A Positive and Unlabeled Learning Algorithm for One-Class Classification of Remote-Sensing Data

Wenkai Li, Qinghua Guo, and Charles Elkan

***Abstract*—In remote-sensing classification, there are situations when users are only interested in classifying one specific land-cover type, without considering other classes. These situations are referred to as one-class classification. Traditional supervised learning is inefficient for one-class classification because it requires all classes that occur in the image to be exhaustively assigned labels. In this paper, we investigate a new positive and unlabeled learning (PUL) algorithm, applying it to one-class classifications of two scenes of a high-spatial-resolution aerial photograph. The PUL algorithm trains a classifier on positive and unlabeled data, estimates the probability that a positive training sample has been labeled, and generates binary predictions for test samples using an adjusted threshold. Experimental results indicate that the new algorithm provides high classification accuracy, outperforming the biased support-vector machine (SVM), one-class SVM, and Gaussian domain descriptor methods. The advantages of the new algorithm are that it can use unlabeled data to help build classifiers, and it requires only a small set of positive data to be labeled by hand. Therefore, it can significantly reduce the effort of assigning labels to training data without losing predictive accuracy.**

***Index Terms*—Biased support-vector machine (SVM) (BSVM), Gaussian domain descriptor (GDD), land cover, one-class classification, one-class SVM (OCSVM), positive and unlabeled learning (PUL), remote sensing.**

I. Introduction

REMOTE sensing has been commonly used in a wide variety of urban and environmental applications, such as monitoring land-use change, measuring water quality, and mapping vegetation [1]. Traditionally, all land types in an image are completely labeled via remote-sensing classification methods. For some applications, however, we may only be interested in a specific class without considering other land types [2], [3]. For example, if the objective of a project is to retrieve roads from remote-sensing data and to update an existing transportation system, we may not be interested in labeling forests and agricultural land in the image. This problem can be referred to as one-class classification. In other words, one-class remote-sensing classification seeks to extract a specific land-cover class from an image given only training examples of the class of interest. We refer to the specific land-cover class of interest as positive and other land classes as negative data. All the pixels to be classified are referred to as unlabeled data.

Supervised classification methods have been successfully applied in classification of remote-sensing data. However, direct applications of traditional supervised classifiers in one-class classification are problematic because traditional supervised classifiers require all classes that occur in a training image to be exhaustively labeled [4]. This will increase the classification difficulty and cost since manually labeling training data is labor intensive and time consuming, particularly when high-spatial-resolution images are used. Therefore, it is necessary to develop classifiers to discriminate the single class of interest from the other classes with incompletely labeled training data.

Different one-class classifiers have been developed for the one-class classification problem in literature. The class of interest is accepted as the target, whereas other classes are rejected as outliers, and only positive (i.e., target) data are required to train the classifier. For example, the Gaussian model assumes that the target data are derived from a Gaussian distribution that can be estimated from the training data [5]. The label of unknown data can be determined by choosing a probability threshold. However, assuming a unimodal and convex model of the data can sometimes be overly rigid and inappropriate [6]. Another commonly used one-class classifier is the one-class support-vector machine (SVM) (OCSVM) method developed by Schölkopf *et al.* [7]. Given n training points, OCSVM tries to find a hypersphere to separate the training data from the origin with maximum margin in a multidimensional space. This method has proved useful in document classification, texture segmentation, image classification, and ecological modeling [3], [4], [8]–[10]. A drawback of this method is that its outcome is sensitive to free parameters that are difficult to tune [8].

Aside from the labeled samples, unlabeled samples also provide useful information for the construction of classifiers [11]. Recent studies that combine both labeled and unlabeled data for classifier training show promise in one-class classification [12]–[14]. Basically, the training set includes a small set of labeled data and a large set of unlabeled data. Learning algorithms applicable to labeled and unlabeled data appear increasingly in literature [11], [13]–[17]. One family of these methods involves building classifiers such as SVMs iteratively until reaching defined convergence criteria [18]–[20]. Some of these methods have been successfully applied in remote-sensing one-class classifications. For example, the transductive SVM (TSVM) was introduced to classify Landsat 5 Thematic Mapper and hyperspectral data [20]–[22]. Previous research also pointed out that semisupervised SVMs with composite kernel functions for simultaneously taking into account spectral

Manuscript received October 16, 2009; revised February 28, 2010 and June 17, 2010; accepted June 26, 2010. This work was supported in part by the National Science Foundation (BDI 0742986) and UC/National Lab grant.

W. Li and Q. Guo are with the Sierra Nevada Research Institute, School of Engineering, University of California at Merced, Merced, CA 95343 USA (e-mail: qguo@ucmerced.edu).

C. Elkan is with the Department of Computer Science and Engineering, University of California at San Diego, La Jolla, CA 92093-0404 USA.

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.2010.2058578

and spatial information can also increase the classification accuracy [23]. A novel context-sensitive semisupervised SVM classifier was proposed to address classification problems where mislabeled patterns exist in the training set [24]. In [25], the authors used a multiobjective genetic SVM approach for image data classification. Gómez-Chova *et al.* [26] proposed a semisupervised method that combined unsupervised clustering, mean map kernel, composite kernel, and SVM together to mitigate the sample selection bias problem in remote-sensing data classifications. Other semisupervised methods, such as graph-based methods [27], semisupervised SVM based on cluster kernels [28], semisupervised kernel-based fuzzy C-means algorithm [29], Laplacian SVM [30], and weighted unlabeled sample SVM [31] have also been applied in the context of remote-sensing classification.

Although existing semisupervised learning methods usually provide good performance by incorporating unlabeled data into the training set, their shortcomings, including too many free parameters and complicated model selection procedures, can preclude their adoption by the nonexpert user [28]. For example, the number of iterations for TSVM is difficult to define [21]. In addition, most of these methods still required labeled negative examples in their training set. Although previous research has well established that unlabeled samples can improve classification accuracy when labeled samples are both positive and negative, it has not been well established previously that unlabeled samples can improve classification accuracy when labeled samples are "only" positive. Learning methods that use only positive and unlabeled data require more studies in remote-sensing one-class classification scenarios. The biased SVM (BSVM) [12] is a state-of-the-art learning algorithm from only positive and unlabeled data [13], [32]. The unlabeled set is regarded as weighted positive and weighted negative data during BSVM training. However, no direct approach is provided to set up the weights, and the trial-and-error approach usually takes long computation time [13].

