Representative Vector Machines: A Unified Framework for Classical Classifiers

Jie Gui, *Member, IEEE*, Tongliang Liu, Dacheng Tao, *Fellow, IEEE*, Zhenan Sun, *Member, IEEE*, and Tieniu Tan, *Fellow, IEEE*

***Abstract*—Classifier design is a fundamental problem in pattern recognition. A variety of pattern classification methods such as the nearest neighbor (NN) classifier, support vector machine (SVM), and sparse representation-based classification (SRC) have been proposed in the literature. These typical and widely used classifiers were originally developed from different theory or application motivations and they are conventionally treated as independent and specific solutions for pattern classification. This paper proposes a novel pattern classification framework, namely, representative vector machines (or RVMs for short). The basic idea of RVMs is to assign the class label of a test example according to its nearest representative vector. The contributions of RVMs are twofold. On one hand, the proposed RVMs establish a unified framework of classical classifiers because NN, SVM, and SRC can be interpreted as the special cases of RVMs with different definitions of representative vectors. Thus, the underlying relationship among a number of classical classifiers is revealed for better understanding of pattern classification. On the other hand, novel and advanced classifiers are inspired in the framework of RVMs. For example, a robust pattern classification method called discriminant vector machine (DVM) is motivated from RVMs. Given a test example, DVM first finds its *k*-NNs and then performs classification based on the robust M-estimator and manifold regularization. Extensive experimental evaluations on a variety of visual recognition tasks such as face recognition (Yale and face recognition grand challenge databases), object categorization (Caltech-101 dataset), and action recognition (Action Similarity LAbeliNg) demonstrate the advantages of DVM over other classifiers.**

Manuscript received January 31, 2015; revised May 11, 2015; accepted July 1, 2015. Date of publication August 13, 2015; date of current version July 15, 2016. This work was supported in part by the National Basic Research Program of China under Grant 2012CB316300, in part by the National Science Foundation of China under Grant 61420106015, Grant 61135002, Grant 61272333, Grant 61572463, and Grant 61273272, in part by the Post-Doctoral Science Foundation of China under Grant 2012M520021 and Grant 2013T60195, and in part by the Australian Research Council Projects under Grant FT-130101457 and Grant LP-140100569. This paper was recommended by Associate Editor X. Wang. *(Jie Gui and Tongliang Liu contributed equally to this work.)*

J. Gui is with the Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China, and also with the Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: guijie@ustc.edu).

T. Liu and D. Tao are with the Centre for Quantum Computation & Intelligent Systems and the Faculty of Engineering and Information Technology, University of Technology, Sydney, Ultimo, NSW 2007, Australia (e-mail: tongliang.liu@student.uts.edu.au; dacheng.tao@uts.edu.au).

Z. Sun and T. Tan are with the Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: znsun@nlpr.ia.ac.cn; tnt@nlpr.ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2015.2457234

***Index Terms*—Discriminant vector machine (DVM), pattern classification, representative vector machines (RVMs), sparse representation, support vector machines (SVMs).**

I. Introduction

CLASSIFICATION [1]–[6] is one of the most fundamental problems in pattern recognition, machine learning and statistics, and numerous classification algorithms have been proposed for different computer vision and pattern recognition tasks. The most widely used classifiers include neural network [7]–[10], support vector machines (SVMs) [11]–[14], *k*-nearest neighbor (NN) [15]–[18], Gaussian mixture model [7], naive Bayes [19], [20], and decision tree [21].

Among the classifiers, NN [15], [22] is a simple yet popular method for classification, but it is a lazy algorithm without training. Given a new example, NN classifies the example as the class of the nearest training example to the observation. Therefore, NN is sensitive to noise.

Nearest feature line (NFL) [23], [24], nearest feature plane (NFP) [25], and nearest feature space (NFS) [25], [26] are representative variants of NN. Any two examples in the same class form a feature line in NFL, while any three examples in the same class form a feature plane (FP) in NFP. All the examples in the same class form a feature space in NFS. NFL, NFP, and NFS classify a test example as the class whose respective feature line, feature plane, and feature space are the nearest to the test example, respectively. NFL and NFP are sensitive to noise, and NFS does not perform well when classes are highly correlated with each other [27].

SVMs [11] construct a hyperplane for binary classification (or a set of hyperplanes for multiclass classification), which maximizes the margin between classes. SVMs have been widely applied in real-world computer vision problems, such as object recognition, pedestrian detection, and pose estimation, due to the good generalization performance.

Recently, sparse learning and compressive sensing [28]–[31] which have been used in classifier design, have performed particularly well in computer vision tasks. A typical example is sparse representation-based classification (SRC) [27]. In [27], a test example is first sparsely encoded over training examples based on lasso [32], and then classified by finding the class that yields the minimum reconstruction error. SRC demonstrates impressive face recognition performance, and is a successful application of lasso for face analysis. Related works can be found in [33]–[39].

These classification algorithms (NN and its variants, SVM and SRC) are conventionally regarded as individually specific and uncorrelated solutions to pattern recognition because they were proposed under significantly different theoretic motivations or application backgrounds. The underlying relationship among these pattern classification methods have not been well addressed in the literature. Therefore, it is desirable to provide a unified perspective to understand the most widely used pattern classification methods such as NN, SVM, and SRC. More importantly advanced ideas on pattern classification can be motivated from the insight of general classification framework. This paper attempts to find a unified representation of classical classifiers, so a novel scheme of pattern classification termed representative vector machines (or RVMs for short) is proposed for this purpose. The core idea of RVMs is to find a representative vector of each class for a test example so that the test example is then classified into the class with the nearest representative vector. It is interesting to find that classical classifiers such as NN, NFL, NFS, SRC, and SVMs can all be interpreted as specific implementations of RVMs. In this way, the underlying pros and cons of different classification algorithms can be directly compared by analyzing the differences in the design of the representative vectors.

Furthermore, a novel and advanced solution for robust pattern classification, named discriminant vector machine (DVM) is motivated from the general framework of RVMs. To suppress the effect of outliers, DVM first finds the k-NNs of a test example, and then classifies it based on the robust M-estimator and manifold regularization. Comprehensive experimental evaluations of DVM in comparison with other classification algorithms demonstrate the effectiveness of DVM for various recognition tasks.

II. Representative Vector Machines

Consider a dataset X, which consists of n examples in a high-dimensional space R^d. Denote by $X_i = [x_{i1}, \ldots, x_{ij}, \ldots, x_{in_i}] \in R^{d\times n_i}$ the training examples of the ith object class, where $x_{ij}(1 \le j \le n_i)$ is the jth example in the ith class. Suppose we have c classes of examples, and let $X = [X_1, X_2, \ldots, X_c]$ be the concatenation of all training examples. Given a test example y, the objective of classification is to predict the label of y.

A number of different approaches have been proposed for robust classification, usually for different purposes. Here, we reformulate them within the unified framework of RVMs as follows:

$$i^* = \arg\min_i \|y - a_i\| \tag{1}$$

where a_i is the representative vector to represent the ith class for y, and i^* is the predicted class label for y. The representative vectors of classical classifiers are summarized in Table I.

TABLE I
Representative Vectors of Classical Classifiers

Methods	Representative vector for class i
NN	$\arg\min_{x_{ij}} \|y - x_{ij}\|$ s.t. $j = 1, \cdots, n_i$
NFL	$\arg\min_{q^i_{jk}} d\left(y, q^i_{jk}\right)$ s.t. $j, k = 1, \cdots, n_i; j \neq k$
NFP	$\arg\min_{p^i_{jkl}} d\left(y, p^i_{jkl}\right)$ s.t. $j, k, l = 1, \cdots, n_i; j \neq k \neq l$
NFS	$X_i\beta_i$
NC	the mean vector of class i : $m_i = \left(\sum_{j=1}^{n_i} x_{ij}\right)\Big/ n_i$
SRC	$X_i\hat{\alpha}_i$
SVMs	$y + \left((i - b - w^T y)/w^T w\right) w$

A. Interpretation of NN, NFL, NFP, NFS, and NC Using RVMs

The NN classifier classifies a test example y according to the label of its NN. Based on this, the decision function of class i is

$$d_i(y) = \min_{j=1,\ldots,n_i} \left\|y - x_{ij}\right\|,\ i = 1, 2, \ldots, c. \tag{2}$$

The decision rule of NN is to assign y to class m if $d_m(y) = \min_{i=1,\ldots,c} d_i(y)$. For NN, the representative vector for class i is $\arg\min_{x_{ij}} \|y - x_{ij}\|$ s.t. $j = 1, \ldots, n_i$.

