Taylor & Francis
Taylor & Francis Group

Check for updates

One-class remote sensing classification: one-class vs. binary classifiers

Xueqing Deng[a,b], Wenkai Li[a], Xiaoping Liu [a], Qinghua Guo[c] and Shawn Newsam[b]

[a]Guangdong Provincial Key Laboratory of Urbanization and Geo-simulation, School of Geography and Planning, Sun Yat-Sen University, Guangzhou, China; [b]Electrical Engineering & Computer Science, University of California at Merced, Merced, CA, USA; [c]State Key Laboratory of Vegetation and Environmental Change, Institute of Botany, Chinese Academy of Sciences, Beijing, China

ABSTRACT

Many applications of remote sensing only require the classification of a single land type. This is known as the one-class classification problem and it can be performed using either binary classifiers, by treating all other classes as the negative class, or one-class classifiers which only consider the class of interest. The key difference between these two approaches is in their training data and the amount of effort needed to produce it. Binary classifiers require an exhaustively labelled training data set while one-class classifiers are trained using samples of just the class of interest. Given ample and complete training data, binary classifiers generally outperform one-class classifiers. However, what is not clear is which approach is more accurate when given the same amount of labelled training data. That is, for a fixed labelling effort, is it better to use a binary or one-class classifier. This is the question we consider in this article. We compare several binary classifiers, including backpropagation neural networks, support vector machines, and maximum likelihood classifiers, with two one-class classifiers, one-class SVM, and presence and background learning (PBL), on the problem of one-class classification in high-resolution remote sensing imagery. We show that, given a fixed labelling budget, PBL consistently outperforms the other methods. This advantage stems from the fact that PBL is a positive-unlabelled method in which large amounts of readily available unlabelled data is incorporated into the training phase, allowing the classifier to model the negative class more effectively.

1. Introduction

Remote sensing is an important source of data for classifying the surface of the earth (Jensen 2000). One fundamental problem for which it is key is mapping land cover such as monitoring urban growth or vegetation change. Quite frequently, we are only interested in mapping a single land-cover type (Jeon and Landgrebe 1999; Foody et al. 2006; Lin et al. 2014; Liu et al. 2014; Xu et al. 2016). This is an instance of what is known in the literature as one-class remote-sensing classification since the user is only

interested in identifying one class from among those that are potentially present in a scene (Foody et al. 2006; Li, Guo, and Elkan 2011a). The class of interest is referred to as the positive class while all the other classes are collectively referred to as the negative class.

Binary and multiclass classification methods have been widely used in remote sensing. For example, maximum likelihood classifiers (MLC), neural networks, and support vector machines (SVM) are commonly applied to extract urban regions using Landsat series data (Bischof, Schneider, and Pinz 1992; Richards and Jia 1999; Kavzoglu and Mather 2003; Foody 2004; Foody and Mathur 2004; Pal and Mather 2005; Ahmad and Quegan 2012). It is possible to perform one-class classification using traditional binary classifiers (or multiclass classifiers in a two-class binary scenario). These binary classifiers, however, require a complete and exhaustively labelled sample set for model training (Munoz-Mari, Bruzzone, and Camps-Valls 2007). In particular, the training set must typically contain labelled samples of all land types that occur in the imagery, which are then divided into positive and negative sample subsets. Manually labelling such samples is labour intensive and time consuming, especially when the aim is to map a specific land cover at large scales such as continental or global (Lin et al. 2014; Wan et al. 2015). Further, this manual labelling will likely not include all land cover types, especially for complex scenes. The training set will be incomplete which will limit the accuracy of the classifier (Guo et al. 2012).

