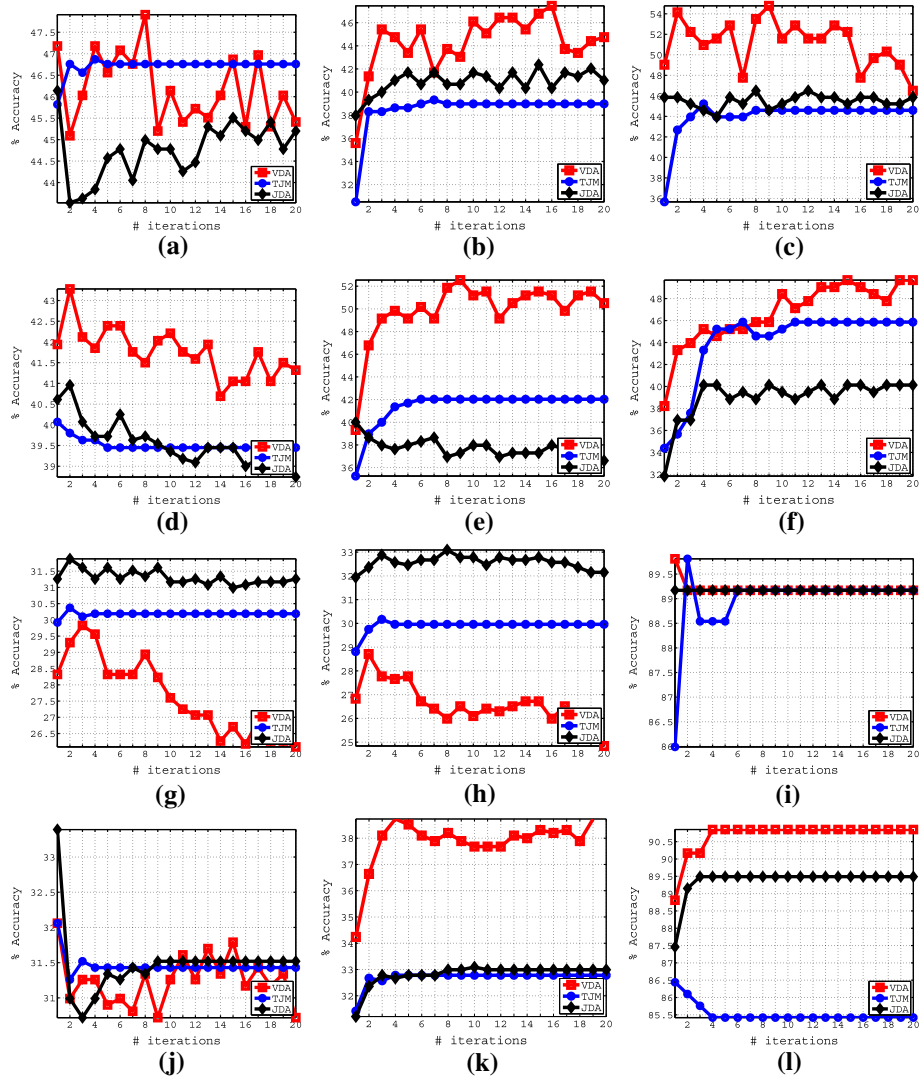
onwards). The performance of VDA in iteration 10 is extraordinary, and it achieves 99.31 and 97.92 % on *COIL1 vs COIL2* and *COIL2 vs COIL1* datasets, respectively. Indeed, VDA misclassifies only 5 out of 720 instances on *COIL1 vs COIL2* dataset and 15 out of 720 instances on *COIL2 vs COIL1* dataset, regarding the significant difference across source and target domains. Indeed, VDA assigns true labels iteratively to target samples. In this way, pre-assigned pseudo-labels in the early steps switch to accurate labels in the final steps.

### 5.3 Impact of parameter settings

VDA is evaluated with respect to different values of parameters to analyze its performance in different conditions. In general, we should tune the number of subspace bases,  $k$ , and the regularization parameter,  $\lambda$ , for VDA on different datasets. We report the results of VDA, JDA and TJM on Office+Caltech, Digits and COIL20 datasets (all three methods need iteration to converge).