For the NFL classifier, any two examples of the same class are generalized by the feature line (FL) passing through the two examples. The classification is based on the shortest distance from a query example to each FL. The straight line passing through x_{ij} and x_{ik} of the class i, denoted as $\overline{x_{ij}x_{ik}}(j, k = 1, \ldots, n_i; j \neq k)$, is called an FL of class i. The query example y is projected onto the FL $\overline{x_{ij}x_{ik}}$ as point q^i_{jk} [Fig. 1(a)]. The FL distance between y and $\overline{x_{ij}x_{ik}}$ is defined as $d(y, \overline{x_{ij}x_{ik}}) = d(y, q^i_{jk})$. The decision function of class i is

$$d_i(y) = \min_{\substack{j,k=1,\ldots,n_i \\ j\neq k}} d\left(y, q^i_{jk}\right),\ i = 1, 2, \ldots, c. \tag{3}$$

NFL assigns y to class m if $d_m(y)$ is the minimum. For NFL, the representative vector for class i is $\arg\min_{q^i_{jk}} d(y, q^i_{jk})$ s.t. $j, k = 1, \ldots, n_i; j \neq k$.

For the NFP classifier, any three examples of the same class are generalized by the FP passing through the three examples. The classification is based on the shortest distance from a query example to each FP. The plane passing through x_{ij}, x_{ik}, and x_{il} of the class i, denoted as $\overline{x_{ij}x_{ik}x_{il}}(j, k, l = 1, \ldots, n_i; j \neq k \neq l)$, is called an FP of class i. The query example y is projected onto the FP $\overline{x_{ij}x_{ik}x_{il}}$ as point p^i_{jkl} [Fig. 1(b)]. The distance between y and $\overline{x_{ij}x_{ik}x_{il}}$ is defined as $d(y, \overline{x_{ij}x_{ik}x_{il}}) = d(y, p^i_{jkl})$. The decision function of class i is

$$d_i(y) = \min_{\substack{j,k,l=1,\ldots,n_i \\ j\neq k\neq l}} d\left(y, p^i_{jkl}\right),\ i = 1, 2, \ldots, c. \tag{4}$$

NFP assigns y to class m if $d_m(y)$ yields the minimum. For NFP, the representative vector for class i is $\arg\min_{p^i_{jkl}} d(y, p^i_{jkl})$ s.t. $j, k, l = 1, \ldots, n_i; j \neq k \neq l$.

For the NFS classifier, NFS assigns a test example y to class i if the distance from y to the subspace spanned by all examples $X_i = [x_{i1}, \ldots, x_{ij}, \ldots, x_{in_i}]$ of class i

$$d_i(y) = \min_{\beta_i} \|y - X_i\beta_i\| \tag{5}$$

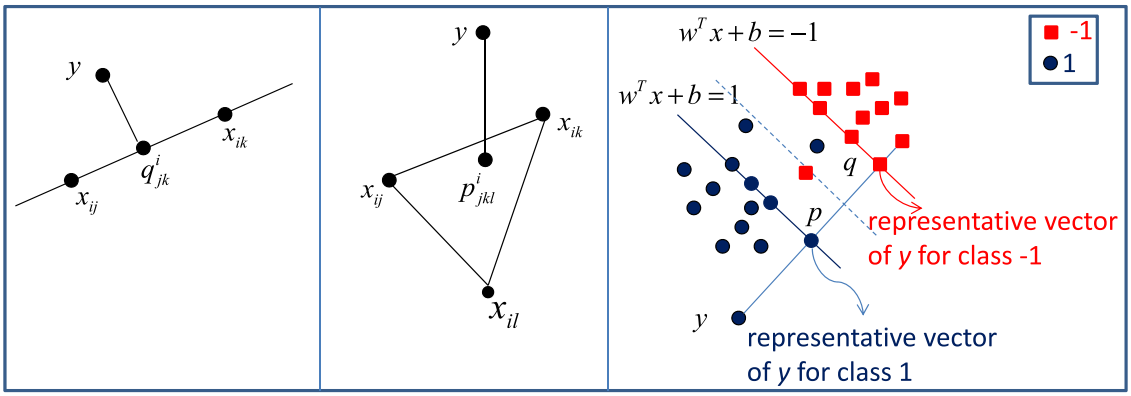


Fig. 1. Illustration of the basic ideas of pattern classifiers NFL, NFP, and SVM. (a) NFL classifier. The feature line $\overline{x_{ij}x_{ik}}$ is generalized by the two examples x_{ij} and x_{ik}. The example y is projected onto the line as point q^i_{jk}. (b) NFP classifier. The FP $\overline{x_{ij}x_{ik}x_{il}}$ is generalized by the three examples x_{ij}, x_{ik}, and x_{il}. The example y is projected onto the plane as point p^i_{jkl}. (c) SVMs. The example y is projected onto the plane $w^T x + b = 1$ as point p and projected onto the plane $w^T x + b = -1$ as point q.

is the minimum among all classes. The closed-form solution of (5) can be easily and directly derived as

$$\beta_i = \left(X_i^T X_i\right)^{-1} X_i^T y. \tag{6}$$

LRC [26] solves (5) by (6), while the nearest linear combination (NLC) [40] solves (5) by means of the pseudo-inverse matrix technique. Based on (6), (5) can be reduced as

$$d_i(y) = \min_i \left\| y - X_i\left(X_i^T X_i\right)^{-1} X_i^T y \right\|. \tag{7}$$

The nearest subspace algorithm [41] assumes that the columns of X_i are orthonormal and thus solves

$$d_i(y) = \min_i \left\| y - X_i X_i^T y \right\|. \tag{8}$$

For all NFS related algorithms, the representative vector for class i is $X_i\beta_i$ in (5).

The nearest centroid classifier (NC) classifies a test example y according to the label of its nearest centroid (mean vector of each class in the training set). First, the mean vector of each class in the training set is computed $m_i = (\sum_{j=1}^{n_i} x_{ij})/n_i$. The distance to each centroid is then given by

$$d_i(y) = \|y - m_i\|. \tag{9}$$

NC assigns y to class m if $d_m(y)$ yields the minimum. For NC, the representative vector for class i is m_i.

B. Interpretation of SRC Using RVMs

SRC encodes a test example y over the basis X such that $y = X\alpha + e$ and e is the error (noise) vector. The sparsity can be measured by l_1-norm: min $\|\alpha\|_1 + \|e\|_1$ s.t. $\|y - X\alpha - e\| < \varepsilon$, where ε is a small constant. SRC assigns y to class i if

$$d_i(y) = \min \left\| y - X_i\hat{\alpha}_i \right\| \tag{10}$$

is the minimum among all classes where $\hat{\alpha}_i$ is the coding coefficient vector associated with class i. For SRC, the representative vector for class i is $X_i\hat{\alpha}_i$ in (10).

C. Interpretation of SVMs Using RVMs

For binary classification using SVMs, a query example y is projected onto $w^T x + b = 1$ as point p [Fig. 1(c)]. Therefore, we have $p - y = \alpha w$, i.e., $p = y + \alpha w$. Since $w^T p + b = 1$ and $p = y + \alpha w$, we have

$$w^T(y + \alpha w) + b = 1. \tag{11}$$

Thus

$$\alpha = \left(1 - b - w^T y\right)/\left(w^T w\right). \tag{12}$$

The representative vector for class $i = 1$ is $y + ((1 - b - w^T y)/w^T w)w$. Similarly, the representative vector for class $i = -1$ is $y + ((-1 - b - w^T y)/w^T w)w$. For SVMs, the representative vector for class i is $y + ((i - b - w^T y)/w^T w)w$.

D. Analysis of Different Classifiers

We have demonstrated that a number of traditional classifiers can be explained as the special cases of RVMs. Although these classifiers are unified into a general framework of pattern classification, they have some specific features and an in-depth analysis of their differences is beneficial to practical applications of these classifiers.

NN is a simple yet popular method for classification due to easy implementation and efficiency. In some practical applications such as face recognition, there are only a small number of examples available per class. It is desirable to have a sufficiently large number of examples stored to cover variations of pose, illumination, and expression for each class. In order to solve this problem, NFL and NFP are proposed to generalize the representation capacity of available prototype images. NFL and NFP were shown to achieve lower classification error than NN in [25]. However, when there are a large number of training examples for each class, NFL and NFP are costly and time-consuming. For example, the Action Similarity LAbeliNg (ASLAN) dataset [42] consists of 6000 examples from two classes. For each class, there are 3000 examples.

