Support Vector Selection and Adaptation for Remote Sensing Classification

Gülşen Taşkın Kaya, *Student Member, IEEE*, Okan K. Ersoy, *Fellow, IEEE*, and Mustafa E. Kamaşak

***Abstract*—Classification of nonlinearly separable data by nonlinear support vector machines (SVMs) is often a difficult task, particularly due to the necessity of choosing a convenient kernel type. Moreover, in order to get the optimum classification performance with the nonlinear SVM, a kernel and its parameters should be determined in advance. In this paper, we propose a new classification method called support vector selection and adaptation (SVSA) which is applicable to both linearly and nonlinearly separable data without choosing any kernel type. The method consists of two steps: selection and adaptation. In the selection step, first, the support vectors are obtained by a linear SVM. Then, these support vectors are classified by using the K-nearest neighbor method, and some of them are rejected if they are misclassified. In the adaptation step, the remaining support vectors are iteratively adapted with respect to the training data to generate the reference vectors. Afterward, classification of the test data is carried out by 1-nearest neighbor with the reference vectors. The SVSA method was applied to some synthetic data, multisource Colorado data, post-earthquake remote sensing data, and hyperspectral data. The experimental results showed that the SVSA is competitive with the traditional SVM with both linearly and nonlinearly separable data.**

***Index Terms*—Classification of multisource, hyperspectral and multispectral images, support vector machines (SVMs), support vector selection and adaptation (SVSA).**

I. Introduction

RECENTLY, particular attention has been dedicated to support vector machines (SVMs) for the classification of multispectral and hyperspectral remote sensing images [1]–[3]. SVMs have often been found to provide higher classification accuracies than other widely used pattern recognition techniques, such as maximum likelihood or multilayer perceptron classifiers [4]. SVMs are also competitive with other kernel-based methods such as radial basis function (RBF) neural networks and kernel Fisher discriminant analysis at a lower computational cost [5].

There have been numerous different research studies that use the SVM model in the classification problem. The classification of multisensor data sets, consisting of multitemporal synthetic aperture radar data and optical imagery, was addressed in [6]. The original outputs of each SVM discriminant function were used in the subsequent fusion process, and it was shown that the SVM-based fusion approach significantly improved the results of a single SVM. In [7], two semisupervised one-class SVM classifiers for remote sensing applications were presented. The first proposed algorithm was based on modifying the one-class SVM kernel by modeling the data marginal distribution. The second one was based on a simple modification of the standard SVM cost function.

A linear SVM (LSVM) is based on determining an optimum hyperplane that separates the data into two classes with the maximum margin [8]. The LSVM typically has high classification accuracy for linearly separable data. However, for nonlinearly separable data, LSVM has poor performance. For this type of data, a nonlinear SVM (NSVM) is preferred. The NSVM transforms the input data using a nonlinear kernel followed by the LSVM. Although the NSVM can achieve higher classification performance, it requires high computation time to map the input to a higher dimensional space by a nonlinear kernel function which is usually a fully dense matrix [9]. The computational complexity of the NSVM grows with the cube of the total number of training data.

It is well known that the major task of the NSVM approach lies in the selection of its kernel. Choosing different kernel functions produces different SVMs and may result in different performances [10]–[12]. Therefore, exploration of new techniques and systematic methodology to construct an efficient kernel function for designing SVMs in a specific application is an important research direction [13], [14]. It is also desirable to have a classifier model with both the efficiency of LSVM and the power of the NSVM. For this purpose, a mixture model combining LSVMs for the classification of nonlinear data based on divide-and-conquer strategy that partitions the input space into hyperspherical regions was proposed in [15]. The SVM-based binary decision tree classifier is another approach for this purpose [16]. Another classification algorithm for remote sensing images was developed to detect border vectors determined by using class centers and misclassified training samples, followed by their adaptation with a technique similar to the learning vector quantization (LVQ) [17].

After the selection of the kernel for the NSVM, the kernel parameters have to be adjusted for optimal performance. These parameters have a regularization effect on the cost function minimized during the training process. They may decrease the overall classification performance if they are not well chosen. Selection of the kernel parameters is empirically done by trying

Manuscript received November 10, 2009; revised March 13, 2010, October 5, 2010, and November 20, 2010; accepted November 20, 2010. Date of publication January 5, 2011; date of current version May 20, 2011. This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK).

G. Taşkın Kaya is with the Department of Computational Science and Engineering, Informatics Institute, Istanbul Technical University, Istanbul 34469, Turkey (e-mail: gulsen.taskin@be.itu.edu.tr).

O. K. Ersoy is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-2035 USA (e-mail: ersoy@purdue.edu).

M. E. Kamaşak is with the Department of Computer Engineering, Faculty of Electrical Engineering, Istanbul Technical University, Istanbul 34469, Turkey (e-mail: kamasak@itu.edu.tr).

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/TGRS.2010.2096822

a finite number of values and keeping those that provide the highest classification accuracy. This procedure requires a grid search process over the space of the parameter values. It may not be easy to locate the grid interval without *prior* knowledge of the problem. Since the optimal parameter value varies from kernel to kernel as well, kernel parameter selection might be a challenging task.

Since choosing kernel parameters is critical to the performance of the NSVM, other methodologies have also been introduced in the literature. One of them discusses an analytical criterion that is a proxy of Vapnik–Chervonenkis (VC) dimension known as the radius–margin bound. This criterion is a quadratic function of the support vector multipliers. In [18], an automatic minimization algorithm was proposed for determining the best kernel parameters. Another paper discusses an SVM model selection criterion based on the minimization of an empirical error estimate so that generalization error can drastically be reduced. The SVM hyperparameters can be adequately chosen based on this empirical error criterion [19].

In the area of hyperspectral remote sensing image classification, an optimal SVM classification system was also proposed to detect the best discriminative features and to estimate the best SVM parameters with radius-margin bound minimization by means of a genetic optimization framework [20].

In this paper, in order to overcome the mentioned drawbacks of the NSVM, a new method called support vector selection and adaptation (SVSA) is introduced with competitive performance to the NSVM [21], [22]. The proposed method has some advantages over the NSVM. In the training stage, the SVSA requires less computation time compared to NSVM, and no kernels are needed. With nonlinearly separable data, classification performance of the SVSA is competitive with NSVM. During the preliminary tests with the SVSA, it was observed that the SVSA also outperforms LSVM [23], [24].