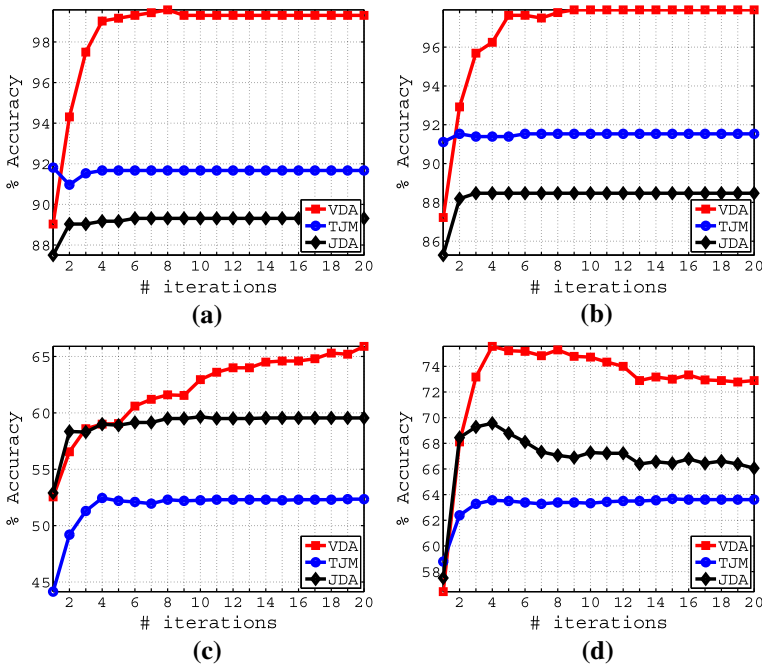
Figure 5 illustrates the experiments on Office+Caltech datasets for evaluating parameter  $k$ . We run VDA, JDA and TJM with varying values of  $k$ . We report the classification accuracy of VDA and baseline methods with  $k \in [20 \ 220]$  on 12 Office+Caltech datasets. The value of  $k$  determines the low-dimensional representation accuracy for data reconstruction. The plots indicate that in most cases increasing the value of  $k$  decreases the VDA performance while the accuracy has negative slope. Indeed, VDA shows better performance in subspaces with low dimension. In this way,  $k \in [20 \ 100]$  for Office+Caltech datasets are chosen. It is worth noting that in some datasets small values of  $k$  show poor performance; however, the overall generalization of VDA with respect to the chosen interval is noteworthy.





**Fig. 9** Convergence evaluation with respect to the classification accuracy (%) in 20 iterations on Office+Caltech20 datasets. In most cases, all three methods converge in first 10 iterations. **a**  $C \rightarrow A$ , **b**  $C \rightarrow W$ , **c**  $D \rightarrow D$ , **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 6 shows the results for parameter  $\lambda$  on Office+Caltech datasets. The shown sub-figures denote the classification accuracy of VDA compared to the baseline methods with  $\lambda \in [0.00001 \ 10]$  on 12 Office+Caltech datasets. As is clear from the plots, in most cases VDA shows acceptable results with large values of  $\lambda$ . Indeed, we choose  $\lambda \in [0.05 \ 10]$  for Office+Caltech datasets. In general, larger values of regularization parameter can give more importance to regularization term, and DA is not performed. However, small values of  $\lambda$  ill-define the optimization problem and make the eigen decomposition difficult.



**Fig. 10** Convergence evaluation with respect to the classification accuracy (%) in 20 iterations on Digits and COIL20 datasets. The classification accuracy of VDA increases consistently with more iteration and converges within 10 iterations. **a** USPS versus MNIST, **b** MNIST versus USPS, **c** COIL1 versus COIL2, **d** COIL2 versus COIL1

Figure 7 illustrates the parameter evaluation with respect to classification accuracy and parameter  $k$  for Digits and COIL20 datasets. We consider  $k \in [20\ 220]$  which is integrated with the aforementioned experiments. The results indicate that VDA on Digits and Coil datasets is not sensitive to the value of  $k$ . In other words, VDA transfers knowledge from source to target domain robustly and effectively on Digits and Coil datasets. Indeed, domain invariant clustering strengthens the extracted embeddings on which the within-class similarity is high while between-class similarity is low. Also, we can conclude that the estimated mean for these datasets has close approximation to the exact mean value of domain.