TABLE II
COMPARISON OF NUMBER OF CLASSIFIERS

Classifier	Advantage	Disadvantage
NN	• Easy implementation • Learning is fast	Sensitive to noise
NFL/NFP	• Easy in software programming • Generalize the representation capacity in case of only a small number of examples available per class [23]–[25]	• Sensitive to noise • Costly and time-consuming when there are a large number of training examples per class
NFS	• Easy to implement • Quick at learning	Poor performance when classes are highly correlated [27]
NC	• Easy to be programmed • Computationally efficient	Poor performance when the number of the examples of some class is small
SRC	Impressive performance in some practical applications such as face recognition [27]	Computationally expensive [33]
SVMs	Good generalization ability on unseen data	Great computational complexity in some cases where large training set are involved

According to [42], 2700 examples of each class are used for training. For each test example, NFL has to compute $2C_{2700}^2$ times of point to line distance while NFP has to compute $2C_{2700}^3$ times of point to plane distance. Therefore, NFL and NFP are not suitable for the dataset which consists of a large number of training examples for each class. NC may be a better choice according to computational efficiency in this case. Since there are a large number of training examples for each class, the mean vector is a good representative vector of each class and NC may work well. Furthermore, NC is obviously more efficient than NFL and NFP in this situation.

Since the representative vectors of each class for NN, NFL, and NFP are related to only one example, two examples, and three examples, respectively, all three are sensitive to noise.

NFS is easy to implement and efficient in learning. However, NFS does not perform well when classes are highly correlated. The reason is that there is no significant difference among the representative vectors for different classes in this case. This is consistent with the description in [27]. If all elements of β_i in (5) are equal to $1/n_i$, NFS is equivalent to NC.

NC is computationally efficient and easy in implementation. When there are a large number of examples for each class, the mean vector m_i is a good representative vector to represent the ith class. However, m_i is not a good choice when the number of the examples for class i is small.

SRC has been studied extensively and has demonstrated impressive results for face recognition. However, it requires solving a time consuming l_1-norm minimization before computing the representative vectors [33].

SVMs have the best generalization ability on the unseen data in comparison with other methods. Before computing the representative vectors, SVMs have to find the optimal separating hyper plane. Thus, SVMs are computationally expensive in some cases where large training set are involved.

The merits of RVMs are at least twofolds. First, given the flexibility of RVMs, the underlying pros and cons of different classification algorithms can be directly compared by analyzing the differences in the design of the representative vectors. Table II briefly summarizes the important advantages and disadvantages of different classifiers. However, not all classifiers can be included in the framework of RVMs such as the Naive Bayes classifier, which remains an open problem. Second, RVMs can be used as a general platform for developing new classification algorithms. A novel and advanced solution for robust pattern classification, named DVM is motivated from the general framework of RVMs.

III. DISCRIMINATIVE VECTOR MACHINE

RVMs not only provide a new perspective to understand the characteristics of classical classifiers, but also identify the possibilities to improve the performance of existing classifiers. In this section, a novel robust classification algorithm, called DVM, is developed as an application of the proposed RVMs framework. We first present the model of DVM, and then derive the statistical analysis of DVM.

A. Model

For each query example y, DVM first finds the k-NNs of y to suppress the effect of outliers. The k-NNs are denoted as a matrix $A_k = [a_1, a_2, \ldots, a_k]$. We also denote A_k as $A_k = [A_{k1}, A_{k2}, \ldots, A_{kc}]$ for derivation convenience, where A_{kj} are the examples from class j. It is possible that some matrixes A_{kj} may be empty, but this has no influence on the following procedures. Then DVM uses the following criterion:

$$\min_{\alpha_k} \sum_{i=1}^{d} \phi\big((y - A_k\alpha_k)_i\big) + \beta\varphi(\alpha_k) + \gamma \sum_{p=1}^{k}\sum_{q=1}^{k} w_{pq}\big(\alpha_k^p - \alpha_k^q\big)^2 \quad (13)$$

to find α_k, where $(y - A_k\alpha_k)_i$ is the ith element of $y - A_k\alpha_k$, α_k^p is the pth element of α_k, α_k can be represented as $\alpha_k = [\alpha_k^1, \alpha_k^2, \ldots, \alpha_k^k]$ or $\alpha_k = [\alpha_{k1}, \alpha_{k2}, \ldots, \alpha_{kc}]$ where α_{kj} is the coefficient with class j and $\phi()$ in the first term

of (13) is the robust M-estimator aiming to enhance robustness [43]. There are a number of possible functions of robust estimators such as Cauchy M-estimator and Welsch M-estimator. We only consider the Welsch M-estimator $\phi(x) = (\sigma^2/2)(1 - \exp(-(x^2/\sigma^2)))$ in this paper where σ is the kernel size and $\varphi()$ in the second term of (13) is the vector norm such as l_1-norm and l_2-norm. In this paper, l_2-norm is used to obtain a closed form solution. The third term of (13) is the manifold regularization where w_{pq} is the similarity between the pth and the qth NN of y. The w_{pq} can be defined by Gaussian kernel or unweighed graph [44], [45]. In this paper, w_{pq} is defined as the cosine distance between the pth and the qth NN of y due to simplicity. It is reasonable to require α_k^p and α_k^q close to each other if the pth and the qth NN of y are close to each other, which is the objective of the third term of (13). Denote W as the similarity matrix constructed by all w_{pq} and D as the diagonal matrix where the ith element of D is the sum of the ith row of W. Thus, we can get the Laplacian matrix $L = D - W$ [45] and the third term of (13) can be represented as $\gamma\alpha_k^T L\alpha_k$ by simple algebra.

According to the multiplicative form of half-quadratic optimization [46], the problem (13) can be solved as follows in an alternate minimization way:

$$p_i = \exp\left(-\left((y - A_k\alpha_k)_i\right)^2 \Big/ \sigma^2\right) \tag{14}$$

$$\alpha_k = \arg\min_{\alpha_k} (y - A_k\alpha_k)^T \mathrm{diag}(p)(y - A_k\alpha_k) + \beta\|\alpha_k\|_2^2 + \gamma\alpha_k^T L\alpha_k \tag{15}$$

where $\mathrm{diag}(p)$ is a diagonal matrix and the ith element is p_i. Fortunately, (15) has a closed form solution as follows:

$$\alpha_k = \left(A_k^T \mathrm{diag}(p)A_k + \beta I + \gamma L\right)^{-1} A_k^T \mathrm{diag}(p)y. \tag{16}$$

The kernel parameter σ in (14) can be updated [35] as follows:

$$\sigma = \sqrt{\left(\theta * (y - A_k\alpha_k)^T * (y - A_k\alpha_k)\right)/d}. \tag{17}$$

where d is the dimension of y, θ is the free parameter and is set to be 1 as in [35]. By using only the coefficients associated with class i, the given query example y can be approximated for each class i as $A_{ki}\alpha_{ki}$. We then classify y based on these approximations by assigning it to the object class that minimizes the residual between y and $A_{ki}\alpha_{ki}$

$$\min_i r_i(y) = \|y - A_{ki}\alpha_{ki}\|, \ i = 1, 2, \ldots c. \tag{18}$$

For relatively large training sets such as Texas Instruments, Inc. and Massachusetts Institute of Technology [47], it is time-consuming to find the k-NNs, so multidimensional indexing methods, such as the k-d tree [38], [39], can be used to speed up exact search. In (14), Gaussian kernel is used, which is also used in discriminant kernel-based SVM (DKSVM) [48]. In DKSVM, the discriminant kernel functions (DKF) including Gaussian-DKF are used. The experimental results show that the discriminant kernel gives the same levels of the classification performance as the linear or radial basis function (RBF) kernels. Furthermore, the visualization results of the kernel matrices show that DKSVMs have more clear kernel matrices than linear or RBF kernel. We note that both DVM and DKSVM are special cases of RVM.

Algorithm 1 DVM

Inputs: training examples and a query example y

Output: identity of y

Find the k-nearest neighbors of y to form $A_k = [A_{k1}, A_{k2}, \ldots, A_{kc}]$.