This paper consists of six sections. The SVSA method is presented in Section II for the two-class problem. This is generalized to the multiclass problem in Section III. The computational complexities for each method in training and testing stage are presented in Section IV. The results obtained with synthetic data, multisource data, post-earthquake data, and hyperspectral remote sensing data are discussed in Section V. Conclusions are given in Section VI.

II. SVSA

Only the support vectors of the LSVM, which can be considered as the most important vectors closest to the decision boundary, are used in the SVSA. Some of the support vectors are selected based on their contribution to overall classification accuracy, and they are then called reference vectors. Afterward, they are adaptively updated by using LVQ with respect to the training data [25]. At the end of the adaptation process, the reference vectors are finalized and used in classification during testing with the 1-nearest neighbor (1NN) method [26].

Let M, N, and J denote the number of training samples, the number of features, and the number of support vectors, respectively. Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ represent the training data with $\mathbf{x}_i \in \mathbf{R}^N$, let $Y \in \mathbf{R}^M$ represent the class labels with $y_i \in \{-1, +1\}$, and let $S \in \{\mathbf{s}_1, \ldots, \mathbf{s}_J\}$ represent the support vectors with $\mathbf{s}_i \in \mathbf{R}^N$.

The LSVM is employed to obtain the support vectors (S) from the training data (X) as follows:

$$S = \left\{ \left(\mathbf{s}_j, y_{s_j}\right) \mid \left(\mathbf{s}_j, y_{s_j}\right) \in X,\ 1 \leq j \leq J \right\} \tag{1}$$

where $y_{s_j} \in \{-1, +1\}$ is the class label of the jth support vector. The number of support vectors is data dependent. If the data are linearly separable, the number of support vectors is typically 20% of the training data. If not, the number of support vectors is about 40% of the training data.

The training data set (T) is next updated to exclude the support vectors as they are used in the selection stage in order to choose some support vectors having the most contribution to classification accuracy

$$T = \{(\mathbf{t}_k, y_{t_k}) \mid (\mathbf{t}_k, y_{t_k}) \in X \setminus S,\ k = 1, \ldots, N - J\}. \tag{2}$$

The exclusion of support vectors from the training set is based on the observation that the classification accuracy is increased by excluding them in the experiments. Moreover, since the size of the training data is decreased, the computation time in the selection stage is also decreased.

In the selection stage, the support vectors in the set S are first classified by using the updated training data set T with the K-nearest neighbor (KNN) algorithm. The leave-one-out algorithm is used to determine the size of the neighborhood K for KNN classification. The result is given by

$$y_{s_j}^p = \{y_{t_l} | l = \arg\min_k \{\|\mathbf{s}_j - \mathbf{t}_k\|\}, s_j \in S, t_k \in T\} \tag{3}$$

where $y_{s_j}^p$ is the predicted class label of the jth support vector. If the original label and the predicted label of a support vector are different, then this support vector is excluded from the set of support vectors.

The remaining support vectors are called reference vectors and constitute the set R

$$R = \left\{ \left(\mathbf{r}_j, y_{r_j}\right) \mid \left(\mathbf{s}_j, y_{s_j}\right) \in S \quad \text{and} \quad y_{s_j}^p = y_{s_j} \right\}. \tag{4}$$

The reference vectors are iteratively adapted based on the training data in a way to make them more representative for classification of data by the nearest neighbor rule. The main logic of adaptation is that a reference vector causing a wrong decision should be further away from the current training vector. Similarly, the nearest reference vector with the correct decision should be closer to the current training vector.

Adaptation is achieved by the LVQ algorithm. The main idea of LVQ is to adapt the prototypes so that they more optimally represent the classes, thereby diminishing misclassification errors [27]. These prototypes result from an update procedure based on the training data set. The learning procedure consists of iteration over the training data and updating the prototypes in order to describe the optimal class boundaries.

It is assumed that $\mathbf{r}_j(\mathbf{t})$ is the nearest reference vector to $\mathbf{t}_k$ with label y_{r_w}. The adaptation is applied as follows:

$$\mathbf{r}_j(t+1) = \begin{cases} \mathbf{r}_j(t) - \eta(t)\left(\mathbf{t}_k - \mathbf{r}_j(t)\right), & \text{if } y_{t_k} \neq y_{r_j} \\ \mathbf{r}_j(t) + \eta(t)\left(\mathbf{t}_k - \mathbf{r}_j(t)\right), & \text{if } y_{t_k} = y_{r_j}. \end{cases} \tag{5}$$

It means that if the class label of the reference vector $\mathbf{r}_j$ (reference vector winner) matches the class label of the training sample $\mathbf{t}_k$, then the reference vector is moved toward $\mathbf{t}_k$.

Otherwise, it is moved away from the input sample, where $0 \leq \eta(t) \leq 1$ is the corresponding learning rate parameter given by

$$\eta(t) = \eta_0 \left(1 - \frac{t}{T}\right) \tag{6}$$

where η_0 is the initial value of η and T is the total number of learning iterations. In this step, the LVQ1 algorithm, being one of the improved algorithms of LVQ, was used [25]. In LVQ1, a single set of best matching prototypes is selected and moved closer or further away from each data vector per iteration according to whether the classification decision is correct or wrong, respectively.

The adaptation is an iterative process resulting in the reference vectors to be used for classification of the test data by the nearest neighbor rule. In the classification of test data, unseen instances are classified by using 1NN with the finalized reference vectors. Since the SVSA represents the feature space by using a small number of reference vectors, the nearest reference vectors typically have different class labels, causing $K > 1$ to yield worse results [17]. In all the experiments done, the highest classification accuracy by the SVSA was obtained with $K = 1$.

The proposed algorithm replaces the use of kernel functions by the following steps: a selection step to choose the most significant linear support vectors for classification, subsequent adaptation of the chosen linear support vectors for optimal classification performance by using the LVQ adaptation scheme, and, finally, the one nearest neighbor rule for final classification. It is known that the LVQ adaptation also maximizes the hypothesis margin, and also the sample margin since the sample margin is larger than the hypothesis margin [28]. The step of choosing the most significant linear support vectors reduces the number of reference vectors. Such reduction is also known to result in better generalization performance [29]. The SVSA also keeps the step of determining the LSVM support vectors the same. These support vectors are based on structural risk minimization and VC dimension for an LSVM.

III. SVSA: Multiclass Strategies

The original SVM method was designed for two-class problems. Since the SVSA requires support vectors obtained by LSVM, the algorithm is also based on binary classification. However, the classification of multispectral and hyperspectral remote sensing data usually involves more than two classes. In this paper, one-against-one (OAO) strategy is used to generalize the SVSA to classify multiclass data [30].