Figure 8 shows the classification accuracy of VDA, JDA and TJM with respect to  $\lambda \in [0.00001\ 10]$  on Digits and COIL20 datasets. As is clear from the figures, the parameter  $\lambda$  has partly uniform manner on Digits dataset, particularly on *MNIST vs USPS*. We choose  $\lambda \in [0.00001\ 1.0]$  where the sensitivity of VDA is low for dealing with Digits datasets. However, VDA behaves a little differently on COIL20 datasets, while TJM has a rather constant behavior on most values of  $\lambda$ ; VDA shows a descending manner on large values of  $\lambda$ . In other words, when the value of  $\lambda$  increases, VDA cannot construct robust representation for cross-domain classification. In this way, VDA cannot handle DA and TL across domains. As a result, we choose  $\lambda \in [0.00001\ 0.005]$  on COIL20 datasets.

### 5.4 Convergence evaluation

We evaluate the convergence property of VDA by conducting comprehensive experiments on Office+Caltech, Digits and COIL20 datasets and compare the results of VDA against TJM

and JDA. Figure 9 shows the classification accuracy of VDA and other DA baseline methods in 20 iterations on Office+Caltech datasets. The results indicate that all three methods in most cases converge in 10 iterations, but in some datasets they fluctuate, particularly VDA and JDA. The fluctuation for these datasets originates from the imprecise estimated mean value for domain invariant clustering. The amount of fluctuation is variable according to different datasets; however, it has a confined interval after 10 iterations. In this way, we empirically choose to test VDA in 10 iterations and report its performance on Office+Caltech dataset. Figure 10 shows that classification accuracy of VDA increases consistently with more iteration and converges within 10 iterations on Digits and COIL20 datasets. In general, VDA, TJM and JDA have a steady manner when faced with Digits and COIL20 datasets and their classification accuracy follows a persistent curve.

## 6 Conclusion and future work

In this paper, we presented a VDA approach for cross-domain classification. VDA exploits transfer learning and domain adaptation strategies to cope with domain shift problem. Moreover, VDA employs domain invariant clustering to enhance the adaptation performance in a principled dimensionality reduction subspace. The extracted embedding for source and target domains is the most effective and robust representation for cross-domain problems. Performance of VDA is evaluated from different perspectives such as results, effectiveness, parameters and convergence, and its yields are compared with six state-of-the-art baseline methods. Our comprehensive experiments on a variety of vision datasets with different difficulties show that VDA significantly outperforms other adaptation methods. One important direction remains worth researching, to extend VDA to confront different transfer learning and domain adaptation scenarios. Here, we suggest some directions for future VDA extension to interested researchers to investigate: multiple sources, i.e., the number of sources is more than one; zero target training, i.e., no target training data exist; online transfer learning, i.e., using online and real-time data; and inductive transfer learning, i.e., the target test instances are unseen.

## References

1. Gopalan R, Li R, Chellappa R (2014) Unsupervised adaptation across domain shifts by generating intermediate data representations. *IEEE Trans Pattern Anal Mach Intell* 36(11):2288–2302
2. Gong B, Grauman K, Sha F (2013) Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In: *Proceedings of the 30th international conference on machine learning*, pp 222–230
3. Bergamo A, Torresani L (2010) Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In: *Advances in neural information processing systems*, pp 181–189
4. Hoffman J, Kulis B, Darrell T, Saenko K (2012) Discovering latent domains for multisource domain adaptation. In: *Computer Vision–ECCV 2012*, pp 702–715. Springer
5. Chen M, Weinberger KQ, Blitzer J (2011) Co-training for domain adaptation. In: *Advances in neural information processing systems*, pp 2456–2464
6. Baktashmotlagh M, Harandi MT, Lovell BC, Salzmann M (2013) Unsupervised domain adaptation by domain invariant projection. In: *2013 IEEE international conference on computer vision (ICCV)*, pp 769–776
7. Gheisari M, Baghshah MS (2015) Unsupervised domain adaptation via representation learning and adaptive classifier learning. *Neurocomputing* 165:300–311
8. Tahmoresnezhad J, Hashemi S (2015) Common feature extraction in multi-source domains for transfer learning. In *IEEE 2015 7th Conference on information and knowledge technology (IKT)*, pp 1–5