Solve

$$\min_{\alpha_k} \sum_{i=1}^{d} \phi\left((y - A_k\alpha_k)_i\right) + \beta\varphi(\alpha_k) + \gamma \sum_{p=1}^{k}\sum_{q=1}^{k} w_{pq}\left(\alpha_k^p - \alpha_k^q\right)^2$$

to get α_k as follows:

repeat

$$\sigma = \sqrt{\left(\theta * (y - A_k\alpha_k)^T * (y - A_k\alpha_k)\right)/d}$$

$$p_i = \exp\left(-\left((y - A_k\alpha_k)_i\right)^2 \Big/ \sigma^2\right)$$

$$\alpha_k = \left(A_k^T diag(p)A_k + \beta I + \gamma L\right)^{-1} A_k^T diag(p)y$$

until convergence

Compute the residuals, $r_i(y) = \|y - A_{ki}\alpha_{ki}\|$.

Decision is made in favor of the class with the minimum distance $r_i(y)$.

Algorithm 1 summarizes the complete recognition procedure. The statistical analysis of DVM is given in the next section.

B. Statistical Analysis of DVM

First, we provide a generalization-error-like bound for the DVM algorithm by using the distribution-free inequalities obtained for k-local rules. Then, we prove that DVM algorithm is a probably approximately correct (PAC)-learning algorithm for classification, which means that DVM will be able to learn the target concept if sufficient training examples are provided.

1) Problem Setup: We use the k-local rules setting in Deveroye and Wagner [49]. Let $Z^n = \{(x_1, c_1), \ldots, (x_n, c_n)\}$ be n independent identically distributed training examples drawn from c different classes, $c_n(y, Z^n)$ be the class label learned for the query example y using training examples Z^n. Without considering ties in determining the k nearest observations for any y,[1] a k-local rule is any rule for which

$$C_n\left(y, Z^n\right) = g\left(Z^n\right)$$

where g is an any measurable function. Note that the DVM algorithm is an example of a k-local rule. Define the expected error of the k-local rule as

$$R = P\left\{c_n\left(y, Z^n\right) \neq c \middle| Z^n\right\} \tag{19}$$

where c is the true class label of y. We also define the resubstitution estimate error R_n^R and the deleted estimation error R_n^D as follows:

$$R_n^R = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left[c_n\left(x_i, Z^n\right) \neq c_i\right] \tag{20}$$

$$R_n^D = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left[c_n\left(x_i, Z^{n,i}\right) \neq c_i\right] \tag{21}$$

[1]There have been many ways to avoid ties, for example, the fuzzy k-NN classifier [50].

where $\mathbf{1}[\cdot]$ is the indicator function and $Z^{n,i}$ denotes the training examples with the ith one deleted.

We notice that for DVM algorithm, the resubstitution estimation error R_n^R is very small. If we further assume that the following Assumption 1 holds, then $R_n^R = 0$.

Assumption 1: DVM will always allocate the correct class label to the query example y when the minimum distance $r_i(y)$ is not unique and the distance to the true class is also in the minimum distance set.

The DVM algorithm does not assume any generative models. However, we can analyze the generalization error-like bound for the error distance $|R - R_n^R|$.

PAC learning framework is developed to see if a concept class is learnable. It is also frequently used to seek for efficient algorithms. A PAC-learning algorithm will learn as well as the best model if there are sufficient training examples.

Definition 1 (PAC Learning [51]): A concept class C is said to be PAC-learnable if there exists an algorithm L and a polynomial function poly$(\cdot, \cdot)$ such that for any target concept $c \in C$, any $\epsilon \in (0, 1)$ and any $\delta \in (0, 1)$, if $n \geq \text{poly}(1/\epsilon, 1/\delta)$, for all distribution P^n with probability at least $1 - \delta$, the following holds:

$$R < \epsilon. \tag{22}$$

When such an algorithm L exists, it is called a PAC-learning algorithm for C.

Besides providing a generalization-error like bound, we also prove that DVM is a PAC-learning algorithm for classification.

2) Main Results: Our first result is a generalization-error-like bound for DVM.

Theorem 1: For DVM algorithm with $k \leq n - 1$, we have

$$P\left\{\left|R - R_n^R\right| \geq \epsilon\right\} \leq 2\exp\left(-\frac{n\epsilon^2}{18}\right) + 6\exp\left(-\frac{n\epsilon^3}{108k(2+\gamma_d)}\right)$$

where γ_d is the maximum number of distinct points in $\mathbb{R}^d$ (a d-dimensional Euclidean space) which can share the same NN and $\gamma_d \leq 3^d - 1$.

The proof method of Theorem 1 is the same as that of Theorem 1 in [49]. We also prove that DVM is a PAC-learning algorithm.

Theorem 2: Under Assumption 1, DVM algorithm is a PAC-learning algorithm for classification.

3) Proof: The following lemma [49], [52] will play a central role in proving Theorem 2.

Lemma 1: For DVM algorithm with $k \leq n - 1$, we have

$$\begin{aligned} E\left(R_n^D - R\right)^2 &\leq \frac{1}{n} + 6\max_i P\left\{c_n\left(x, Z^n\right) \neq c_n\left(x, Z^{n,i}\right)\right\} \\ &\leq \frac{1}{n} + \frac{6k}{n}. \end{aligned}$$

Remark 1: Deveroye and Wagner [49] proved a faster convergence rate ($R \to R_n^D$) for $c = 2$.

Proof of Theorem 2: Under Assumption 1, using $(a+b)^2 \leq 2(a^2 + b^2)$ and

$$\begin{aligned} \left|R_n^R - R\right| &\leq \left|R_n^D - R\right| + \left|R_n^D - R_n^R\right| \\ &\leq \left|R_n^D - R\right| + \left|\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\left(c_n\left(x_i, Z^n\right) \neq c_n\left(x_i, Z^{n,i}\right)\right)\right| \end{aligned}$$

we have

$$\begin{aligned} ER^2 &= E\left(R_n^R - R\right)^2 \leq 2E\left(R_n^D - R\right)^2 \\ &\quad + 2E\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\left(c_n\left(x_i, Z^n\right) \neq c_n\left(x_i, Z^{n,i}\right)\right)\right)^2 \\ &\leq 2E\left(R_n^D - R\right)^2 \\ &\quad + 2E\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\left(c_n\left(x_i, Z^n\right) \neq c_n\left(x_i, Z^{n,i}\right)\right) \\ &\leq 2E\left(R_n^D - R\right)^2 \\ &\quad + 2\max_i E\mathbf{1}\left(c_n\left(x_i, Z^n\right) \neq c_n\left(x_i, Z^{n,i}\right)\right) \\ &= 2E\left(R_n^D - R\right)^2 \\ &\quad + 2\left(\max_i EE\mathbf{1}\left(c_n\left(x_i, Z^n\right) \neq c_n\left(x_i, Z^{n,i}\right)|Z^{n,i}\right)\right) \\ &= 2E(R_n^D - R)^2 \\ &\quad + 2\max_i EP\left\{c_n\left(x_i, Z^n\right) \neq c_n\left(x_i, Z^{n,i}\right)|Z^{n,i}\right\} \\ &\leq \frac{2}{n} + \frac{12k}{n} + \frac{2k}{n} \\ &= \frac{2+14k}{n}. \end{aligned}$$

The last inequality holds because of Lemma 1.

Using Chebyshev's inequality $P\{|x| \geq \epsilon\} \leq (Ex^2/\epsilon^2)$, we have

$$P\{|R| \geq \epsilon\} \leq \frac{ER^2}{\epsilon^2} \leq \frac{2+14k}{n\epsilon^2}. \tag{23}$$

Let $(2 + 14k)/(n\epsilon^2) = \delta$, we have $\epsilon = \sqrt{(2+14k)/(n\delta)}$ [and $n = (2 + 14k)/(\delta\epsilon^2)$]. Thus, the following holds with probability at least $1 - \delta$:

$$R \leq \sqrt{\frac{2+14k}{n\delta}}. \tag{24}$$

We have that for any $\epsilon \in (0, 1)$ and any $\delta \in (0, 1)$, when $n \geq (2 + 14k)/(\delta\epsilon^2)$, with probability at least $1 - \delta$, $R \leq \epsilon$. This concludes the proof. ■

IV. Experiments

The proposed DVM is a generic pattern classification method, so a variety of visual recognition tasks including face recognition, object categorization and action recognition are used to evaluate the effectiveness of DVM.