In comparison, one-class classifiers only require manually labelled samples of the class of interest – the positive samples. There are generally two approaches to one-class classification which differ on the nature of the training set: positive-only and positive-unlabelled. The positive-only methods only require positive training samples. Examples of positive-only methods include Gaussian domain descriptor (Tax 2001) and one-class SVM (OCSVM) (Schölkopf et al. 2001). The positive-unlabelled methods utilize a training set that contains unlabelled samples in addition to the labelled positive samples. The unlabelled samples can be from any class, including the class of interest, but their labels need not be known. They can therefore be obtained without any labelling effort. Examples of positive-unlabelled methods include biased SVM (Liu et al. 2003), maximum entropy methods (Phillips, Dud´lk, and Schapire 2004), the positive and unlabelled learning (PUL) algorithm (Li, Guo, and Elkan 2011a), and the presence and background learning (PBL) algorithm (Li, Guo, and Elkan 2011b). The general motivation behind these approaches is that the unlabelled samples still represent a source of useful information for classifier training (Castelli and Cover 1996), and, indeed, positive-unlabelled methods generally outperform positive-only ones (Li and Guo 2010; Li, Guo, and Elkan 2011a; Baldeck and Asner 2015). Positive-unlabelled methods have recently seen increased application to one-class classification problems in remote sensing (Li and Guo 2010; Li, Guo, and Elkan 2011a; Zhu et al. 2012; Doherty et al. 2014; Lin et al. 2014; Mack, Roscher, and Waske 2014; Baldeck and Asner 2015; Chen et al. 2016; Mack et al. 2016). For example, Wan et al. (2015) used the PUL algorithm to map US urban extents from Moderate Resolution Imaging Spectroradiometer and the Defense Meteorological Satellite Program Operational Linescan System night stable light data. And, Xu et al. (2016) applied the PBL algorithm to estimate global snow cover with space-borne microwave brightness temperature measurements.

So how should practitioners decide between using binary or one-class classifiers when faced with a one-class remote sensing classification problem? Binary classifiers have generally been preferred since they are perceived as being more accurate given appropriate training data (Li, Guo, and Elkan 2011b; Baldeck and Asner 2015; Mack et al. 2016). Yet, binary classifiers are typically more expensive to train since they require more labelled data, and their performance might be limited by incomplete training data. Practitioners are sensitive to both accuracy and cost when choosing a method. Here, we define the cost as the effort required to manually label the training samples in terms of number of pixels. In reality, practitioners can manually label either a number of random pixels or polygons from the image to create samples. While the latter approach is more efficient since each polygon contains multiple pixels, the samples are selected in a biased rather than random approach. Please note that the random sampling assumption is necessary for many supervised classifiers and accuracy assessment (Pal and Mather 2005; Foody et al. 2006). In this study, therefore, we only consider the former approach, i.e. randomly labelling pixels instead of polygons from the image, which is commonly used in the literature (Kavzoglu and Mather 2003; Foody et al. 2006; Wan et al. 2015). Meanwhile, we assume that the cost of visually labelling a sample (pixel) is identical irrespective of the class, although it might not be the case in reality.

Suppose that we want to detect urban areas at the pixel level from an image that contains urban, tree, grass, soil, and water pixels. If the cost is set at 1000 labelled pixels, then these can all be from the positive class in the case of a one-class classifier. In contrast, these 1000 pixels would need to be split between the positive class and a negative class consisting of the other four classes in the case of a binary classifier. Would a binary classifier still be more accurate than a one-class classifier in this case? Previous studies have investigated the performances of different one-class classifiers (Munoz-Mari, Bruzzone, and Camps-Valls 2007; Li, Guo, and Elkan 2011a); the one-class and binary classifiers have also been compared previously, without considering the cost of collecting samples for both approaches (Li, Guo, and Elkan 2011b; Guo et al. 2012; Mack et al. 2016). However, no studies, to our knowledge, have compared one-class and binary classifiers under the same labelling budget.

The comparison between one-class and binary classifiers is important in order to better understand the strengths and drawbacks of different methods. The major contribution of this article is a quantitative comparison of the accuracy of one-class and binary classifiers under the same labelling cost, which can provide guidance on method selection for one-class remote-sensing classification. The classification accuracies of different methods including OCSVM, PBL, MLC, backpropagation neural network (BPNN), and SVM are compared. Detailed descriptions about the methodology, comparison of methods, and discussion of the implications are provided in the following sections.