9. Fernando B, Habrard A, Sebban M, Tuytelaars T (2013) Unsupervised visual domain adaptation using subspace alignment. In: 2013 IEEE international conference on computer vision (ICCV), pp 2960–2967
10. Xiong C, McCloskey S, Hsieh S-H, Corso JJ (2014) Latent domains modeling for visual domain adaptation. In: Proceedings of AAAI conference on artificial intelligence (AAAI)
11. Gong B, Grauman K, Sha F (2014) Learning kernels for unsupervised domain adaptation with applications to visual object recognition. *Int J Comput Vision* 109(1–2):3–27
12. Cuong V, Duin RPW, Piqueras-Salazar I, Loog M (2013) A generalized fisher based feature extraction method for domain shift. *Pattern Recognit* 46(9):2510–2518
13. Gopalan R, Li R, Chellappa R (2011) Domain adaptation for object recognition: an unsupervised approach. In: 2011 IEEE international conference on computer vision (ICCV), pp 999–1006
14. Ben-David S, Blitzer J, Crammer K, Pereira F et al (2007) Analysis of representations for domain adaptation. *Adv Neural Inf Process Syst* 19:137
15. Uguroglu S, Carbonell J (2011) Feature selection for transfer learning. In: *Machine learning and knowledge discovery in databases*, pp 430–442. Springer
16. Pan SJ, Kwok JT, Yang Q (2008) Transfer learning via dimensionality reduction. *AAAI* 8:677–682
17. Quanz B, Huan J, Mishra M (2012) Knowledge transfer with low-quality data: a feature extraction issue. *IEEE Trans Knowl Data Eng* 24(10):1789–1802
18. Zhong E, Fan W, Peng J, Zhang K, Ren J, Turaga D, Verscheure O (2009) Cross domain distribution adaptation via kernel mapping. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 1027–1036
19. Sun Q, Chattopadhyay R, Panchanathan S, Ye J (2011) A two-stage weighting framework for multi-source domain adaptation. In: *Advances in neural information processing systems*, pp 505–513
20. Long M, Wang J, Ding G, Sun J, Yu Philip S (2013) Transfer feature learning with joint distribution adaptation. In: 2013 IEEE international conference on computer vision (ICCV), pp 2200–2207
21. Tahmoresnezhad J, Hashemi S (2015) A generalized kernel-based random k-sample sets method for transfer learning. *Iran J Sci Technol Trans Electrical Eng* 39:193–207
22. Gretton A, Borgwardt KM, Rasch M, Schölkopf B, Smola AJ (2006) A kernel method for the two-sample problem. In: *Advances in neural information processing systems*, pp 513–520
23. Jolliffe I (2002) *Principal component analysis*. Wiley, New York
24. Jie L, Behbood V, Hao P, Zuo H, Xue S, Zhang G (2015) Transfer learning using computational intelligence: a survey. *Knowl Based Syst* 80:14–23
25. Pan SJ, Yang Q (2010) A survey on transfer learning. *IEEE Trans Knowl Data Eng* 22(10):1345–1359
26. Blitzer J, McDonald R, Pereira F (2006) Domain adaptation with structural correspondence learning. In: Proceedings of the 2006 conference on empirical methods in natural language processing, Association for Computational Linguistics, pp 120–128
27. Pan SJ, Ni X, Sun J-T, Yang Q, Chen Z (2010) Cross-domain sentiment classification via spectral feature alignment. In: Proceedings of the 19th international conference on World wide web, ACM, pp 751–760
28. Huang J, Gretton A, Borgwardt KM, Schölkopf B, Smola AJ (2006) Correcting sample selection bias by unlabeled data. In: *Advances in neural information processing systems*, pp 601–608
29. Pan SJ, Tsang IW, Kwok JT, Yang Q (2011) Domain adaptation via transfer component analysis. *Neural Netw* 22(2):199–210
30. Daumé H III (2009) Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*
31. Duan L, Tsang IW, Xu D, Maybank SJ (2009) Domain transfer SVM for video concept detection. In: IEEE Conference on computer vision and pattern recognition, CVPR 2009, pp 1375–1381
32. Gong B, Shi Y, Sha F, Grauman K (2012) Geodesic flow kernel for unsupervised domain adaptation. In: 2012 IEEE Conference on computer vision and pattern recognition (CVPR), pp 2066–2073
33. Saenko K, Kulis B, Fritz M, Darrell T (2010) Adapting visual category models to new domains. In: *Computer vision—ECCV 2010*, pp 213–226. Springer
34. Chen Q, Xue B, Zhang M (2015) Generalisation and domain adaptation in GP with gradient descent for symbolic regression. In: 2015 IEEE congress on evolutionary computation (CEC), pp 1137–1144
35. Iqbal M, Browne WN, Zhang M (2014) Reusing building blocks of extracted knowledge to solve complex, large-scale Boolean problems. *IEEE Trans Evol Comput* 18(4):465–480
36. Jiang J, Zhai CX (2007) Instance weighting for domain adaptation in NLP. *ACL* 7:264–271
37. Jiang J (2008) A literature survey on domain adaptation of statistical classifiers. URL: <http://sifaka.cs.uiuc.edu/jiang4/domainadaptation/survey>
38. Duan L, Tsang IW, Xu D, Chua T-S (2009) Domain adaptation from multiple sources via auxiliary classifiers. In: Proceedings of the 26th annual international conference on machine learning, ACM, pp 289–296
39. Long M, Wang J, Ding G, Pan SJ, Yu PS (2014) Adaptation regularization: a general framework for transfer learning. *IEEE Trans Knowl Data Eng* 26(5):1076–1089