Recently, Elkan and Noto have proposed a new positive and unlabeled learning (PUL) algorithm that has good potential in one-class classification [13]. This algorithm does not need labeled negative data in the training set and has shown promise in document classification. However, its application in remote-sensing classification has not been studied. Hence, we investigate the proposed PUL algorithm for one-class classification of remote-sensing data. To evaluate the performance of the new algorithm, it was applied to classify data extracted from two scenes of a high-spatial-resolution image with the assumption that only positive data are available for training. The specific objective of this paper is to evaluate the effectiveness of this algorithm in one-class classification of remote-sensing data.

II. Training Classifier From Positive and Unlabeled Data

In this paper, the target class of interest is referred to as the positive class $(y=1)$, whereas all other classes are referred to together as the negative class $(y=-1)$. Let x be a pixel; we refer to x as labeled $(s=1)$ if its class is explicitly known, and it is unlabeled $(s=0)$ if its class is unknown. Note that $y \in \{1,-1\}$ denotes the class of the pixel (positive or negative), whereas $s \in \{1,0\}$ denotes whether a pixel is assigned a label or not.

We aim to estimate the function $f(x)=p(y=1|x)$ from the finite training data. Binary classifiers, such as neural networks and SVM, can learn the function $f(x)$ directly if both positive data $\langle x, y=1\rangle$ and negative data $\langle x, y=-1\rangle$ are available in the training set $\langle x, y\rangle$. In one-class classification, however, only a set of labeled positive examples $\langle x, s=1\rangle$, and a set of unlabeled examples $\langle x, s=0\rangle$ are available in the training set $\langle x, s\rangle$. Hence, the function $f(x)$ cannot be estimated directly from the training set $\langle x, y\rangle$. Nevertheless, Elkan and Noto [13] proved a lemma indicating that $f(x)$ can be estimated indirectly from the training set $\langle x, s\rangle$.

Because only positive examples are labeled, we can infer the following: that a labeled pixel must be positive ($y=1$ if $s=1$); that an unlabeled pixel can be either positive or negative ($y=1$ or $y=-1$ if $s=0$); and that the probability of a negative pixel x being labeled is zero, as stated in

$$p(s=1|x, y=-1)=0. \tag{1}$$

A "selected-completely-at-random" assumption is required: The labeled examples are chosen completely randomly from all positive examples. In other words, if $y=1$, the probability that a positive example is labeled is the same constant regardless of x, as stated in

$$p(s=1|x, y=1)=p(s=1|y=1)=c \tag{2}$$

where c is the constant probability that a positive example is labeled. Therefore, a training set that consists of the "labeled" $(s=1)$ and "unlabeled" $(s=0)$ pixels is a random sample that satisfies (1) and (2). If we train a binary classifier with the training set $\langle x, s\rangle$, we can obtain a classifier $g(x)$ such that $g(x)=p(s=1|x)$ approximately.

With (2), we have

$$\begin{aligned} g(x) &= p(s=1|x)=p(y=1 \wedge s=1|x) \\ &= p(y=1|x)p(s=1|y=1,x) \\ &= p(y=1|x)p(s=1|y=1). \end{aligned}$$

Since we define $f(x)=p(y=1|x)$ and $c=p(s=1|y=1)$, this results in

$$f(x)=g(x)/c \tag{3}$$

which shows that the desired classifier $f(x)$ and the trained classifier $g(x)$ differ by only a constant factor [13]. If we can estimate the factor c, then $f(x)$ can be obtained.

We provide an approach to estimate the constant c on a validation set. Let V be a validation set randomly held out from the original training set $\langle x, s\rangle$. Let P be the subset of V that is labeled (and hence positive). Therefore

$$\begin{aligned} g(x) &= p(s=1|x) \\ &= p(s=1|x, y=1)p(y=1|x) \\ &\quad + p(s=1|x, y=-1)p(y=-1|x) \\ &= p(s=1|x, y=1) \times 1 + 0 \times 0 \text{ (since } x \in P) \\ &= p(s=1|y=1) \\ &= c \end{aligned}$$

which means that any single $g(x)$ from the subset P can be used to estimate c if $g(x) = p(s = 1|x)$. However, in reality, it is difficult to guarantee that $g(x) = p(s = 1|x)$ for every $x \in P$ since $g(x)$ is learned from a random finite training set. An alternative more reliable estimator of c is the average value of $g(x)$ for all $x \in P$ [13]

$$e = \frac{1}{n} \sum_{x \in P} g(x) \tag{4}$$

where e is an estimator of c and n is the cardinality of P.

In summary, the classifier $g(x)$ can be trained on only positive and unlabeled samples that satisfy the selected-completely-at-random assumption in (1) and (2). With the lemma (3), the desired classifier $f(x)$ can be obtained by dividing a constant factor into the classifier $g(x)$, where the constant factor c is estimated from a separate validation set with (4). We call this algorithm PUL. More details about this algorithm and its proof can be found in [13].

III. Experiment Design and Results

In this section, we investigate the performances of the proposed PUL for one-class classifications of remote-sensing data. BSVM is a state-of-the-art alternative learning method for the same positive/unlabeled scenario [12], [13], [32], while the Gaussian domain descriptor (GDD) and OCSVM methods are commonly used one-class classifiers [4]–[6]. Hence, they are also compared with the proposed PUL in our experiments.

A. Data-Set Description

The initial data set is a high-resolution aerial photograph acquired in 2004 by a Leica ADS40 digital camera, with 0.3-m spatial resolution. Three bands are available in the image: red (R) (610–660 nm), green (G) (535–585 nm), and blue (B) (430–490 nm). We then extracted 15 features for classification, including mean values, variance, homogeneity, contrast, and second moment of the R, G, and B bands. All features were calculated in ENVI software with a 3 × 3 pixel template and then rescaled into the range [0, 1].

In our experiments, we used appropriate sizes for the two scenes of aerial photograph so that different land types occur in the image. The first scene is an area of 350 m × 350 m with 1 366 561 pixels. The study area is located in the city of Richmond, CA [Fig. 1(a)], which includes houses, roads, trees, grasses, soils, and water. The second scene is an area of 500 m × 500 m with 2 778 889 pixels. The study area is located in the city of El Cerrito, CA [Fig. 1(b)], which includes houses, roads, trees, grasses, and soils.