The OAO strategy, also known as "pairwise coupling," consists of constructing one SVSA for each pair of classes. Thus, for a problem with c classes, $c(c-1)/2$ SVSAs are trained to distinguish the samples of one class from the samples of another class. The final decision in the OAO strategy is based on the "winner-takes-all" rule. In other words, reference vectors obtained by training of each pairwise class give one vote to the winning class, and the sample is labeled with the class having the most votes.

In the classification of test data, all the data are classified by the reference vectors by all $c(c-1)/2$ SVSA models obtained during the OAO approach in the learning process, and $c(c-1)/2$ labels are predicted for each test data. The final label is obtained by determining the winning class between the predicted classes with the majority rule.

IV. Computational Complexity Analysis

In order to compare the computation time between the algorithms, all the codes used in the experiments have to be implemented with the same programming language because the language can affect the computational time. Otherwise, the computation time between different algorithms could lead to a wrong decision. Moreover, the code optimization is also required, which means removing the redundant script from the code, for the time comparison.

Hence, in order to make a fair comparison between the algorithms, it is necessary to either rewrite all the codes in the same language or give the computational complexity of the algorithms. Thus, instead, we present the computational complexities for the SVSA and the SVM methods.

During training, SVM needs to solve a quadratic programming (QP) problem in order to find a separation hyperplane, which has high computational complexity. In order to speed up the SVM, some decomposition methods faster than QP are introduced in the literature [31], [32]. In this way, the large QP problem is broken into a series of smaller QP problems.

The computational complexity of the SVM can change depending on linear or nonlinear kernel used in training of the SVM. The complexity degree of NSVM is higher than that of the LSVM due to the use of kernel function. The computational complexity of the LSVM is $O(N^2)$, where N is the training set size. In NSVM, the computational complexity is $O(N^3)$ because of computing the kernel function values [33].

The computational complexity of the SVSA in training is analyzed for selection and adaptation parts of the algorithm step by step as follows:

1) $O(N^2)$ for obtaining support vectors by LSVM;
2) $O(N^2 logN)$ for selection of support vectors by KNN [34];
3) $O(N' logN')$ for adaptation part in order to find the nearest reference vector to the training data randomly selected, where N' is the number of reference vectors and $N' < N$.

The step number 3) is repeated k times, which is the number of iterations, so the worst case computational complexity for this process is $O(kNlogN)$. Including all the processes, the computational complexity of the SVSA is $O(N^2 logN)$, which is much smaller than the complexity of an NSVM and is equal to $O(N^3)$.

During testing, the computational complexities for both LSVM and NSVM are $O(N)$. Since the SVSA requires sorting the distances from the reference vectors to an unclassified vector in order to find the nearest reference vector, the computational complexity of the SVSA during testing is $O(NlogN)$.

Therefore, it can be stated that the SVSA takes a longer time than the LSVM in terms of speed performance in the training stage because of selection and adaptation of support vectors in addition to obtaining support vectors. On the other hand, it requires less time than NSVM since the method does not contain time-consuming kernel processes. The advantage of the SVSA method is that the classification performance of the NSVM can be reached with faster calculations during training. In testing, the SVSA takes a bit longer time compared to LSVM and NSVM because of sorting the 1-NN process during classification.

V. Experimental Data Sets and Results

Both synthetic data and remote sensing data were used to test and to compare the proposed algorithm with the conventional SVM in terms of classification accuracy. For comparison, LSVM, NSVM with RBF kernel, and NSVM with polynomial function kernel were used. The SVMs were computed using the Library SVM (LIBSVM) library [35].

In the synthetic data experiments, synthetic data with two or more classes were created. For binary classification, four different types of synthetic data with different types of nonlinearity were generated to analyze the proposed algorithm.

In the remote sensing experiments, three different types of data were used. In the first experiment, it was of interest to see how well the SVSA method works in the classification of multisource data. For this purpose, the Colorado data set consisting of ten classes and seven features was used for classification. In the second experiment, a post-earthquake satellite image taken from Iran, Bam, was used to classify buildings as damaged and undamaged. In the third experiment, a hyperspectral image was classified by using the proposed algorithm, and the user's and producer's accuracies of the SVSA were compared to SVM's accuracies in order to check the effectiveness of the proposed approach in a high-dimensional space.

A. *Choice of Parameters*

The parameters of the LSVM related to the cost C and slack variable σ were chosen as the default parameters of the LIBSVM tool since these parameters were not so sensitive to the results obtained with all the experiments. The values of C and σ were selected as 1 and 1/(number of features), respectively.

For NSVM with RBF kernel, two parameters have to be estimated while using RBF kernels: kernel parameter γ and penalty parameter C. In order to provide the best parameters for the kernel, tenfold cross-validation was utilized in the model selection with a grid search strategy. The potential combinations of C and γ were tested in a user-defined range, where $C = 2^{-5}$ to 2^{15} and $\gamma = 2^{-15}$ to 2^{3}, and the best combinations for C and γ were selected based on the results of cross-validation.

The initial value of learning rate parameter η_0 for the adaptation part of SVSA was determined by using the tenfold cross-validation as well. The training data were randomly divided into ten groups of equal size. Then, each of the ten groups served as a validation set in turn, and the remaining data of the original training served as a new training set. Validations were carried out on these tenfold to examine the classification accuracies associated with a number of values of initial values of learning rate parameter, $\alpha_0 = 0.1{-}1.0$, with 0.1 steps, and performed for the SVSA [36].

In terms of classification performance and computational cost, scaling of data is an important preprocessing step for SVM and SVSA. Therefore, each feature of a data vector was scaled to the range between -1 and $+1$ before using the algorithms. The same method was used to scale both training and testing data [37].

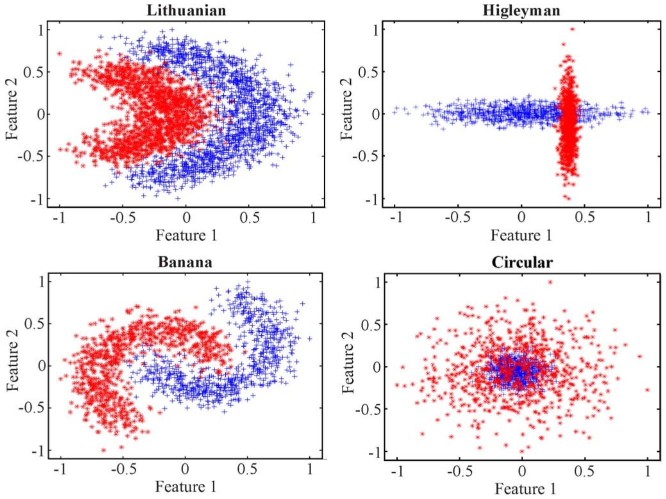


Fig. 1. Synthetic data types used in the binary classification with two features.