In the face recognition and object categorization tasks, each dataset was partitioned into a training set and a test set containing different numbers. For ease of representation, the experiments were named "p-train," which means that p images per class were selected for training, and the remaining images were used for testing. To robustly evaluate the performance of different algorithms in different training and testing conditions of the Yale face database, we selected p images randomly and ran all possible combinations for each condition. That is to say, there are C_{11}^p runs for p-train. For the remaining three relatively large datasets, we do not run all possible combinations. For the large-scale face recognition grand challenge (FRGC) face dataset, there are 20 random splittings. For the Caltech 101, there are five partitions as did in the "spatial pyramid

Fig. 2. Eleven cropped and resized examples of one person in the Yale face database.

matching based on sparse coding (ScSPM) MATLAB codes for image classification."[2] For the dataset ASLAN [42], the original partitions are used and there are ten partitions of the training and test datasets. Thus, there are 20, 5, and 10 runs for FRGC, Caltech 101, and ASLAN, respectively. We presented the results in the form of the mean recognition rate with standard deviation.

We compared our algorithm with classical algorithms, including NN, NC, NFL, NFP, NFS, SRC, SVM with linear kernel (LSVM), and SVM with Gaussian kernel (GSVM). The LibSVM [53] is used for both LSVM and GSVM. For SRC, LSVM (GSVM) and DVM, the results depend on the choice of the parameter ε, C, and k, respectively. We choose the best parameter through fivefold cross-validation.[3] The regularization parameter C of LSVM and GSVM is set by fivefold cross-validation from {100, 200, 300, 400, 500, 600, 700, 800, 900, 1000}. For GSVM, the Gaussian kernel between x_i and x_j is defined as $\exp\{-\lambda\|x_i - x_j\|_2^2\}$. We set $d_m = n^2/\sum_{i,j=1}^{n}\|x_i - x_j\|_2^2$ [54], [55]. The parameter λ is set by fivefold cross-validation from $\{d_m/8,\ d_m/4,\ d_m/2,\ d_m,\ 2d_m,\ 4d_m,\ 8d_m\}$. The parameters β, γ and θ of DVM are empirically set as 0.01, 0.001, and 1, respectively.

A. Experimental Results on the Yale Database

The Yale face database[4] was constructed at the Center for Computational Vision and Control at Yale University. There are 165 grayscale images of 15 subjects (each individual having 11 different images). The images include variations in lighting conditions (right-light, center-light, and left-light), facial expression (normal, sleepy, surprised, happy, sad, and wink), and with/without glasses. All images were cropped and resized to 32×32 pixels. We preprocessed the data by normalizing each face vector to the unit. Fig. 2 shows sample images of one person.

Table III shows the average recognition rates of each method and their corresponding standard deviations (std), where the best results are highlighted in bold. There is no result for "2 Train" of NFP since NFP requires that the number of the training examples per class should be at least three. These experimental results are also shown in Fig. 3. Due to space limitations, we only present the NN, NC, NFS, linear SVM, and DVM curves in Fig. 3. As can be seen, DVM outperforms all other methods in all cases, while the NN method has the poorest performance except "9 Train" and "10 Train."

[2]http://www.ifp.illinois.edu/~jyang29/

[3]The process of fivefold cross-validation is described here. We first split the whole dataset into a training set and a testing set, and then we take the training set and split it into fivefolds. During the cross-validation, we take fourfolds for training and the left fold for testing, and repeat the process five times and choose the parameter settings with the highest average accuracy. Then the parameter will be used to learn the whole training set and classify the testing set.

[4]http://cvc.yale.edu/projects/yalefaces/yalefaces.html

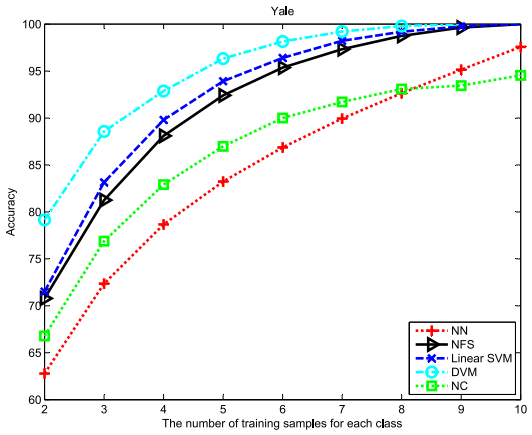


Fig. 3. Average recognition rates (percent) as functions of the number of training examples per class on Yale.

Fig. 4. Ten cropped and resized examples of one person in the FRGC face database.

B. Experimental Results on Large-Scale Face Database FRGC

The FRGC version two face database [56] is a large-scale and challenging benchmark face database. There are 8014 grayscale images from 466 individuals in the query set for FRGC experiment 4. These uncontrolled images demonstrate variations in expression, illumination, blurring, and time. In our experiment, we only selected the individuals which have over ten images in the database, then had 3160 images from 316 individuals. The size of each cropped image in all the experiments is 32×32 pixels by fixing the positions of two eyes, with 256 gray levels per pixel. For each selected person, seven images were randomly selected for training and the rest were used for testing. Fig. 4 shows some images in the FRGC database.

Four features are used on FRGC, including the original 32×32 pixels representation (OR), local binary pattern (LBP), linear discriminant analysis (LDA), and LBP plus LDA (LBPLDA). The experimental results are shown in Table IV. DVM performs the best using LDA and LBPLDA while SRC performs the best using OR and LBP.

C. Experiments Using the Image Categorization Dataset Caltech-101

The Caltech-101 dataset [57] contains 9144 images from 101 classes, including objects such as airplanes, chairs, elephants. Fig. 5 shows the sample images of Caltech-101 (randomly selected 20 classes). We followed the common experimental setup for this dataset, training on 15 and 30 images per class and testing on the remainder. In our analysis, we used the ScSPM feature on the scale-invariant feature transform descriptors of the images, as proposed by Yang *et al.* [58]. Principal component analysis is used to preserve 98% energy. NN, NC, NFL,

TABLE III
AVERAGE RECOGNITION RATES (PERCENT) ACROSS ALL POSSIBLE PARTITIONS ON THE YALE DATABASE AND THE CORRESPONDING STANDARD DEVIATIONS (STD)

Method	2 Train	3 Train	4 Train	5 Train	6 Train	7 Train	8 Train	9 Train	10 Train
NN	62.79±22.80	72.36±19.92	78.67±17.94	83.23±16.64	86.87±15.44	89.94±14.10	92.65±12.55	95.15±10.62	97.58±8.04
NC	66.79±20.83	76.89±17.34	82.91±14.55	86.98±11.82	90.00±9.73	91.72±7.82	93.09±6.46	93.45±4.71	94.55±2.70
NFL	70.67±19.36	80.81±15.40	86.93±12.98	91.66±10.30	95.01±7.85	97.31±5.54	98.79±3.40	99.64±1.53	**100±0**
NFP	-	81.54±15.26	88.38±11.47	93.10±8.44	96.32±6.01	98.36±3.80	99.43±2.00	99.88±0.90	**100±0**
NFS	70.79±19.09	81.25±15.31	88.10±11.56	92.41±8.96	95.37±6.83	97.33±4.84	98.75±3.00	99.64±1.53	**100±0**
SRC	78.79±15.45	87.27±11.54	91.92±8.66	94.57±6.59	96.36±5.13	97.47±4.15	98.42±3.11	98.79±2.60	99.39±2.01
LSVM	71.52±18.88	83.15±13.80	89.80±10.80	93.93±8.06	96.41±6.01	98.22±4.07	99.19±2.42	99.76±1.26	**100±0**
GSVM	71.52±18.57	82.91±13.90	89.74±10.90	93.93±8.16	96.58±5.80	98.24±3.93	99.27±2.21	99.88±0.90	**100±0**
DVM	**79.15±14.63**	**88.57±10.99**	**92.87±8.83**	**96.33±6.15**	**98.15±4.17**	**99.21±2.34**	**99.80±1.15**	**100±0**	**100±0**