In our experiments, we define the extraction of urban areas (including houses and roads), trees, grasses, soils, and water as separate examples of one-class classification. The GDD and OCSVM require only positive data for training, whereas PUL and BSVM require positive and unlabeled data for training. For all methods, both positive and negative data are required for evaluation. More labeled training data can probably result in higher accuracy but also increase the labeling effort. Hence, for each land-type extraction, we randomly labeled 5000 pixels from the aerial photograph by manual interpretation, 3000 positive (the class of interest) and 2000 negative (other classes). We also randomly selected 5000 background pixels as the unlabeled set. The training set included 1000 positive pixels, whereas the testing set included 2000 positive and 2000 negative pixels. The labeled pixels were less than 0.4% of the whole image. In order to obtain statistically reliable results, ten different random realizations of the training data were tried for each land-type classification, and the classification results were evaluated using overall accuracy (OA) and kappa coefficient (κ).

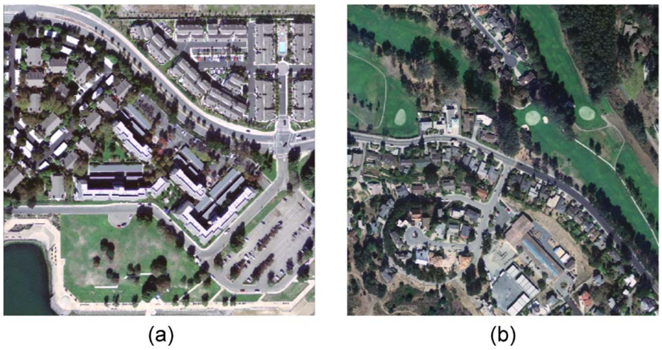


Fig. 1. Aerial photographs of study areas. (a) Scene one: Richmond, CA. (b) Scene two: El Cerrito, CA.

B. Model Development

The PUL requires a classifier that is able to estimate conditional probabilities correctly. Classifiers that can estimate conditional probabilities can be used to implement PUL. Research has shown that artificial neural networks can accurately estimate posterior probability [33]–[35]. In this paper, we used a backpropagation (BP) neural network [36] to train the classifier $g(x)$. In order to estimate meaningful probabilities, we trained the BP network with regularized mean-squared-error objective function and used the log-sigmoid transfer function so that the output fell between zero and one. Typically, a threshold of 0.5 is used to convert the output of a probabilistic classifier into binary classes. However, $g(x)$ is a probabilistic classifier trained on labeled and unlabeled data, and its relationship with the desired classifier is $f(x) = g(x)/c$. Consequently, the correct threshold is $g(x)/c = 0.5$. In order to estimate c according to (4), we split the initial training set and hold out 25% as a validation set. The output of the BP network is different depending on individual training episodes. In order to increase the model reliability, we trained the BP ten times with the same input and network structure for each case, and the output of each episode was similar and consistent; hence, they were averaged to generate the final prediction.

We implemented BSVM by the SVM$^{\mathrm{light}}$ package [37]. We held out 25% of the original training set as the validation set, which consists of only positive and unlabeled data. Without negative data, the commonly used performance measure F score cannot be calculated [38]. Alternatively, $r^2/\Pr[f(X) = 1]$ can be used as the model selection criteria, where the recall $r = \Pr[f(X) = 1|Y = 1]$ can be estimated as the proportion of correctly predicted positive data on the positive data in the validation set, and $\Pr[f(X) = 1]$ can be estimated as the proportion of predicted positive data on the whole validation set [12],