B. *Synthetic Data Implementations*

In these experiments, synthetic data with two and more classes and two features were created [38]. In binary classification, four different types of data with different nonlinearities were created. In the classification of multiclass problem, the data with two features and eight classes were generated. A number of data sets were generated to do averaging of the results to minimize the variance of the results and increase certainty of conclusions. For all the synthetic data implementations, training and test data were randomly created with ten realizations with random Gaussian noise.

The performance of the SVSA method was compared to those of LSVM, NSVM with radial basis kernel function (NSVM-1), and NSVM with polynomial kernel function (NSVM-2) methods in terms of class by class and overall classification accuracy (OA). The standard deviations for each method were also calculated and averaged over ten random realizations.

1) Binary Classification: For binary classification, four different types of synthetic data with different types of nonlinearity generated to analyze the proposed algorithm are visualized in Fig. 1.

With all these data, two features and two classes were used. The numbers of training and test data were also the same. Some 40% and 60% of 4000 samples for each data type were randomly grouped as training and test data, respectively. For each data type, ten realizations with random Gaussian noise were generated. All the algorithms were used to classify these ten realizations for each data type, and the classification performance of each method was based on the average classification accuracy as in Table I.

According to the experimental results provided in Table I, it is observed that the SVSA usually has better classification accuracy than the LSVM. The SVSA is also competitive with the NSVM with RBF kernel and even better than the NSVM with polynomial function kernel. Moreover, it is noted that the NSVM with different types of kernels gives different classification performances, and hence, it is required to know which type of kernel is supposed to be chosen during learning. The standard deviations of the SVSA for each data set show that the SVSA is a robust method in the classification of these data sets.

TABLE I
CLASSIFICATION OF TEST ACCURACIES FOR SYNTHETIC DATA WITH TWO FEATURES AND TWO CLASSES. OA AND STD REFER TO OVERALL ACCURACY AND STANDARD DEVIATION, RESPECTIVELY

DATASETS	CLASS	METHODS			
		LSVM	NSVM-1	NSVM-1	SVSA
Lithuanian	c_1	78.5	**97.1**	76.8	96.9
	c_2	87.5	**97.2**	88.7	96.1
	OA	83.0	**97.1**	82.8	97.0
	STD	0.5	0.2	0.6	0.3
Higleyman	c_1	71.9	88.1	71.0	**90.7**
	c_2	**100.0**	99.5	99.6	97.0
	OA	86.1	**93.9**	85.4	**93.9**
	STD	0.4	0.4	0.6	0.4
Circular	c_1	79.5	**93.1**	79.7	91.5
	c_2	31.9	74.5	28.6	**75.1**
	OA	55.5	**83.9**	54.0	83.4
	STD	0.5	0.6	0.4	0.6
Banana	c_1	84.6	96.4	79.2	**97.5**
	c_2	85.2	97.3	78.7	**97.8**
	OA	84.9	96.9	79.0	**97.7**
	STD	0.8	0.4	1.0	0.5

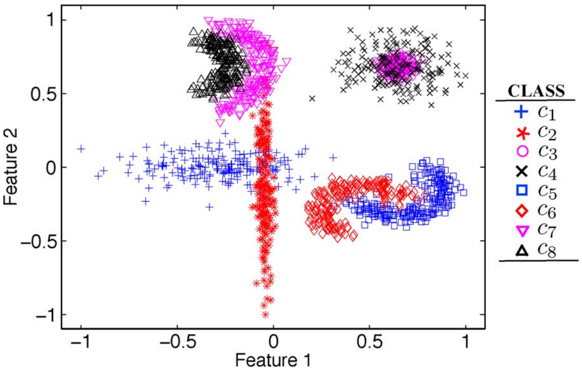


Fig. 2. Multiclass data with different type of nonlinearities.

TABLE II
NUMBER OF SAMPLES FOR TRAINING AND TEST DATA FOR EIGHT CLASS DATA WITH TWO FEATURES

CLASS	NUMBER OF SAMPLES		
	Training	Test	Total
c_1	194	290	484
c_2	202	302	504
c_3	206	309	515
c_4	185	277	462
c_5	218	328	546
c_6	211	317	528
c_7	198	296	494
c_8	187	280	467
Total	**1601**	**2399**	**4000**

TABLE III
TEST ACCURACIES FOR CLASSIFICATION OF MULTICLASS DATA. OA AND STD REFER TO OVERALL ACCURACY AND STANDARD DEVIATION, RESPECTIVELY

CLASS	METHODS			
	LSVM	NSVM-1	NSVM-2	SVSA
c_1	68.7	81.4	63.2	**90.5**
c_2	**94.2**	93.9	93.4	92.0
c_3	16.6	**96.8**	10.0	93.4
c_4	82.8	65.9	**89.9**	72.6
c_5	86.4	95.0	86.3	**97.3**
c_6	82.7	92.8	80.5	**95.3**
c_7	75.7	**95.9**	74.4	94.5
c_8	91.8	**97.2**	92.0	96.2
OA	71.2	89.8	74.1	**91.4**
STD	1.5	4.1	1.5	1.8
Time (s)	0.9	1.3	1.5	1.7

In terms of computational time, since the training and test data were not so large, the computational time spent during classification of test data for all the methods was less than 1 s. Moreover, the computational time during learning for the SVSA is less compared to that of NSVM due to estimation of optimized kernel function parameters.

In these experiments, learning rate parameter and maximum number of iterations for the SVSA were determined as 0.5 and 40.000 for all the Lithuanian, Higleyman, and Banana shaped data, and 0.8 and 40.000 for the circular data. The optimized kernel parameters of NSVM-1 for Lithuanian, Higleyman, Circular, and Banana data set were estimated as $[C, \gamma] = [8, 4]$, $[C, \gamma] = [64, 8]$, $[C, \gamma] = [0.125, 0.125]$, and $[C, \gamma] = [0.03, 8]$, respectively.

2) Multiclass Problem: For multiclass classification, a synthetic data set with eight classes and two features created from the same data used in binary classification is visualized as in Fig. 2.

The way of generating training and test data in multiclass classification is the same as in binary classification. The number of samples for training and test data used for each class is tabulated in Table II.