TABLE IV
AVERAGE RECOGNITION RATE (PERCENT) COMPARISON ON THE FRGC DATASET

Method	NN	NC	NFL	NFP	NFS	SRC	LSVM	GSVM	DVM
OR	78.98±1.08	55.51±1.31	85.56±1.08	88.31±0.99	89.94±0.92	**95.49±0.72**	91.00±0.83	90.93±0.87	88.41±0.98
LBP	88.52±1.12	78.33±0.91	93.37±1.01	93.38±1.06	93.42±0.99	**97.56±0.46**	95.27±0.91	95.07±0.93	97.28±0.61
LDA	93.61±0.76	93.74±0.79	94.47±0.83	94.56±0.86	94.42±0.84	93.90±0.70	92.65±0.86	94.11±0.86	**95.33±0.64**
LBPLDA	96.00±0.66	95.94±0.54	95.99±0.64	95.94±0.69	95.30±0.71	95.92±0.67	95.77±0.48	95.92±0.67	**96.16±0.55**

TABLE V
AVERAGE RECOGNITION RATES (PERCENT) COMPARISON ON THE CALTECH-101 DATASET

Method	15 Train	30 Train
LCC + SPM + Linear SVM [59]	65.43	73.44
Boureau et al. [60]	-	77.1±0.7
Jia et al. [61]	-	75.3±0.7
ScSPM + LSVM [58]	67.0±0.45	73.2±0.54
ScSPM + GSVM	67.83±0.62	75.78±0.40
ScSPM + NN	49.95±0.92	56.53±0.96
ScSPM + NC	61.27±0.69	65.96±0.63
ScSPM + NFL	63.54±0.68	70.17±0.45
ScSPM + NFP	67.09±0.66	74.04±0.30
ScSPM + NFS	68.63±0.63	76.69±0.34
ScSPM + SRC	71.09±0.57	**78.28±0.52**
ScSPM + DVM	**71.69±0.49**	77.74±0.46

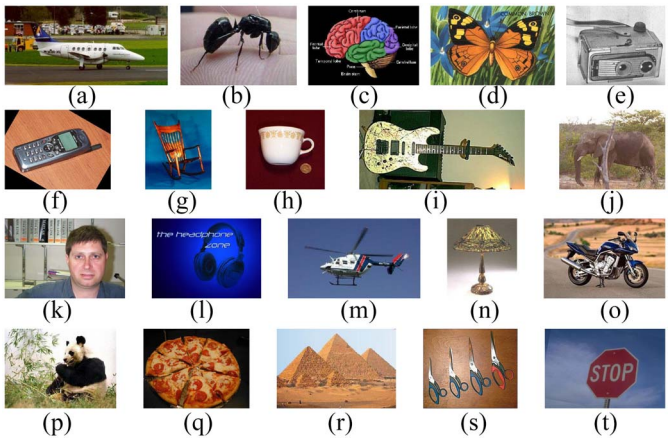


Fig. 5. Sample images of Caltech-101 (randomly selected 20 classes). (a) Airplanes. (b) Ant. (c) Brain. (d) Butterfly. (e) Camera. (f) Cellphone. (g) Chair. (h) Cup. (i) Electric_guitar. (j) Elephant. (k) Faces. (l) Headphone. (m) Helicopter. (n) Lamp. (o) Motorbikes. (p) Panda. (q) Pizza. (r) Pyramid. (s) Scissors. (t) Stop_sign.

NFP, NFS, SRC, and DVM were used for classification. In Table V, we compare the performance of DVM with that of other approaches. The results of the first four methods shown in Table V use linear SVM for classification in their work. As shown in Table V, DVM outperforms other algorithms with similar pipelines for "15 Train."

D. Experiment Using the Action Recognition Dataset ASLAN

The ASLAN dataset [42] includes 1571 videos collected from the Web, in 432 complex action classes, including walking, running, and swimming. The benchmark protocols focus on action similarity (same/not-same), rather than action classification. Thus, it is a binary classification problem.

Table VI compares DVM with other methods and shows that DVM outperforms all the other methods. The results of SVM are reported in [42].

TABLE VI
AVERAGE RECOGNITION RATES (PERCENT) ON THE ASLAN DATABASE AND THE CORRESPONDING STANDARD DEVIATIONS (STD)

Methods	Performance
NN	53.95±0.76
NC	57.38±0.74
NFL	54.25±0.94
NFP	54.42±0.72
NFS	49.98±0.02
SRC	56.40±0.87
LSVM [42]	60.88±0.77
GSVM	61.08±0.71
DVM	**61.37±0.68**

E. Parameter Selection for DVM

In the proposed DVM, there are three parameters, i.e., β, γ, and θ. It is time-consuming to select these parameters using the grid search. Fortunately, these parameters affect the performance slightly if they are set in feasible ranges. Figs. 6–8 show accuracy versus β with γ and θ fixed, γ with β and θ fixed, θ with β and γ fixed, respectively. The proposed DVM model is stable with varying β and γ within $(10^{-4}, 10^{-1})$ and $(10^{-4}, 10^{-3})$, respectively. In all the above experiments, we set $\beta = 0.01$ and $\gamma = 0.001$. The parameter θ is set to be 1 as in [35].

TABLE VII
RESULTS OF TWO-TAILED PAIRED t-TEST ON THE ">" RELATIONSHIP BETWEEN THE TWO ACCURACIES (MEANS) REPORTED IN TABLES III–VI

Data	DVM>NN	DVM>NC	DVM>NFL	DVM>NFP	DVM>NFS	DVM>SRC	DVM>LSVM	DVM>GSVM
Yale 2 Train	1	1	1	-	1	0	1	1
Yale 3 Train	1	1	1	1	1	1	1	1
Yale 4 Train	1	1	1	1	1	1	1	1
Yale 5 Train	1	1	1	1	1	1	1	1
Yale 6 Train	1	1	1	1	1	1	1	1
Yale 7 Train	1	1	1	1	1	1	1	1
Yale 8 Train	1	1	1	1	1	1	1	1
Yale 9 Train	1	1	0	0	0	1	0	0
Yale 10 Train	0	1	0	0	0	0	0	0
FRGC OR	1	1	1	0	0	0	0	0
FRGC LBP	1	1	1	1	1	0	1	1
FRGC LDA	1	1	1	1	1	1	1	1
FRGC LBPLDA	0	1	1	1	1	1	1	1
Caltech 15 Train	1	1	1	1	1	1	1	1
Caltech 30 Train	1	1	1	1	1	0	1	1
ASLAN	1	1	1	1	1	1	0	0

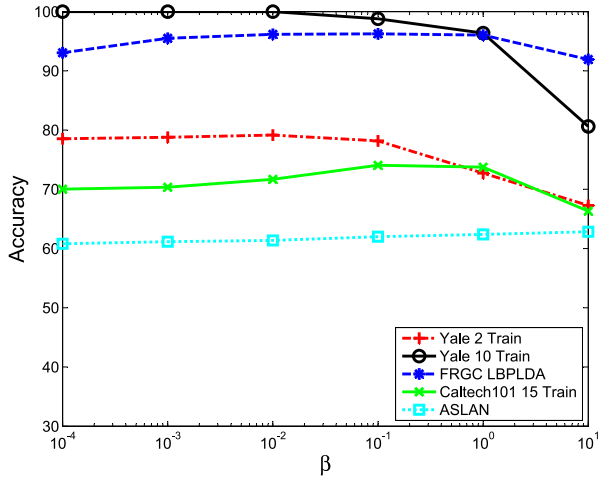


Fig. 6. Accuracy versus β with γ and θ fixed on Yale, FRGC, Caltech 101, and ASLAN. The proposed DVM model is stable with varying β within $(10^{-4}, 10^{-1})$.

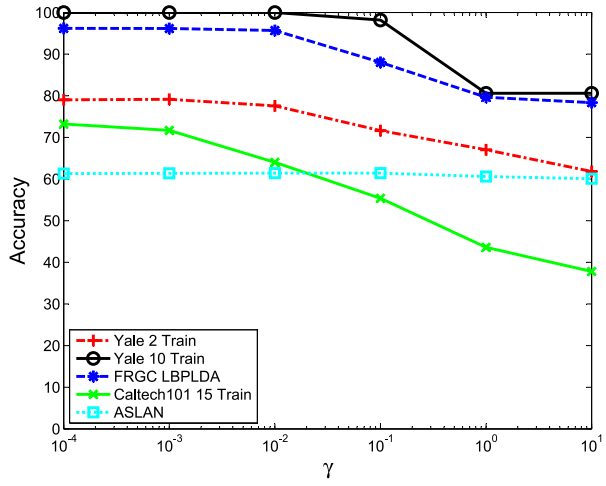


Fig. 7. Accuracy versus γ with β and θ fixed on Yale, FRGC, Caltech 101, and ASLAN. The proposed DVM model is stable with varying γ within $(10^{-4}, 10^{-3})$.