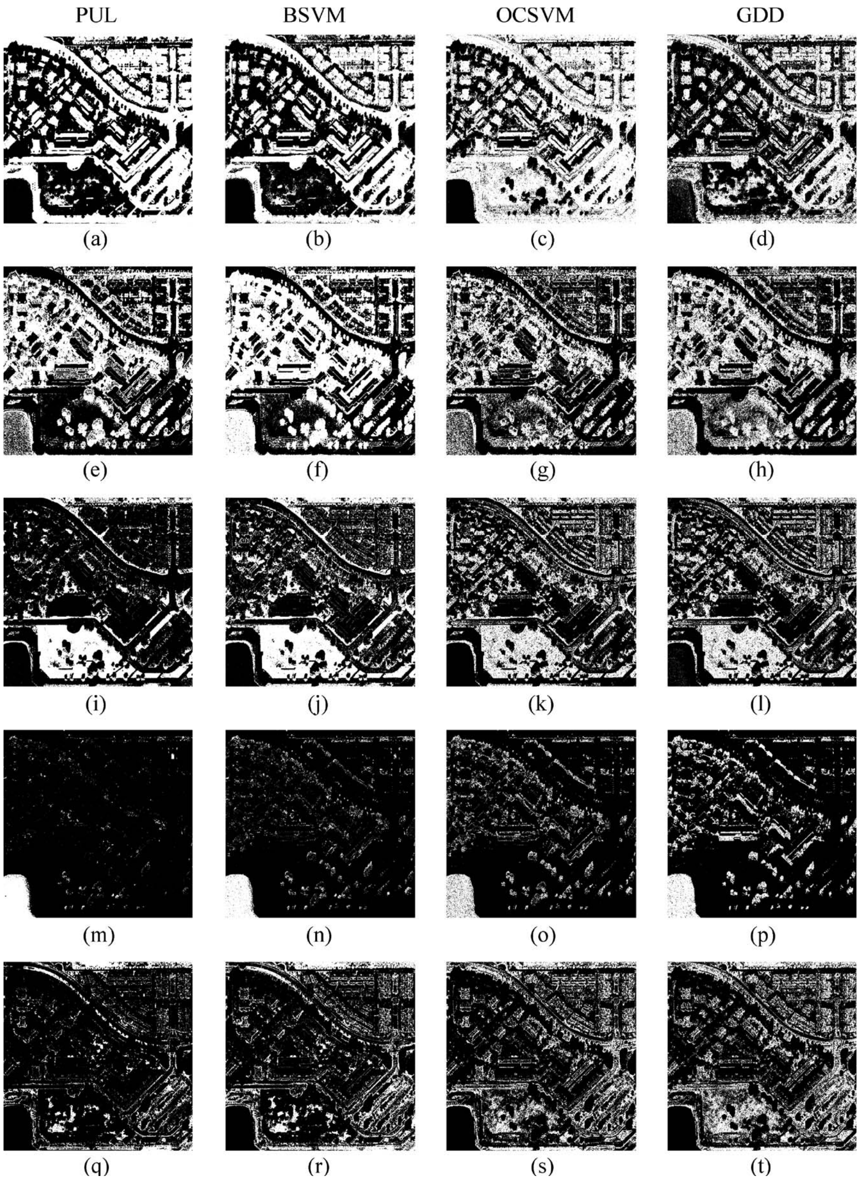
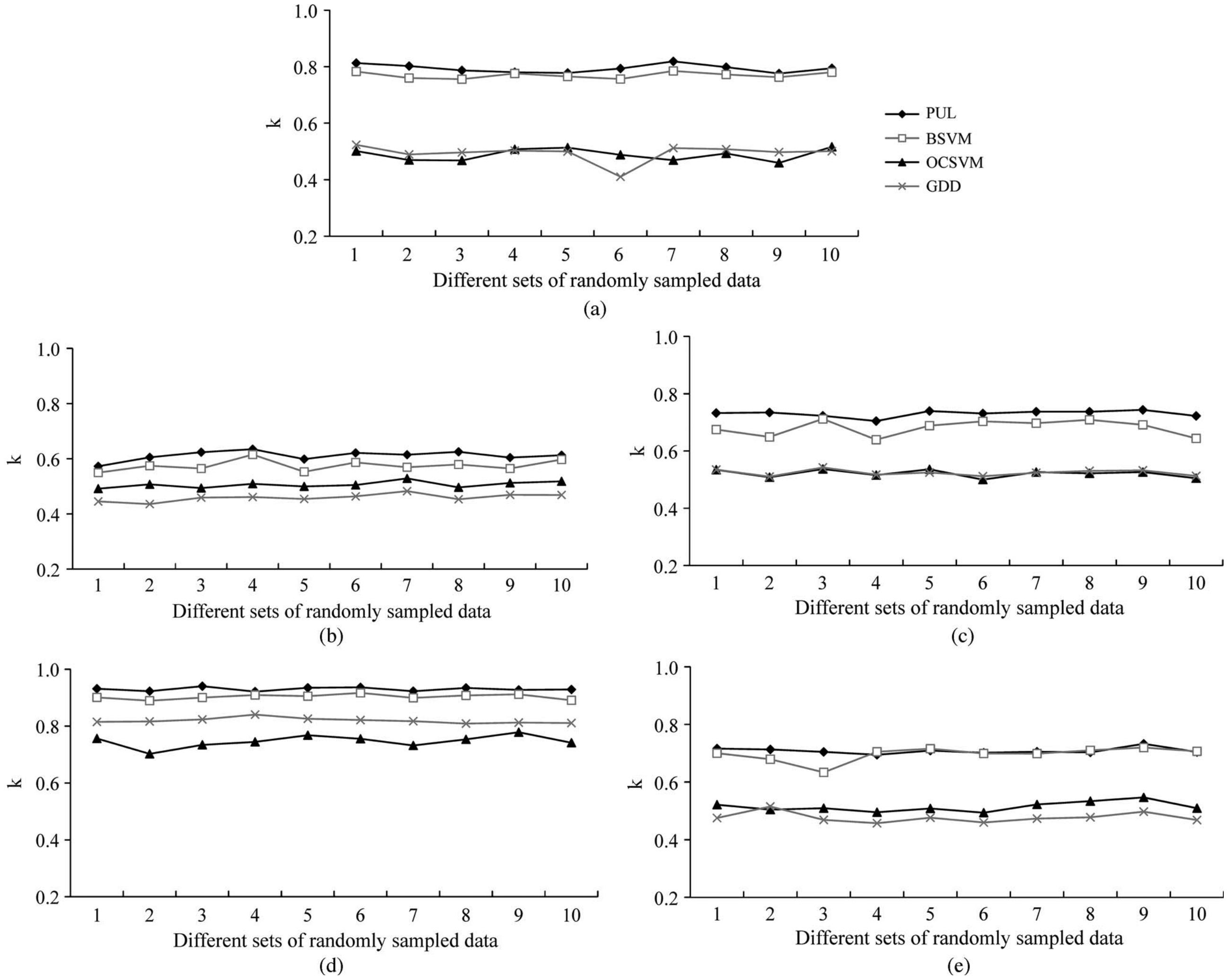


Fig. 2. Prediction maps of each land type (Scene one). (a)–(d) Urban. (e)–(h) Tree. (i)–(l) Grass. (m)–(p) Water. (q)–(t) Soil. White: positive; black: negative.

[38]. We used the Gaussian radial basis function (RBF) kernel and followed the empirical approach in [12] and the guideline in [39] to tune three parameters: $c = 2^{-7}, 2^{-6}, \ldots, 2^{0}$; $j = 2^{3}, 2^{4}, \ldots, 2^{8}$; and RBF kernel width $\gamma = 2^{-4}, 2^{-2}, \ldots, 2^{10}$. Here, $c = C_{-}$ and $j = C_{+}/C_{-}$, where C_{+} weights positive errors and C_{-} weights negative errors [12]. After c, j, and γ were selected, we then trained the model again using the original training set and generated the final predictions.

We implemented OCSVM by a library for SVMs—LIBSVM developed by Chang and Lin [40]. Only labeled samples (and hence positive) were used to train the classifier, with unlabeled samples being discarded. We used a Gaussian RBF kernel function. The output of OCSVM is binary (positive and negative); hence, no threshold is required (or the threshold is zero). Two parameters that normally need to be tuned in OCSVM are the RBF kernel width and the rejection fraction. In this paper, we tuned the RBF kernel width γ in the range (0,1000] with step of 0.1 and the rejection fraction ν in the range (0, 1) with step of 0.01 [4], [39]. It should be noted that our training data did not include any negative data; therefore, parameters tuned with only positive data can only report true-positive rate, and it is difficult to guarantee high accuracy when they are applied to a separate testing set that consisted of both positive and negative data. To investigate its best achievable performance, we trained OCSVM using the whole training set and tuned the parameters using the testing data set that consists of both positive and negative data.

GDD was implemented by the data description toolbox (dd_tools) [41]. Only labeled positive data were used to train the classifier. We used the simple Gaussian target distribution without any robustifying and tuned two parameters: the threshold θ in the range [0.01, 1] with step of 0.01 and the regularization parameter r in the range [0.01, 1] with step of 0.01. For the same reason as OCSVM, we trained GDD using the whole training set and tuned the two parameters using the testing data set that consists of both positive and negative data to obtain the model's best performance.



(a)

(b)

(c)

(d)

(e)

Fig. 3. Comparison of kappa coefficient obtained by different classifiers (Scene one). (a) Urban. (b) Tree. (c) Grass. (d) Water. (e) Soil.