The average classification accuracies of all the methods on multiclass data are listed in Table III. The average standard deviation and the computational time in seconds during the classification of test data were also tabulated for each method.

In terms of overall classification performance, the SVSA yields better classification accuracy than LSVM and NSVM with RBF and polynomial function kernels in classification of the given multiclass data. The SVSA also has considerably less standard deviation compared to the NSVM-1.

We also note that LSVM classifications are done with respect to the hyperplane, whereas SVSA classifications are done with reference vectors using 1-NN. Hence, the two methods do not need to give the same accuracy even with a linear classification application.

The learning rate parameter and the maximum number of iterations for the adaptation part of the SVSA were determined as 0.5 and 40.000, respectively. The kernel parameters of NSVM-1, C and γ, were estimated as 1024 and 4, respectively.

C. Remote Sensing Data Implementations

Three different types of data sets were used in the remote sensing experiments, which are Colorado data set; Iran, Bam, post-earthquake image; and Washington DC mall hyperspectral image.

1) Colorado Data Experiment: Classification was performed with the Colorado data set consisting of the following four data sources [39]:

1) Landsat MSS data (four spectral data channels);
2) elevation data (one data channel);
3) slope data (one data channel);
4) aspect data (one data channel).

TABLE IV
TRAINING AND TEST SAMPLES OF THE COLORADO DATA SET WITH SEVEN FEATURES

CLASS	TYPE OF CLASS	NUMBER OF SAMPLES	
		Training	Test
c_1	Water	408	195
c_2	Colorado Blue Spruce	88	24
c_3	Mountane/Subalpine meadow	45	42
c_4	Aspen	75	65
c_5	Ponderosa Pine	105	139
c_6	Ponderosa Pine/Douglas Fir	126	188
c_7	Engelmann Spruce	224	70
c_8	Douglas Fir/White Fir	32	44
c_9	Douglas Fir/Ponderosa Pine/Aspen	25	25
c_{10}	Douglas Fir/White Fir/Aspen	60	39
	TOTAL	**1188**	**831**

TABLE V
TEST ACCURACIES FOR CLASSIFICATION OF COLORADO DATA. PA AND UA REFER TO PRODUCER'S AND USER'S ACCURACIES, RESPECTIVELY

Class	Accuracy	METHODS			
		LSVM	NSVM-1	NSVM-2	SVSA
c_1	*PA*	100	94.4	100	99.5
	UA	99.0	94.2	98.5	99.0
c_2	*PA*	37.5	91.7	62.5	91.7
	UA	11.4	10.7	5.50	18.0
c_3	*PA*	4.8	2.40	2.40	38.1
	UA	2.80	0.00	0.00	23.9
c_4	*PA*	33.9	36.9	20.0	41.5
	UA	45.8	25.9	23.1	34.4
c_5	*PA*	3.60	1.44	0.00	20.9
	UA	21.7	0.00	0.00	32.3
c_6	*PA*	59.0	47.3	61.2	34.6
	UA	38.3	46.6	42.1	60.7
c_7	*PA*	92.9	100.0	85.7	94.3
	UA	89.0	53.3	52.0	83.1
c_8	*PA*	0.00	0.00	0.00	0.00
	UA	0.00	0.00	0.00	0.00
c_9	*PA*	0.00	0.00	0.00	0.00
	UA	0.00	0.00	0.00	0.00
c_{10}	*PA*	20.5	69.2	0.00	64.1
	UA	21.1	0.00	0.00	39.7
Overall Accuracy		50.2	50.4	48.0	53.4

Each channel comprises an image of 135 rows and 131 columns, and all channels are spatially coregistered. There are ten ground-cover classes listed in Table IV.

One class is water; the others are forest types. It is very difficult to distinguish among the forest types using Landsat MSS data alone since the forest classes show very similar spectral responses.

The Colorado data were classified by multiclass SVSA, LSVM, and NSVM with radial basis and polynomial function kernels. The user's and producer's accuracies for each class and overall classification accuracies based on producer's accuracies for all the methods are listed in Table V.

According to the results in Table V, the overall classification performance is generally quite low for all methods since the Colorado data set represents a very difficult classification problem. The overall classification accuracy of the SVSA is better than those of all the other methods. In addition, it gives

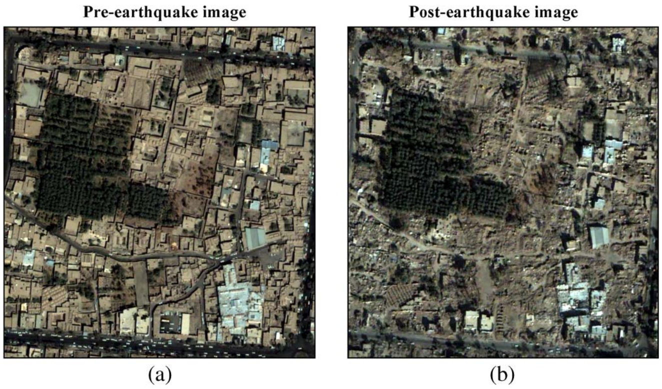


Fig. 3. Pre- and post-earthquake pansharpened images for area of interest. (a) September 30, 2003. (b) January 3, 2004.

TABLE VI
NUMBER OF TRAINING AND TEST SAMPLES WITH RESPECT TO THE CLASSES

CLASSES	Number of Samples		
	Training	Test	Total
Damage	17	23	40
Building	27	40	67
Vegetation	28	33	61
Open Ground	19	39	58
Total	**91**	**135**	**226**

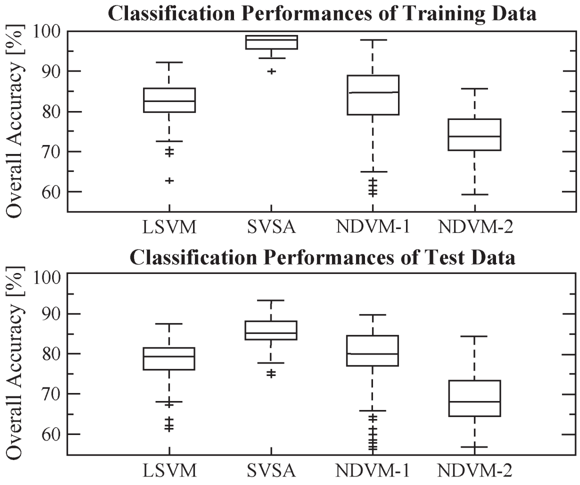


Fig. 4. Classification accuracies of the methods over the 100 realizations.