F. Analysis of the Experimental Results

On Yale, DVM performs the best in all experiments according to average classification accuracy while the performance of NFL, NFP, NFS, LSVM, and GSVM is equal to that of DVM for "10 Train." On FRGC, DVM performs the best in two experiments while SRC performs the best in two experiments. On Caltech-101 (15 Train) and ASLAN, DVM performs better than all other methods. In summary, DVM performs the best in 13 experiments out of all the 16 experiments according to average classification accuracy while SRC performs the best in three experiments and the performance of NFL, NFP, NFS LSVM, and GSVM is equal to that of DVM in one experiment. Furthermore, two-tailed paired t-test is adopted to statistically measure the difference between these methods. Here, the hypothesis is "the mean accuracy of DVM is larger than that of the other (given) method." The results of the statistical tests are reported in Table VII. We find that the performance of DVM is statistically better than that of other algorithms on the specific metric (based on two-tailed paired t-test at 5% significance level) in most cases. In summary, the overall performance of DVM is better than that of all other methods.

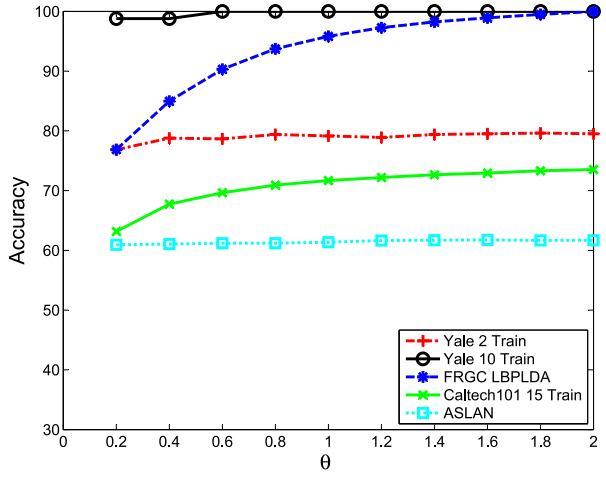


Fig. 8. Accuracy versus θ with β and γ fixed on Yale, FRGC, Caltech 101, and ASLAN.

V. CONCLUSION

In this paper, we propose a framework RVMs to explain several classification algorithms. To the best of our knowledge,

it is the first study to summarize the commonalities of various classifiers as RVM, allowing different algorithms to be analyzed and comprehensively compared. The study shows that the core idea of different classification algorithms is almost based on the same rule, but varies with respect to different representative vectors.

Based on this framework, we developed a new classification method called DVM, which can be viewed as a special case of the framework. Experimental evaluations demonstrated the effectiveness of DVM by comparing with other popular classification methods, such as SVMs, nearest NN, and SRC.

It should be noted that RVMs currently cannot be used to interpret all classifiers such as Naive Bayes, random forest [62], and the classifier fusion approach based on upper integral [63]. This deserves further study.

The employment of kernel [64] and tensor [65] techniques in SRC enables the introduction of the kernel and tensor trick in DVM. Developing the kernelized and tensorized form of DVM is an issue well worth studying.

In the framework of RVMs, effective and efficient classification methods can be developed for specific requests. Additional case studies will be valuable.

Note that the representative vector framework is a flexible framework. We can use l_2 distance, l_1 distance, Gaussian kernel, and any other arbitrary similarity measures, such as Mahalanobis distance. However, the selection of an appropriate similarity measure for different applications is still an unsolved problem. This is also a thought-provoking direction for future study.

REFERENCES

[1] J. Yu, Y. Rui, Y. Y. Tang, and D. Tao, "High-order distance-based multiview stochastic learning in image classification," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2431–2442, Dec. 2014.

[2] L. Zhang, X. Zhen, and L. Shao, "Learning object-to-class kernels for scene classification," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3241–3253, Aug. 2014.

[3] D. Tao, L. Jin, Y. Wang, and X. Li, "Person reidentification by minimum classification error-based KISS metric learning," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 242–252, Feb. 2015.

[4] L. Shao, L. Liu, and X. Li, "Feature learning for image classification via multiobjective genetic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1359–1371, Jul. 2014.

[5] S. Yan, X. Xu, D. Xu, S. Lin, and X. Li, "Image classification with densely sampled image windows and generalized adaptive multiple kernel learning," *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 395–404, Mar. 2015.

[6] F. Zhu and L. Shao, "Weakly-supervised cross-domain dictionary learning for visual recognition," *Int. J. Comput. Vis.*, vol. 109, nos. 1–2, pp. 42–59, 2014.

[7] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[8] S. Haykin, *Neural Networks and Learning Machines*. Harlow, U.K.: Pearson, 2008.

[9] P. C. Rech, "Period-adding and spiral organization of the periodicity in a hopfield neural network," *Int. J. Mach. Learn. Cybern.*, vol. 6, no. 1, pp. 1–6, 2015.

[10] L. Shao, D. Wu, and X. Li, "Learning deep and wide: A spectral method for learning deep networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2303–2308, Dec. 2014.

[11] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 2000.

[12] Y. Tian, Z. Qi, X. Ju, Y. Shi, and X. Liu, "Nonparallel support vector machines for pattern classification," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1067–1079, Jul. 2014.

[13] L. Zhang and P. N. Suganthan, "Oblique decision tree ensemble via multisurface proximal support vector machine," *IEEE Trans. Cybern.*, to be published.

[14] L. V. Utkin, "A framework for imprecise robust one-class classification models," *Int. J. Mach. Learn. Cybern.*, vol. 5, no. 3, pp. 379–393, 2014.

[15] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[16] J. R. Prasad and U. Kulkarni, "Gujrati character recognition using weighted k-NN and mean χ 2 distance measure," *Int. J. Mach. Learn. Cybern.*, vol. 6, no. 1, pp. 69–82, 2015.

[17] X. Wu *et al.*, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.

[18] N. Li, G.-D. Guo, L.-F. Chen, and S. Chen, "Optimal subspace classification method for complex data," *Int. J. Mach. Learn. Cybern.*, vol. 4, no. 2, pp. 163–171, 2013.

[19] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Mach. Learn.*, vol. 29, nos. 2–3, pp. 103–130, 1997.

[20] D. J. Hand and K. Yu, "Idiot's Bayes—Not so stupid after all?" *Int. Statist. Rev.*, vol. 69, no. 3, pp. 385–398, 2001.

[21] J. R. Quinlan, *C4.5: Programs for Machine Learning*, vol. 1. San Mateo, CA, USA: Morgan Kaufmann, 1993.

[22] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 6, pp. 607–616, Jun. 1996.

[23] S. Li and J. Lu, "Face recognition using the nearest feature line method," *IEEE Trans. Neural Netw.*, vol. 10, no. 2, pp. 439–443, Mar. 1999.

[24] S. Li, K. Chan, and C. Wang, "Performance evaluation of the nearest feature line method in image classification and retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1335–1339, Nov. 2000.

[25] J. Chien and C. Wu, "Discriminant waveletfaces and nearest feature classifiers for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1644–1649, Dec. 2002.

[26] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2106–2112, Nov. 2010.

[27] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[28] Z. Lai, W. K. Wong, Y. Xu, C. Zhao, and M. Sun, "Sparse alignment for robust tensor learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1779–1792, Oct. 2014.

[29] R. Yan, L. Shao, and Y. Liu, "Nonlocal hierarchical dictionary learning using wavelets for image denoising," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4689–4698, Dec. 2013.

[30] J. Luo, C.-M. Vong, and P.-K. Wong, "Sparse Bayesian extreme learning machine for multi-classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 4, pp. 836–843, Apr. 2014.

[31] Z. Bai, G.-B. Huang, D. Wang, H. Wang, and M. Westover, "Sparse extreme learning machine for classification," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1858–1870, Oct. 2014.

[32] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2001.

[33] Q. Shi, A. Eriksson, A. van den Hengel, and C. Shen, "Is face recognition really a compressive sensing problem?" in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2011, pp. 553–560.

[34] H. Lee, R. Raina, A. Teichman, and A. Ng, "Exponential family sparse coding with applications to self-taught learning," in *Proc. Int. Joint Conf. Artif. Intell.*, Pasadena, CA, USA, 2009, pp. 1113–1119.

[35] R. He, W.-S. Zheng, and B.-G. Hu, "Maximum correntropy criterion for robust face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1561–1576, Aug. 2011.