C. Results

Scene One: Fig. 2 shows the classification maps of Scene one for each land type. In general, PUL provides the best classification results in the extraction of a single land type from the aerial photograph. Its prediction maps for each land type have good agreement with the original aerial photograph, particularly for the urban areas, grasses, and water. BSVM also provides relatively good results, particularly for urban areas and water, but OCSVM and GDD result in bad results, and their classification maps show more "salt-and-pepper" effect. Fig. 3 shows the comparison of κ obtained by different classifiers on ten different random realizations of the training set. PUL generally provides the highest κ, whereas the κ obtained by BSVM is lower than PUL. OCSVM and GDD have similar behavior, and they always provide the lowest κ. The behavior of OA is similar to κ and is not shown here. The mean and standard deviation of OA and κ over ten different random realizations are shown in Table I. In general, PUL obtains the highest mean values of OA and κ with relatively low standard deviations.

Scene Two: Fig. 4 shows the classification maps of Scene two for different land types. As with Scene one, the classification map for each land type obtained by PUL shows good agreement with the original aerial photograph, particularly for the urban areas and grasses. BSVM also produces relatively good classification results for each land type except for trees. By contrast, OCSVM and GDD result in worse classification maps with more salt-and-pepper effect. Because OA and κ result in similar behavior, we only show the comparison of κ obtained by different classifiers in Fig. 5, and their mean values and standard deviations of OA and κ over different random realizations are reported in Table II. In general, PUL provides the highest classification accuracies and is more stable than the other methods. For example, in the classification of urban areas, PUL provides the highest mean value of κ, which is 0.82, with a standard deviation of 0.01, whereas the mean values of κ obtained by BSVM, OCSVM, and GDD are 0.80, 0.41, and 0.42 with standard deviations of 0.01, 0.02, and 0.02, respectively.

IV. Discussion

Traditional supervised learning methods assume the availability of both positive and negative training data. However, in many applications, it is common that negative data are not available, or they are time consuming to collect. For example,

TABLE I
MEAN AND STANDARD DEVIATIONOF OA AND KAPPA COEFFICIENT (κ) ON TEST DATA (SCENE ONE)

Class	PUL		BSVM		OCSVM		GDD	
	OA(%)	κ	OA(%)	κ	OA(%)	κ	OA(%)	κ
Urban	89.71	0.79	88.49	0.77	74.43	0.49	74.71	0.49
	(0.72)	(0.01)	(0.55)	(0.01)	(1.05)	(0.02)	(1.54)	(0.03)
Tree	80.56	0.61	78.76	0.58	75.30	0.51	72.96	0.46
	(0.87)	(0.02)	(1.01)	(0.02)	(0.58)	(0.01)	(0.66)	(0.01)
Grass	86.53	0.73	84.05	0.68	76.06	0.52	76.21	0.52
	(0.57)	(0.01)	(1.38)	(0.03)	(0.67)	(0.01)	(0.54)	(0.01)
Water	96.51	0.93	95.17	0.90	87.33	0.75	90.97	0.82
	(0.32)	(0.01)	(0.44)	(0.01)	(1.06)	(0.02)	(0.46)	(0.01)
Soil	85.44	0.71	84.86	0.70	75.73	0.51	73.86	0.48
	(0.52)	(0.01)	(1.24)	(0.02)	(0.84)	(0.02)	(0.87)	(0.02)

Values in parentheses are standard deviations.

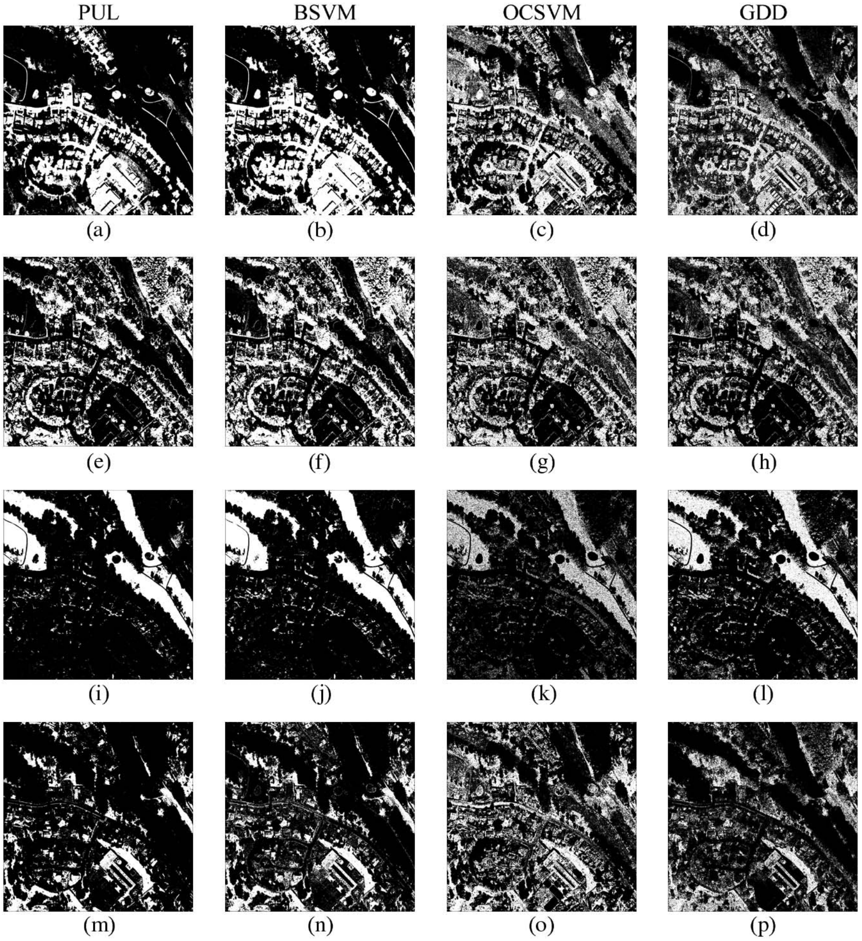


Fig. 4. Prediction maps of each land type (Scene two). (a)–(d) Urban. (e)–(h) Tree. (i)–(l) Grass. (m)–(p) Soil. White: positive; black: negative.

in one-class classification of remote-sensing data, the interest is focused on a single class, and the effort of labeling data will be significantly increased if explicit data belonging to each class in the image are required. Hence, training classifiers that do not require negative data becomes very important. In this paper, the proposed PUL algorithm proved to be successful in one-class classification of two scenes of a high-spatial-resolution image. A major advantage of the new algorithm is that it enables us to reduce the cost of labeling training data while maintaining high classification accuracy. This algorithm does not require labeled negative data in the training set, thus saving the effort of labeling training samples of other classes. Instead, only positive and unlabeled data are required. The unlabeled set probably consists of both positive and negative pixels, but it is not necessary to know their true classes. We can easily collect as many unlabeled data as desired from the background of the image.