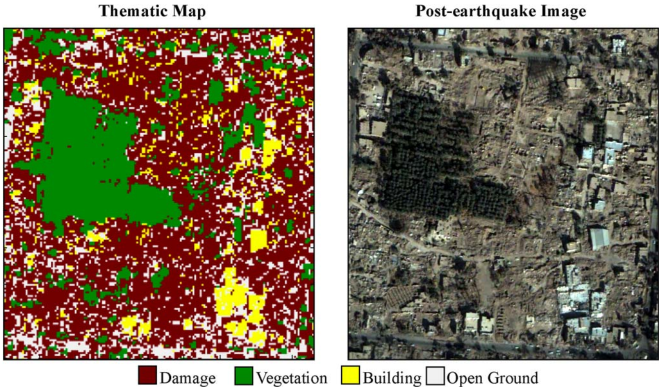


Fig. 5. Thematic map of the classes obtained by the SVSA algorithm.

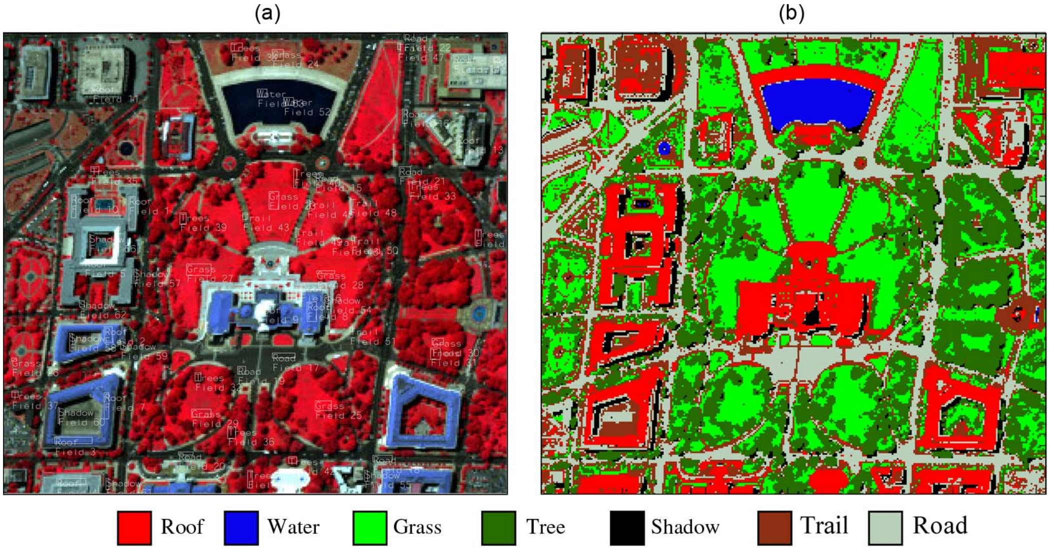


Fig. 6. (a) Seven classes were selected in hyperspectral image of Washington DC Mall. (b) The thematic map of DC Mall image obtained by SVSA.

higher classification accuracy for many classes individually in comparison to NSVMs. The computational time spent during testing is less than 1 s for all the methods due to a bit number of testing data.

The learning rate parameter and the number of maximum iterations for the adaptation part of the SVSA were determined as 0.5 and 40.000, respectively. The optimized parameters of NSVM, C and γ, with RBF were determined as 2 and 8, respectively.

2) Extraction of Damage in Iran Bam Earthquake: Quickbird satellite images of the city of Bam acquired on September 30, 2003 preearthquake and January 03, 2004 post-earthquake were obtained, and only a post-earthquake multispectral image having four spectral bands (RGB and near infrared) was used to distinguish damage patterns.

All the algorithms were evaluated using a small area of the city, approximately 8 ha in size. Since there is no ground truth available, pansharpened images with 0.6-m spectral resolution for pre- and post-earthquake images were obtained and considered as the ground-truth data (Fig. 3).

In order to obtain the pansharpened images, principle component resolution merge method using a bilinear interpolation resampling technique was used. Since the pansharpened image has high resolution as compared to the multispectral image, it was used as ground truth.

For the thematic classification, four classes were identified: damage, buildings, vegetation, and open ground. Training and test data were visually selected from the multispectral image after being verified with pansharpened pre- and post-earthquake images. The number of training and test samples for each class is tabulated in Table VI.

In the experiments, a multispectral image with 2.4-m spectral resolution was used for classification. During classification, 100 realizations with random Gaussian data added as noise were generated with data split as 40% training and 60% test data. All the methods were used to identify the four classes over the 100 samples, and the overall classification performances of each method were compared. The geometric average of classification accuracies for the algorithms and their standard deviations are shown in Fig. 4.

In this figure, it is observed that the SVSA gives the highest overall classification accuracy compared to both LSVM and NSVM. The standard deviation of the SVSA algorithm is also the smallest among all the classification algorithms, indicating the stability of the SVSA as a robust method.

The thematic map generated by the SVSA and the post-earthquake image are shown in Fig. 5. Although a pixel-based classification was performed by the SVSA, the land use classes on the thematic map were considerably identified when it is visually compared to the post-earthquake image for the area of interest.

The learning rate parameter and the number of maximum iterations for the adaptation part of the SVSA were determined as 0.4 and 40.000, respectively. The kernel parameters of NSVM-1, C and γ, were estimated as 0.25 and 8, respectively.

3) Hyperspectral Data Implementation: A higher dimensional data set captured by an airborne sensor (HYDICE) over Washington DC mall was used for exploring the performance of the SVSA in a high-dimensional space [40]. The original data set consists of 220 bands across 0.4–2.5 μm, where low signal-to-noise-ratio bands were discarded, resulting in a data set of 191 bands.

The DC data set is a challenging one to analyze since the classes are complex. There is a large diversity in the materials used in constructing rooftops, and consequently, no single spectral response is representative of the class roofs [41].

A small segment of Washington DC mall data set with 279 × 305 pixels was selected for evaluating the classification performance of the proposed approach [Fig. 6(a)].

The training and test samples shown in Table VII were collected from a reference data set which was supplied with the original data. The target land use units were seven classes used in previous studies with the DC mall data set: roofs, road, trail, grass, tree, water, and shadow [42].

The proposed method, SVSA, and all the SVM methods were used in classification of the hyperspectral data. The user's and producer's classification accuracies of all the methods for test data are tabulated in Table VIII.