2. Methodology

2.1. *Data set*

The first evaluation data set is an aerial photograph of Contra Costa county in California (CA) obtained by a Leica ADS40 digital camera with a spatial resolution of 0.3 m,

acquired in 2004. Three bands are provided in the imagery: red (R) (610–660 nm), green (G) (535–585 nm), and blue (B) (430–490 nm). The second evaluation data set is a WorldView-3 image of the city Guangzhou, China, acquired in 2014, with 2 m spatial resolution. There are four bands available: B (450–510 nm), G (510–580 nm), R (630–690 nm), and near infrared (NIR) (770–895 nm). We consider both spectral and textural features to perform pixel-wise classification. We extract the pixel-level mean, contrast, homogeneity, and entropy separately for each band using ENVI software with a 3 × 3 pixel template. This results in a total of 12 and 16 features, respectively, for the two data sets. Each feature is then rescaled to the range [0, 1].

Figure 1 shows two scenes from the aerial photograph that we use as our study sites. Each scene measures 500 m × 500 m and contains 2,778,889 pixels. The scenes mostly contain grass, trees, buildings, roads, soils, shadows, and water. For our experiments, we consider the following six separate one-class classification problems: urban (including buildings and roads), tree, grass, shadow, soil, and water. The manually labelled ground truth used for training and testing is shown in Figure 2. A small number of pixels that are difficult to identify are masked and excluded from the evaluations. Figure 3 shows the third scene extracted from the WorldView-3 image. It measures 3000 m × 3000 m with 2,250,000 pixels, including houses, roads, trees, grass, soils, and shadows. The whole
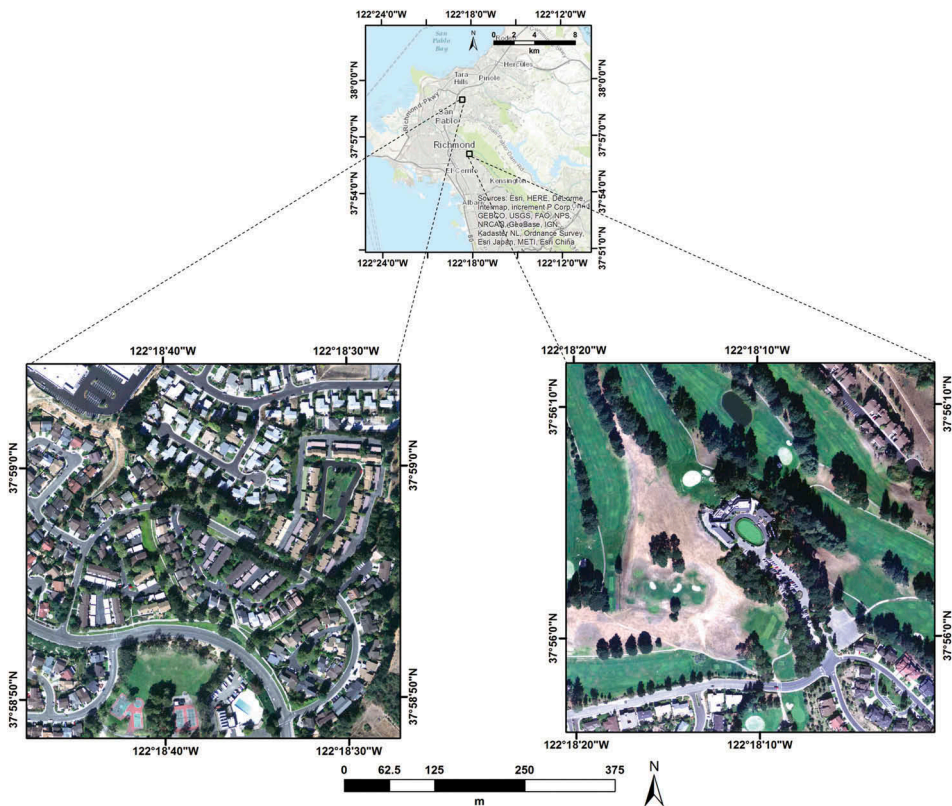