The PUL requires the selected-completely-at-random assumption. In other words, any positive example should be

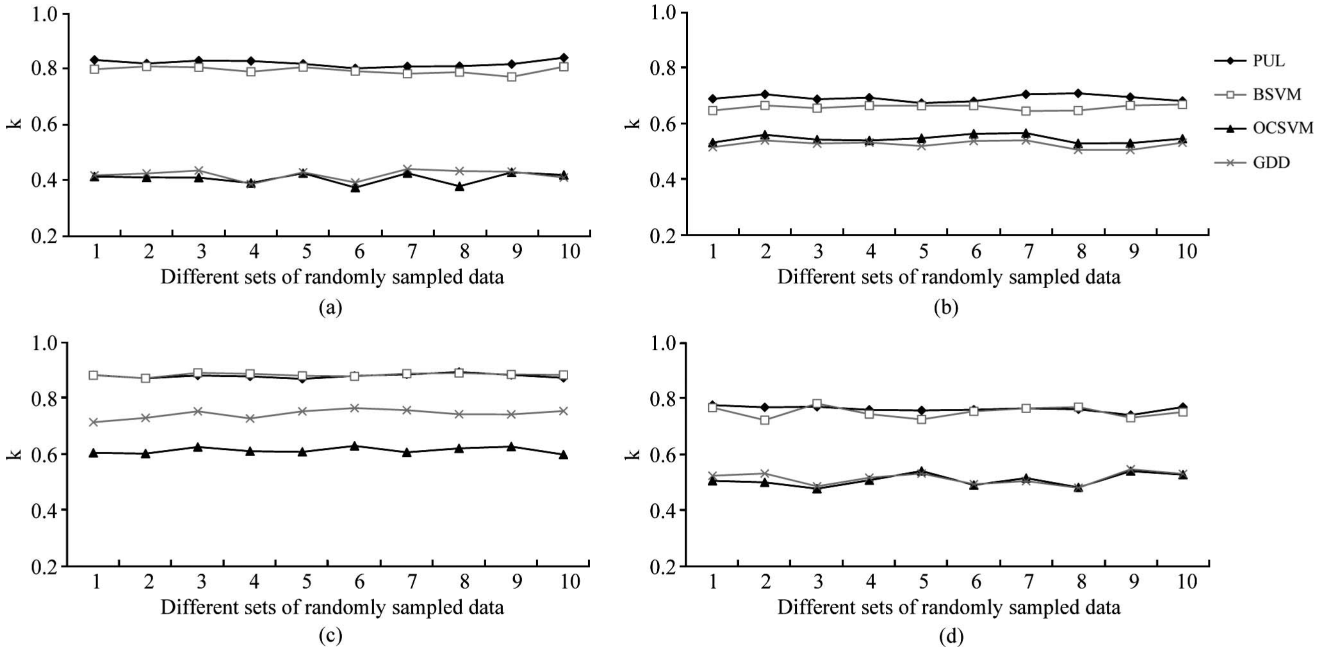


Fig. 5. Comparison of kappa coefficient obtained by different classifiers (Scene two). (a) Urban. (b) Tree. (c) Grass. (d) Soil.

TABLE II
MEAN AND STANDARD DEVIATION OF OA AND KAPPA COEFFICIENT (κ) ON TEST DATA (SCENE TWO)

Class	PUL		BSVM		OCSVM		GDD	
	OA(%)	κ	OA(%)	κ	OA(%)	κ	OA(%)	κ
Urban	91.04	0.82	89.77	0.80	70.34	0.41	70.97	0.42
	(0.60)	(0.01)	(0.62)	(0.01)	(1.00)	(0.02)	(0.92)	(0.02)
Tree	84.64	0.69	82.98	0.66	77.33	0.55	76.33	0.53
	(0.59)	(0.01)	(0.45)	(0.01)	(0.69)	(0.01)	(0.66)	(0.01)
Grass	94.00	0.88	94.19	0.88	80.69	0.61	87.21	0.74
	(0.36)	(0.01)	(0.30)	(0.01)	(0.57)	(0.01)	(0.79)	(0.02)
Soil	88.15	0.76	87.58	0.75	75.51	0.51	75.80	0.52
	(0.49)	(0.01)	(1.01)	(0.02)	(1.12)	(0.02)	(1.11)	(0.02)

Values in parentheses are standard deviations.

labeled with the same probability. In a real-world application, however, there is no prior information on the number of positive and negative pixels in the image. Therefore, how many positive pixels should be labeled at a specific constant probability c is unknown. Alternatively, the constant probability c can be treated as the effort of labeling positive samples. A higher value of c means more effort of labeling and more positive pixels being collected. Since c is not an input to the classifier and its value can be estimated from the validation set, we can collect training samples in an easier and more practical approach: randomly selecting a small set of positive pixels as the labeled set, and a large set of background pixels as the unlabeled set. Our experimental results show that this sampling approach generates good classification results, indicating that the selected-completely-at-random assumption is satisfied by this sampling approach.

Although we used a BP network to estimate $g(x)$, the outputs of PUL and traditional BP network are quite different. The difference between the PUL and traditional BP lies in the thresholds to convert the probabilistic output into binary classes. Traditional BP used 0.5 as the threshold, so the classifier was actually trained on positive and pseudonegative data since it assumes all of the unlabeled data to be negative. By contrast, PUL used $0.5 \times c$ as the threshold, so the classifier was actually trained on labeled and unlabeled data. The success of PUL lies in its ability to estimate the constant probability c from the validation set, which is then used to adjust the threshold. The classification accuracy of traditional BP is significantly affected by how noisy its negative set is, with less noisy set and higher accuracy. If no positive samples were left in the unlabeled set and hence the negative samples became "pure" and "true," then the performances of PUL and traditional BP are supposed to be similar.

Recent research has indicated that OCSVM generally performs well in one-class classification of remote-sensing data, with classification accuracies that are relatively high, which are above 90% in some situations [3], [4]. However, there were also situations where OCSVM did not perform well, with an OA of only 77% in [4]. In this paper, our results showed that OCSVM and GDD did not perform so well, with overall accuracies of 70%–80% most of the time. Since OCSVM and GDD only use the positive data and discards the unlabeled data, the number of training data available for them is much smaller than that of PUL and BSVM. Also, the outcome of OCSVM is sensitive to its free parameters that are difficult to tune [8], [10]. In addition, outliers will also degrade the performance of OCSVM [42]. We labeled training data through manual interpretation, and thus, misclassifications were inevitable. Although we can tune the rejection fraction to control the number of rejected outliers, the true rate of outliers in the training set is always unknown. By contrast, our proposed PUL seemed not to be sensitive to outliers and parameters. More importantly, it can use the large set of unlabeled data to help the classifier training. Therefore, it is reasonable that PUL outperformed OCSVM and GDD in this paper.