40. Bruzzone L, Marconcini M (2010) Domain adaptation problems: a DASVM classification technique and a circular validation strategy. *IEEE Trans Pattern Anal Mach Intell* 32(5):770–787
41. Satpal S, Sarawagi S (2007) Domain adaptation of conditional probability models via feature subsetting. In: *Knowledge discovery in databases: PKDD 2007*, pp 224–235. Springer
42. Si S, Tao D, Geng B (2010) Bregman divergence-based regularization for transfer subspace learning. *IEEE Trans Knowl Data Eng* 22(7):929–942
43. Jhuo I-H, Liu D, Lee DT, Chang S-F et al (2012) Robust visual domain adaptation with low-rank reconstruction. In: *2012 IEEE conference on computer vision and pattern recognition (CVPR)*, pp 2168–2175
44. Qiu Q, Patel VM, Turaga P, Chellappa R (2012) Domain adaptive dictionary learning. In: *Computer vision–ECCV2012*, pp 631–645. Springer
45. Roy SD, Mei T, Zeng W, Li S (2012) Social transfer: cross-domain transfer learning from social streams for media applications. In: *Proceedings of the 20th ACM international conference on multimedia*, pp 649–658
46. Long M, Wang J, Ding G, Sun J, Yu PS (2014) Transfer joint matching for unsupervised domain adaptation. In: *2014 IEEE conference on computer vision and pattern recognition (CVPR)*, pp 1410–1417
47. Long M, Wang J, Sun J, Yu PS (2015) Domain invariant transfer kernel learning. *IEEE Trans Knowl Data Eng* 27(6):1519–1532
48. Griffin G, Holub A, Perona P (2007) Caltech-256 object category dataset
49. Bay H, Tuytelaars T, Van Gool L (2006) Surf: speeded up robust features. In: *Computer vision–ECCV 2006*, pp 404–417. Springer
50. Nene SA, Nayar SK, Murase H et al (1996) Columbia object image library (coil-20). Technical report, TechnicalReport CUCS-005-96



**Jafar Tahmoresnezhad** received the PhD degree in computer science from Shiraz University, Shiraz, Iran, in 2015. Following academic appointments at Urmia University of Technology, he is currently an assistance professor at Faculty of IT and Computer Engineering, Urmia University of Technology, Urmia, Iran. His research interests include pattern recognition, transfer learning, deep learning, data mining and computer security.



**Sattar Hashemi** received the PhD degree in computer science from Iran University of Science and Technology in conjunction with Monash University, Australia, in 2008. Following academic appointments at Shiraz University, he is currently an associate professor at Electrical and Computer Engineering School, Shiraz University, Shiraz, Iran. He is recognized for contributions in the fields of machine learning and data mining. He has published many refereed papers and book chapters on data stream classification, game theory, social networks, database intrusion detection and computer security.