[36] Y. Xu, D. Zhang, J. Yang, and J. Yang, "A two-phase test sample sparse representation method for use with face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1255–1262, Sep. 2011.

[37] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?," in *Proc. Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 471–478.

[38] T. N. Sainath *et al.*, "Exemplar-based processing for speech recognition: An overview," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 98–113, Nov. 2012.

[39] A. Saeb, F. Razzazi, and M. Babaie-Zadeh, "SR-NBS: A fast sparse representation based N-best class selector for robust phoneme classification," *Eng. Appl. Artif. Intell.*, vol. 28, pp. 155–164, Feb. 2014.
[40] S. Z. Li, "Face recognition based on nearest linear combinations," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Santa Barbara, CA, USA, 1998, pp. 839–844.
[41] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Madison, WI, USA, 2003, pp. 11–18.
[42] O. Kliper-Gross, T. Hassner, and L. Wolf, "The action similarity labeling challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 615–621, Mar. 2012.
[43] W. Liu, P. P. Pokharel, and J. C. Príncipe, "Correntropy: Properties and applications in non-Gaussian signal processing," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5286–5298, Nov. 2007.
[44] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2004, pp. 153–160.
[45] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.
[46] N. Mila and K. N. Miacael, "Analysis of half-quadratic minimization methods for signal and image recovery," *SIAM J. Sci. Comput.*, vol. 27, no. 3, pp. 937–966, 2005.
[47] L. F. Lamel, R. H. Kassel, and S. Seneff, "Speech database development: Design and analysis of the acoustic-phonetic corpus," in *Proc. DARPA Speech Recognit. Workshop*, 1989, pp. 100–109.
[48] A. Hidaka and T. Kurita, "Discriminant kernels based support vector machine," in *Proc. Asian Conf. Pattern Recognit.*, Beijing, China, 2011, pp. 159–163.
[49] L. Devroye and T. Wagner, "Distribution-free inequalities for the deleted and holdout error estimates," *IEEE Trans. Inf. Theory*, vol. 25, no. 2, pp. 202–207, Mar. 1979.
[50] M. K. James, R. G. Michael, and A. G. J. James, "A fuzzy K-nearest neighbor algorithm," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, no. 4, pp. 937–966, Jul./Aug. 2005.
[51] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2012.
[52] L. Devroye, "Nonparametric discrimination and density estimation," Dept. Electr. Eng., Univ. Texas, Austin, TX, USA, Tech. Rep. 183, 1976.
[53] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm
[54] S. Xiang, F. Nie, G. Meng, C. Pan, and C. Zhang, "Discriminative least squares regression for multiclass classification and feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 11, pp. 1738–1754, Nov. 2012.
[55] M. Guillaumin, J. Verbeek, and C. Schmid, "Multimodal semi-supervised learning for image classification," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 902–909.
[56] P. J. Phillips *et al.*, "Overview of the face recognition grand challenge," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, 2005, pp. 947–954.
[57] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Comput. Vis. Image Understand.*, vol. 106, no. 1, pp. 59–70, 2007.
[58] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, 2009, pp. 1794–1801.
[59] J. Wang *et al.*, "Locality-constrained linear coding for image classification," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 3360–3367.
[60] Y. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. LeCun, "Ask the locals: Multi-way local pooling for image recognition," in *Proc. Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 2651–2658.
[61] Y. Jia, C. Huang, and T. Darrell, "Beyond spatial pyramids: Receptive field learning for pooled image features," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 3370–3377.
[62] L. Wang *et al.*, "LINKS: Learning-based multi-source integration framework for segmentation of infant brain images," *NeuroImage*, vol. 108, pp. 160–172, Mar. 2015.
[63] X.-Z. Wang, R. Wang, H.-M. Feng, and H.-C. Wang, "A new approach to classifier fusion based on upper integral," *IEEE Trans. Cybern.*, vol. 44, no. 5, pp. 620–635, May 2014.
[64] S. Gao, I. Tsang, and L. Chia, "Kernel sparse representation for image classification and face recognition," in *Proc. Eur. Conf. Comput. Vis.*, Heraklion, Greece, 2010, pp. 1–14.
[65] H. Qiu, D. Pham, S. Venkatesh, W. Liu, and J. Lai, "A fast extension for sparse representation on robust face recognition," in *Proc. Int. Conf. Pattern Recognit.*, Istanbul, Turkey, 2012, pp. 1023–1027.

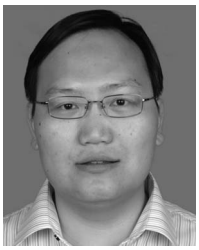
Jie Gui (M'12) received the BS degree in computer science from Hohai University, Nanjing, China, in 2004, the MS degree in computer applied technology from the Hefei Institutes of Physical Science, Chinese Academy of Sciences, Anhui, China, in 2007, and the PhD degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2010.

He is an Associate Professor with the Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, Anhui. He is a Post-Doctoral Fellow with the National Laboratory of Pattern Recognition, Institute of Automation Chinese Academy of Sciences, Beijing, China. His current research interests include machine learning, pattern recognition, and image processing.

Tongliang Liu received the BS degree in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, in 2012. He is currently pursuing the PhD degree in computer science from the University of Technology, Sydney, Ultimo, NSW, Australia.

His current research interests include machine learning, computer vision, and optimization.

Dacheng Tao (F'15) is Professor of Computer Science with the Centre for Quantum Computation and Intelligent Systems, and the Faculty of Engineering and Information Technology in the University of Technology, Sydney. He mainly applies statistics and mathematics to data analytics problems and his research interests spread across computer vision, data science, image processing, machine learning, and video surveillance. His research results have expounded in one monograph and 100+ publications at prestigious journals and prominent conferences, such as IEEE T-PAMI, T-NNLS, T-IP, JMLR, IJCV, NIPS, ICML, CVPR, ICCV, ECCV, AISTATS, ICDM; and ACM SIGKDD, with several best paper awards, such as the best theory/algorithm paper runner up award in IEEE ICDM'07, the best student paper award in IEEE ICDM'13, and the 2014 ICDM 10 Year Highest-Impact Paper Award.



Zhenan Sun (M'07) received the BS degree in industrial automation from the Dalian University of Technology, Dalian, China in 1999, the MS degree in system engineering from the Huazhong University of Science and Technology, Hubei, China, in 2002, and the PhD degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2006.

He is currently a Professor with the National Laboratory of Pattern Recognition (NLPR), CASIA. Since 2006, he has been with the NLPR of CASIA as a Faculty Member. His current research interests include biometrics, pattern recognition, and computer vision. He has authored/co-authored over 100 technical papers.

Dr. Sun is an Associated Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and the IEEE BIOMETRICS COMPENDIUM. He is a member of the IEEE Computer Society and the IEEE Signal Processing Society.

Tieniu Tan (F'03) received the BS degree in electronic engineering from Xi'an Jiaotong University, Xi'an, China, in 1984, and the MS and PhD degrees in electronic engineering from Imperial College London, London, U.K., in 1986 and 1989, respectively.

In 1989, he joined the Department of Computer Science, University of Reading, Reading, U.K., where he was a Research Fellow, a Senior Research Fellow, and a Lecturer. In 1998, he joined the National Laboratory of Pattern Recognition (NLPR), Institute of Automation of the Chinese Academy of Sciences (CAS), Beijing, China, as a Full Professor. He was the Director General of the CAS Institute of Automation from 2000 to 2007, and the Director of the NLPR from 1998 to 2013. He is currently the Director of the Center for Research on Intelligent Perception and Computing, Institute of Automation and also serves as the Deputy Secretary-General of the CAS and the Director General of the CAS Bureau of International Cooperation. His current research interests include biometrics, image and video understanding, and information forensics and security. He has published over 450 research papers in refereed international journals and conferences in the areas of image processing, computer vision, and pattern recognition, and has authored or edited 11 books. He holds over 70 patents.

Dr. Tan is an Editor-in-Chief of the *International Journal of Automation and Computing*. He has given invited talks and keynotes at many universities and international conferences, and has received numerous national and international awards and recognitions. He is a member (Academician) of the Chinese Academy of Sciences, Fellow of the World Academy of Sciences for the advancement of sciences in developing countries, an International Fellow of the U.K. Royal Academy of Engineering, and a Fellow of the International Association of Pattern Recognition.