It is worth noting that the labeled data are positive only, but the unlabeled data may be both positive and negative. Hence, training a one-class classifier (such as OCSVM and GDD) is similar to training a binary classifier with only positive and unlabeled data (such as PUL and BSVM) in that only samples representing the positive class (class of interest) are required to be labeled. A major difference is that training a one-class classifier does not use the unlabeled data at all. Unlabeled data can improve classification accuracy when labeled samples that are only positive is also justified in this paper. According to our experimental results, methods that use both positive and unlabeled data (PUL and BSVM) generally provide higher accuracies than methods that use only positive data (OCSVM and GDD). BSVM is one of the state-of-the-art algorithms for learning using positive and unlabeled data [13], [32], but it suffers from several drawbacks. Two important parameters (c and j) are required to tune empirically on a validation set. If other kernel type rather than linear kernel is used, more free parameters (e.g., RBF kernel width γ) are introduced. In our experiment, we run BSVM using the SVM$^{\text{light}}$ package on a Windows Server 2008 with Intel Quad-Core 2.50-GHz processors and 8-GB memory, and tuning of three parameters (c, j, and γ) took about 80 min for a single random realization. By contrast, the training time of PUL was only about 2 min for each single random realization, and its classification results are more accurate and stable than BSVM.

Our proposed PUL originates from the study of Elkan and Noto [13]. This algorithm estimates the constant probability that can then be used to adjust the threshold or reweight the unlabeled set. A reweighting approach generates similar output to the adjusting threshold, but it requires retraining the classifier and hence, is not implemented in this study. This algorithm is applied in this paper for the first time to remote-sensing one-class classifications, and it provides good classification results. The limitation of this work is that we only test the new algorithm on a high-spatial-resolution aerial photograph data set, but we expect it to have good potential in other remote-sensing one-class classification scenarios using different data sets. The reason is that the PUL is not a specific classifier but a general learning method for classifiers. All classifiers that can estimate calibrated probabilities can be used to implement PUL. Implementation of PUL with SVM was investigated in [13]. In this paper, we show that the BP network is also able to be used with PUL. Hence, PUL is very flexible and can use appropriate classifiers in different scenarios. One limitation of the proposed PUL is that the selected-completely-at-random assumption is difficult to satisfy and/or verify in some applications. Although the random-sampling assumption is necessary for many remote-sensing classifiers and accuracy assessments [43]–[45], future research is needed to study the effect of biased samples on performances of PUL and strategies to debias the training samples.

V. Conclusion

In this paper, the proposed algorithm, learning from positive and unlabeled data (also called PUL), has been successful in one-class classification of high-spatial-resolution image data. The PUL algorithm estimates the constant probability of being labeled c accurately, which is then used to calibrate the threshold for generating binary predictions. In general, PUL provides the best results compared with BSVM, OCSVM, and GDD. The advantage of this algorithm is that it can use unlabeled data to help build classifiers and requires only a small set of positive data to be labeled. Therefore, it can significantly reduce the cost of labeling training data without losing accuracy.

Acknowledgment

The authors would like to thank Prof. B. Liu and Prof. C.-J. Lin who provided guidelines on the BSVM and LIBSVM, J. He and J. Zhu for their help on manual classification, and the reviewers for their constructive comments that helped strengthen this paper.

References

[1] J. R. Jensen, *Remote Sensing of the Environment: An Earth Resource Perspective*. Englewood Cliffs, NJ: Prentice-Hall, 2006.

[2] J. Byeungwoo and D. A. Landgrebe, "Partially supervised classification using weighted unsupervised clustering," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 2, pp. 1073–1079, Mar. 1999.

[3] G. M. Foody, A. Mathur, C. Sanchez-Hernandez, and D. S. Boyd, "Training set size requirements for the classification of a specific class," *Remote Sens. Environ.*, vol. 104, no. 1, pp. 1–14, Sep. 2006.

[4] J. Munoz-Marf, L. Bruzzone, and G. Camps-Vails, "A support vector domain description approach to supervised classification of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 8, pp. 2683–2692, Aug. 2007.

[5] D. M. J. Tax, "One-class classification, concept-learning in the absence of counter-examples," Ph.D. dissertation, Delft Univ. Technol., Delft, The Netherlands, 2001.

[6] C. Sanchez-Hernandez, D. S. Boyd, and G. M. Foody, "One-class classification for mapping a specific land-cover class: SVDD classification of Fenland," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 4, pp. 1061–1073, Apr. 2007.

[7] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[8] L. M. Manevitz and M. Yousef, "One-class SVMs for document classification," *J. Mach. Learn. Res.*, vol. 2, pp. 139–154, Dec. 2001.

[9] D. M. J. Tax and R. P. W. Duin, "Uniform object generation for optimizing one-class classifiers," *J. Mach. Learn. Res.*, vol. 2, pp. 155–173, Dec. 2002.

[10] Q. Guo, M. Kelly, and C. H. Graham, "Support vector machines for predicting distribution of Sudden Oak Death in California," *Ecol. Model.*, vol. 182, no. 1, pp. 75–90, Feb. 2005.

[11] V. Castelli and T. M. Cover, "The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameters," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 2102–2117, Nov. 1996.

[12] B. Liu, Y. Dai, X. Li, W. S. Lee, and P.S. Yu, "Building text classifiers using positive and unlabeled examples," in *Proc. 3rd IEEE Int. Conf. Data Mining*, 2003, pp. 179–186.

[13] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 213–220.

[14] R. Kothari and V. Jain, "Learning from labeled and unlabeled data using a minimal number of queries," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1496–1505, Nov. 2003.

[15] X. Wu, "Incorporating large unlabeled data to enhance EM classification," *J. Intell. Inf. Syst.*, vol. 26, no. 3, pp. 211–226, May 2006.

[16] R. R. Vatsavai, S. Shekhar, and T. E. Burk, "A semi-supervised learning method for remote sensing data mining," in *Proc. 17th IEEE Int. Conf. Tools Artif. Intell.*, 2005, pp. 207–211.

[17] M. M. Dundar and D. Landgrebe, "A cost-effective semi-supervised classifier approach with kernels," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 1, pp. 264–270, Jan. 2004.