Figure 1. High-spatial-resolution aerial imagery of scenes one and two in Contra Costa county, CA.
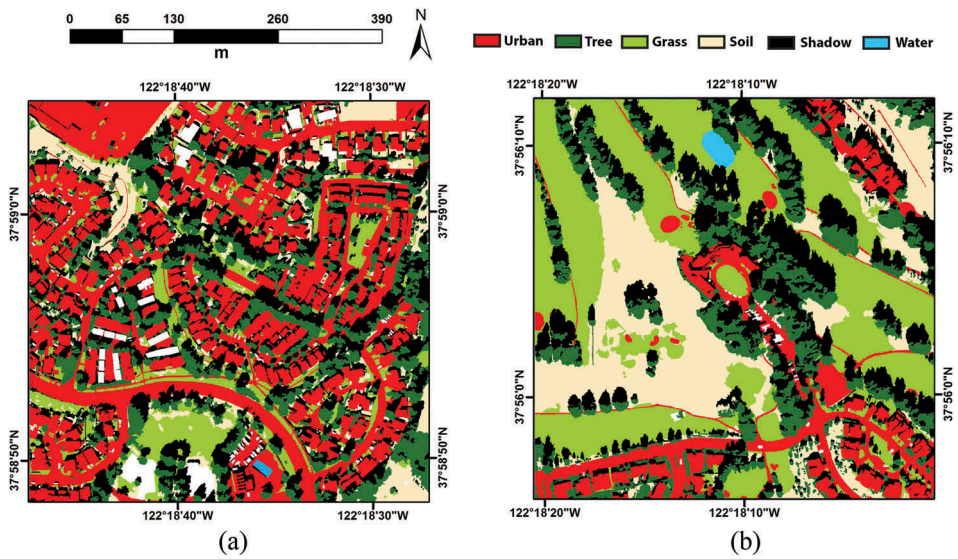
Figure 2. Manually labelled ground truth maps of the study sites: scene one (a) and scene two (b). White indicates masked pixels that are difficult to label.
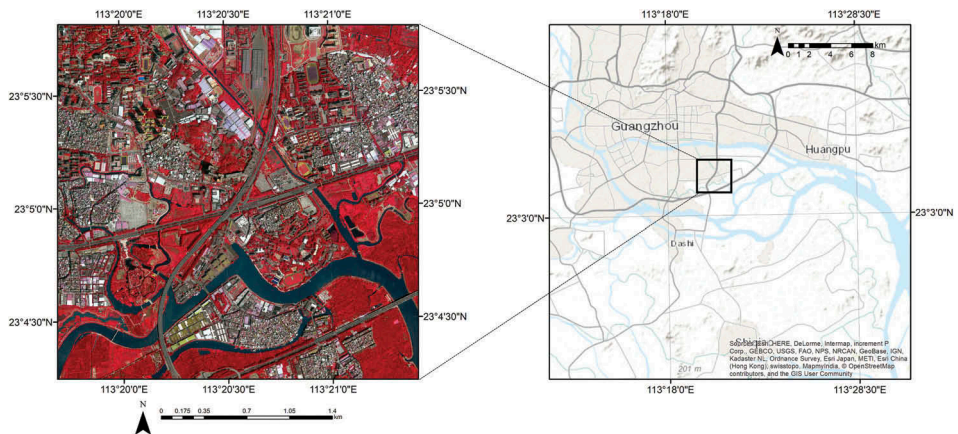

Figure 3. The WorldView-3 image shown in NIR, R, and G bands of scene three in Guangzhou, China.

ground truth map for scene three is not available, and we only consider the classification of urban areas (i.e. buildings and roads) for this scene.

As mentioned earlier, the training sets for the two categories of classification methods are different. One-class classifiers only require positive labelled data but binary classifiers require both positive and negative labelled data. The methods are compared using the same amount of labelled training data. Specifically, let N denote the number of labelled training samples per class, and k denote the number of the classes. When training the binary classifiers, the positive sample size is N and the negative sample size is the sum of the other classes, i.e. $(k - 1) \times N$. In contrast, the training set for the one-class classifiers contains only positive labelled data and so all $k \times N$ labelled samples can be from the positive class. Note that we make the reasonable assumption that unlabelled data can
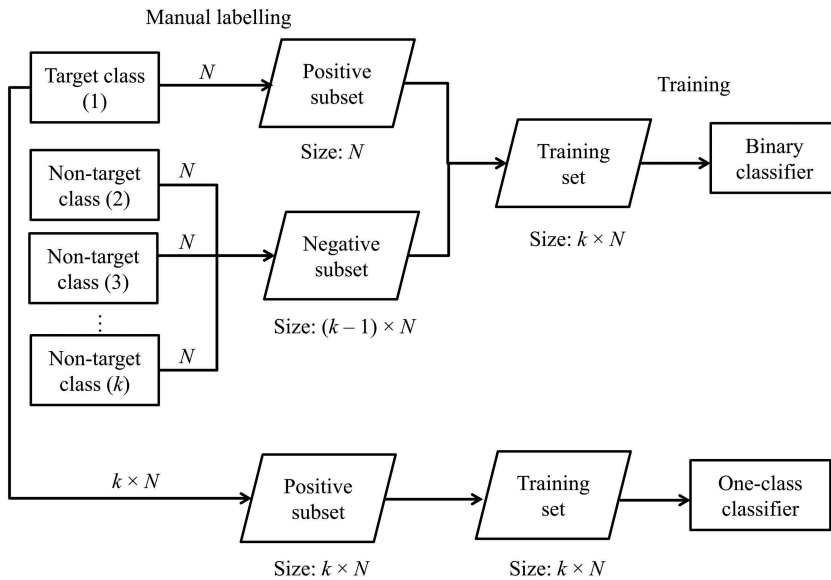
Figure 4. Flow chart of the sampling scheme for one-class and binary classifiers with equal sample cost. *k* refers to the number of land types.