According to Table VIII, the SVSA has the highest overall classification accuracy compared to the SVMs. Particularly for

TABLE VII
NUMBER OF TRAINING AND TEST SAMPLES WITH 191 FEATURES AND 7 CLASSES

CLASS	**NUMBER OF SAMPLES**		
	Training	**Test**	**Total**
Roof	419	273	692
Road	132	109	241
Grass	181	215	396
Trees	226	145	371
Trail	37	28	65
Water	30	30	60
Shadow	73	68	141
Total	**1098**	**868**	**1966**

TABLE VIII
CLASSIFICATION ACCURACIES OF TEST DATA FOR WASHINGTON DC MALL [%]. PA AND UA REFER TO PRODUCER'S ACCURACY AND USER'S ACCURACY, RESPECTIVELY

Class	**Accuracy**	**METHODS**			
		LSVM	**NSVM-1**	**NSVM-2**	**SVSA**
Roof	*PA*	99.6	91.2	99.6	98.5
	UA	94.8	97.3	94.8	97.5
Road	*PA*	100	99.08	100.0	99.1
	UA	100	100	99.1	100
Grass	*PA*	100	99.5	100	100
	UA	98.6	99.1	99.5	99.5
Trees	*PA*	100	100	100	100.
	UA	98.0	98.6	98.6	99.3
Trails	*PA*	35.7	67.9	42.9	71.4
	UA	90.9	43.2	100	76.9
Water	*PA*	100	100	100	100
	UA	100	100	100	100
Shadow	*PA*	95.6	98.5	97.1	98.5
	UA	100	100	100	100
Overall Accuracy		97.5	95.9	97.8	98.4
Time (s)		15	21	40	6

TABLE IX
NUMBER OF REFERENCE VECTORS USED IN CLASSIFICATION

Roof	**Road**	**Grass**	**Tree**	**Trail**	**Water**	**Shadow**	**Total**
4	5	9	6	8	4	9	45

the class trail, which is quite hard to distinguish due to the small number of training data, the classification accuracy of the SVSA has the highest performance as well.

In the SVSA, the classification of the hyperspectral image was carried out with the 45 reference vectors tabulated in Table IX. These results suggest that the proposed algorithm is also highly competitive in the classification of hyperspectral data and does not seem to be affected by the curse of dimensionality [43].

Since the number of reference vectors obtained by the SVSA is 45, the computational time spent by the SVSA during testing is less than that spent by all the SVMs.

The thematic map generated by the SVSA is shown in Fig. 6(b).

According to Fig. 6(b), the land use classes were considerably identified by the SVSA with quite low salt and pepper classification noise.

The learning rate parameter and the number of maximum iterations for the adaptation part of the SVSA algorithm were determined as 0.1 and 40.000, respectively. The parameters of NSVM, C and γ, were determined as 32 and 0.25, respectively.

VI. CONCLUSION

The NSVM uses kernel functions to model nonlinear decision boundaries in the original data space. It is known that the kernel type and the kernel parameters affect the shape of the decision boundaries as modeled by the NSVM and thus influence the performance of the NSVM.

In this paper, a novel SVSA method has been proposed in order to overcome the drawback of NSVM on choosing a proper kernel type and associated parameters. The proposed method is reliable for classification of both linearly separable and nonlinearly separable data without any kernel. The SVSA method consists of selection and adaptation of support vectors which most contribute to classification accuracy. Their adaptation generates the reference vectors which are next used for classification of test data by the 1NN method.

The proposed algorithm was tested with synthetic and remote sensing data with multispectral and hyperspectral channels in comparison to LSVM and NSVM algorithms. The SVSA method gave better classification results as compared to LSVM with nonlinearly separable data. The SVSA performance was also competitive with and often better than that of NSVM with both synthetic data and real remote sensing data without requiring any kernel function.

In terms of computational complexities, the SVSA is faster than NSVM during training but slightly slower than NSVM during testing. As a future work, the SVSA method during testing will be speeded up by using parallel programming.

For the remote sensing data set, it may be better to use the leave-one-out approach for training and testing since this may provide better statistics for evaluation of the system. This will also be explored in the near future.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions and also P. Zull for his feedback.

REFERENCES

[1] M. Pal and M. Mather, "Support vector machines for classification in remote sensing," *Int. J. Remote Sens.*, vol. 26, no. 5, pp. 1007–1011, Mar. 2005.

[2] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using svms and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

[3] V. Heikkinen, T. Tokola, J. Parkkinen, I. Korpela, and T. Jaaskelainen, "Simulated multispectral imagery for tree species classification using support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 3, pp. 1355–1364, Mar. 2010.

[4] C. Huang, L. S. Davis, and J. R. G. Townshend, "An assessment of support vector machines for land cover classification," *Int. J. Remote Sens.*, vol. 23, no. 4, pp. 725–749, Feb. 2002.

[5] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005.

[6] B. Waske and J. A. Benediktsson, "Fusion of support vector machines for classification of multisensor data," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 3858–3866, Dec. 2007.

[7] J. M. Mari, F. Bovolo, L. G. Chova, L. Bruzzone, and G. Camps-Valls, "Semisupervised one-class support vector machines for classification of

remote sensing data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 8, pp. 3188–3197, Aug. 2010.

[8] V. Cherkassky and F. Mulier, *Learning From Data: Concepts, Theory, and Methods*. New York: Wiley-Interscience, 1998.

[9] S. Yue, P. Li, and P. Hao, "SVM classification: Its contents and challenges," *Appl. Math.—J. Chin. Univ.*, vol. 18, no. 3, pp. 332–342, Sep. 2003.

[10] B. Scholkopf, "Support vector learning," Ph.D. dissertation, Technische Universitat Berlin, Berlin, Germany, 1997.

[11] J. Shawe-Taylor, P. L. Bartlett, R. C. Willianmson, and M. Anthony, "Structural risk minimization over data-dependent hierarchies," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1926–1940, Sep. 1998.

[12] T. Kavzoglu and I. Colkesen, "A kernel functions analysis for support vector machines for land cover classification," *Int. J. Appl. Earth Obs. Geoinformation*, vol. 11, no. 5, pp. 352–359, Oct. 2009.

[13] Y. Tan and J. Wang, "A support vector machine with a hybrid kernel and minimal Vapnik-Chervonenkis dimension," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 4, pp. 385–395, Apr. 2004.

[14] E. Blanzieri and F. Melgani, "Nearest neighbor classification of remote sensing images with the maximal margin principle," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1804–1811, Jun. 2008.

[15] Z. Fu and A. Robles-Kelly, "On mixtures of linear SVMs for nonlinear classification," in *Proc. SSPR/SPR*, 2008, pp. 489–499.