[18] H. Yu, "Single-class classification with mapping convergence," *Mach. Learn.*, vol. 61, no. 1–3, pp. 49–69, Jun. 2005.

[19] H. Yu, J. Han, and K. C.-C. Chang, "PEBL: Web page classification without negative examples," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 1, pp. 70–81, Jan. 2004.

[20] L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive SVM for semisupervised classification of remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3363–3373, Nov. 2006.

[21] L. Bruzzone and M. Marconcini, "An advanced semi-supervised SVM classifier for the analysis of hyperspectral remote sensing data," in *Proc. Image Signal Process. Remote Sens. XII*, 2006, pp. 636 50Y-1–636 50Y-12.

[22] M. Chi and L. Bruzzone, "Semisupervised classification of hyperspectral images by SVMs optimized in the primal," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 6, pp. 1870–1880, Jun. 2007.

[23] M. Marconcini, G. Camps-Valls, and L. Bruzzone, "A composite semisupervised SVM for classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 2, pp. 234–238, Apr. 2009.

[24] L. Bruzzone and C. Persello, "A novel context-sensitive semisupervised SVM classifier robust to mislabeled training samples," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 7, pp. 2142–2154, Jul. 2009.

[25] N. Ghoggali, F. Melgani, and Y. Bazi, "A multiobjective genetic SVM approach for classification problems with limited training samples," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 6, pp. 1707–1718, Jun. 2009.

[26] L. Gómez-Chova, G. Camps-Valls, L. Bruzzone, and J. Calpe-Maravilla, "Mean map kernel methods for semisupervised cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 1, pp. 207–220, Jan. 2010.

[27] G. Camps-Valls, T. V. B. Marsheva, and D. Zhou, "Semi-supervised graph-based hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3044–3054, Oct. 2007.

[28] D. Tuia and G. Camps-Valls, "Semisupervised remote sensing image classification with cluster kernels," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 2, pp. 224–228, Apr. 2009.

[29] X. Liu, B. He, and X. Li, "Semi-supervised classification for hyperspectral remote sensing image based on PCA and kernel FCM algorithm," in *Proc. Geoinformatics/Joint Conf. GIS, Built Environment: Classification Remote Sensing Images*, 2008, p. 714 71I.

[30] L. Gómez-Chova, G. Camps-Valls, J. Muñoz-Marí, and J. Calpe, "Semisupervised image classification with Laplacian support vector machines," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 3, pp. 336–340, Jul. 2008.

[31] Z. Liu, W. Shi, D. Li, and Q. Qin, "Partially supervised classification-based on weighted unlabeled samples support vector machine," in *Proc. 1st Int. Conf. Adv. Data Mining Appl.*, 2005, pp. 118–129.

[32] P. Garg and S. Sundararajan, "Active learning in partially supervised classification," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1783–1786.

[33] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural Comput.*, vol. 3, no. 4, pp. 461–483, 1991.

[34] S. W. Palocsay, S. P. Stevens, and R. G. Brookshire, "An empirical evaluation of probability estimation with neural networks," *Neural Comput. Appl.*, vol. 10, no. 1, pp. 48–55, Apr. 2001.

[35] M. S. Hung, M. Y. Hu, M. S. Shanker, and B. E. Patuwo, "Estimating posterior probabilities in classification problems with neural networks," *Int. J. Comput. Intell. Org.*, vol. 1, no. 1, pp. 49–60, 1996.

[36] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Proc. Int. Joint Conf. Neural Netw.*, 1989, vol. 1, pp. 593–605.

[37] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods-Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1999.

[38] W. Lee and B. Liu, "Learning with positive and unlabeled examples using weighted logistic regression," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 448–455.

[39] C. Hsu, C. Chang, and C. Lin, "A practical guide to support vector classification," Dept. Comput. Sci. Inf. Eng., Nat. Taiwan Univ., Taipei, Taiwan, 2003.

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

[41] D. M. J. Tax, DDtools, The Data Description Toolbox for Matlab, 2009. [Online]. Available: http://ict.ewi.tudelft.nl/~davidt/dd_tools.html

[42] X. Song, G. Fan, and M. Rao, "SVM-based data editing for enhanced one-class classification of remotely sensed imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 2, pp. 189–193, Apr. 2008.

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

[44] S. V. Stehman, "Estimating the kappa coefficient and its variance under stratified random sampling," *Photogramm. Eng. Remote Sens.*, vol. 62, no. 4, pp. 401–407, Apr. 1996.

[45] G. M. Foody, "Status of land cover classification accuracy assessment," *Remote Sens. Environ.*, vol. 80, no. 1, pp. 185–201, Apr. 2002.

Wenkai Li received the B.S. degree in environmental science from Sun Yat-Sen University, Guangzhou, China, and the M.S. degree in environmental engineering from Peking University, Beijing, China, in 2005 and 2008, respectively. He is currently working toward the Ph.D. degree in the School of Engineering, University of California, Merced (UC Merced).

He is affiliated with the Sierra Nevada Research Institute, UC Merced. His research interests include climate change and terrestrial ecosystems, ecological niche modeling, and statistical learning methods.

Qinghua Guo received the B.S. degree in environmental science in 1996 and the M.S. degree in remote sensing and geographic information system (GIS) from Peking University, Beijing, China, in 1999 and the Ph.D. degree in environmental science from the University of California, Berkeley, in 2005.

Since then, he has been with the School of Engineering, University of California, Merced, as an Assistant Professor and founding faculty and is also affiliated with the Sierra Nevada Research Institute. His recent research areas include GIS and remote-sensing algorithm development and their environmental applications, such as object-based image analysis, change detection, Lidar data processing, and one-class geographic data analysis. He has developed an integrated software platform for environmental niche modeling and is Principal Investigator on several research grants funded by National Science Foundation, U.S. Forest Service, and U.S. Geological Service.

Charles Elkan received the B.S. degree from Cambridge University, Cambridge, U.K., and the Ph.D. degree from Cornell University, Ithaca, NY.

In 2005 and 2006, he was on sabbatical at the Massachusetts Institute of Technology, Cambridge, and in 1998 and 1999, he was a Visiting Associate Professor at Harvard University, Cambridge. He is currently a Professor with the Department of Computer Science and Engineering, University of California, San Diego. He is known for his research in machine learning, data mining, and computational biology. The MEME algorithm he developed with his Ph.D. student T. Bailey has been used in over 1000 publications in biology.

Dr. Elkan was the recipient of several best paper awards and data mining contests awards.