[16] H.-M. Chi and O. K. Ersoy, "Support vector machine decision trees with rare event detection," *Int. J. Smart Eng. Syst. Des.*, vol. 4, no. 4, pp. 225–242, 2002.

[17] N. G. Kasapoglu and O. K. Ersoy, "Border vector detection and adaptation for classification of multispectral and hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 3880–3893, Dec. 2007.

[18] O. Chapelle and V. N. Vapnik, "Choosing multiple parameters for support vector machines," *Mach. Learn.*, vol. 46, no. 1–3, pp. 131–159, Jan. 2002.

[19] N. E. Ayat, M. Cheriet, and C. Y. Suen, "Automatic model selection for the optimization of svm kernels," *Pattern Recognit.*, vol. 38, no. 10, pp. 1733–1745, Oct. 2005.

[20] Y. Bazi and F. Melgani, "Toward an optimal SVM classification system for hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3374–3385, Nov. 2006.

[21] G. Taskin Kaya and O. K. Ersoy, "Support vector selection and adaptation for classification of remote sensing images," Elect. Comput. Eng., Purdue Univ., West Lafayette, IN, Tech. Rep. TR-ECE-09-2, Feb. 2008.

[22] G. Taskin Kaya, O. K. Ersoy, and M. E. Kamasak, "Support vector selection and adaptation for classification of earthquake images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2009, pp. II-851–II-854.

[23] G. Taskin Kaya, O. K. Ersoy, and M. E. Kamasak, "Support vector selection and adaptation and its application in remote sensing," in *Proc. 4th Int. Recent Adv. Space Technol.*, Jun. 2009, pp. 408–412.

[24] G. Taskin Kaya, O. K. Ersoy, and M. E. Kamasak, "Support vector selection and adaptation," in *Proc. Int. Symp. Innov. Intell. Syst. Appl.*, Jun. 29–Jul. 1, 2009.

[25] T. Kohonen, J. Kangas, J. Laaksonen, and K. Torkkola, "LVQPAK: A software package for the correct application of learning vector quantization algorithms," in *Proc. IJCNN*, 1992, pp. 725–730.

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

[27] T. Kohonen, "Learning vector quantization for pattern recognition," Helsinki Univ. Technol., Espoo, Finland, Tech. Rep. TKK-F-A601, 1986.

[28] B. Hammer, M. Strickert, and T. Villmann, "On the generalization ability of GRLVQ network," *Neural Process. Lett.*, vol. 21, no. 2, pp. 109–120, Apr. 2005.

[29] K. Crammer, R. Gilad-Bachrach, A. Navot, and N. Tishby, "Margin analysis of the LVQ algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 462–469.

[30] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[31] J. X. Dong, "Fast SVM training algorithm with decomposition on very large data sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 603–618, Apr. 2005.

[32] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods—Support Vector Learning*, C. B. B. Schölkopf and A. Smola, Eds. Cambridge, MA: MIT Press, 1999.

[33] X. Liu, L. O. Hall, and K. W. Bowyer, "A parallel mixture of SVMs for very large scale problems," *Neural Comput.*, vol. 16, no. 7, pp. 1345–1351, Jul. 2004.

[34] M. Mullin and R. Sukthankar, "Complete cross-validation for nearest neighbor classifiers," in *Proc. 17th ICML*, 2000, pp. 639–646.

[35] C. C. Chang and C. Lin, LIBSVM: A Library for Support Vector Machines, 2001. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[36] H. Zhu and O. Basir, "An adaptive fuzzy evidential nearest neighbor formulation for classifying remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 8, pp. 1874–1889, Aug. 2005.

[37] C. W. Hsu, C. C. Chang, and C. J. Lin, A Practical Guide to Support Vector Classification, 2010. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[38] R. P. W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D. M. J. Tax, and S. Verzakov, *PRTools4.1, A Matlab Toolbox for Pattern Recognition*. Delft, The Netherlands: Delft Univ. Technol., 2007.

[39] J. A. Benediktsson, P. H. Swain, and O. K. Ersoy, "Neural network approaches versus statistical methods in classification of multisource remote sensing data," *IEEE Trans. Geosci. Remote Sens.*, vol. 28, no. 4, pp. 540–552, Jul. 1990.

[40] A. D. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*. Hoboken, NJ: Wiley, 2003.

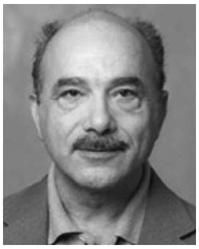
[41] X. Huang and L. Zhang, "An adaptive mean-shift analysis approach for object extraction and classification from urban hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 12, pp. 4173–4185, Dec. 2008.

[42] B. C. Kuo and K. Y. Chang, "Feature extractions for small sample size classification problem," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 3, pp. 756–764, Mar. 2007.

[43] J. A. Gualtieri and R. F. Cromp, Support Vector Machines for Hyperspectral Remote Sensing Classification, 1998.

Gülşen Taşkın Kaya (S'09) received the B.S. degree in geomatic engineering and the M.S. degree in computational science and engineering from Istanbul Technical University (ITU), Istanbul, Turkey, in 2001 and 2003, respectively, where she is currently working toward the Ph.D. degree in computational science and engineering.

From 2008 to 2009, she was a Visiting Scholar with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN. She is currently a Research Assistant with the Department of Computational Science and Engineering, Informatics Institute, ITU. Her current interests include pattern recognition in remote sensing, support vector machines, and neural networks.



Okan K. Ersoy (M'86–SM'90–F'00) received the B.S.E.E. degree from Robert College, Istanbul, Turkey, in 1967 and the M.S. Certificate of Engineering, M.S., and Ph.D. degrees from the University of California, Los Angeles, in 1968, 1971, and 1972, respectively.

He is currently a Professor of electrical and computer engineering with Purdue University, West Lafayette, IN. He has published approximately 250 papers and two books in his areas of research. He is the holder of four patents. His current research interests include statistical and computational intelligence, remote sensing, digital signal/image processing and recognition, transform and time–frequency methods, diffraction theory, and their applications in remote sensing, imaging, diffractive optics, management, and distant learning.

Prof. Ersoy is a Fellow of the Optical Society of America.

Mustafa E. Kamaşak received the B.S.E.E. and M.S.E.E. degrees from Bogazici University, Istanbul, Turkey, in 1997 and 1999, respectively, and the Ph.D. degree from Purdue University, West Lafayette, IN, in 2005.

He is currently an Assistant Professor with the Department of Computer Engineering, Istanbul Technical University, Istanbul, Turkey.