be sampled from the imagery without cost (effort of manual interpretation). The training set size is thus considered just to be the total number of labelled samples (i.e. $k \times N$) in either case (Figure 4). We use manual labelling to construct training sets of sizes 200, 400, 600, 800, and 1000 pixels. We also randomly sample 5000 background pixels from the imagery as the unlabelled data for PBL (Li, Guo, and Elkan 2011b). Each experiment is performed 10 times with different random realizations of the training set in order to obtain statistically meaningful results. For scenes one and two, the classification accuracy is determined using the ground truth maps, while an independent test data set including 2000 positive and 2000 negative pixels is used for scene three. The overall accuracy (OA) and kappa coefficient (κ) are used for comparison.

2.2. One-class classification

MLC, BPNN, and SVM are selected as the binary classifiers in our evaluation since they are commonly used for remote sensing classification (Richards and Jia 1999; Foody 2004). We select OCSVM, a positive-only method, as one of the one-class classifiers since it is also commonly used (Foody et al. 2006). These are all established methods, so we do not provide detailed descriptions. We select PBL, a positive-unlabelled method, as the other one-class classifier since it can handle the case–control sampling scenario (Li, Guo, and Elkan 2011b), and has also been successfully applied in one-class remote-sensing classification (Xu et al. 2016). A brief description of PBL is provided in the following.

Let $y = 1$ denote positive data, $y = 0$ denote negative data, and $\boldsymbol{x}$ denote the vector of predictive features (or covariates) associated with a pixel. In one-class classification, we aim to model the probability of a pixel being from the positive

class conditioned on its covariates, $\Pr(y = 1 \mid \boldsymbol{x})$. Traditional supervised learning methods cannot correctly model $\Pr(y = 1 \mid \boldsymbol{x})$ without labelled negative data because the overall population prevalence $\Pr(y = 1)$ and the density of covariates for negative samples $\Pr(\boldsymbol{x} \mid y = 0)$ cannot be estimated. To address this problem, Li et al. developed the PBL algorithm that can model $\Pr(y = 1 \mid \boldsymbol{x})$ without relying on labelled negative data (Li, Guo, and Elkan 2011b).

In PBL, a different model is first trained using a data set that consists of labelled positive (denoted as $s = 1$) and unlabelled data (denoted as $s = 0$). This model is then used to calibrate the desired model. The positive and unlabelled data are sampled separately from the study site which is referred to as case–control sampling. The trained model is denoted as $\Pr(s = 1 \mid \boldsymbol{x}, \eta = 1)$, where $\eta = 1$ denotes the positive-unlabelled case–control sampling scenario. The unlabelled data may contain both positive and negative data, so the trained model $\Pr(s = 1 \mid \boldsymbol{x}, \eta = 1)$ suffers from the problem of 'contaminated controls' and hence it is not equal to the desired model $\Pr(y = 1 \mid \boldsymbol{x})$. However, Li et al. showed that the trained model and the desired model have the following relationship:

$$\Pr(y = 1 \mid \boldsymbol{x}) = \frac{1 - c}{c} \times \frac{\Pr(s = 1 \mid \boldsymbol{x}, \ \eta = 1)}{1 - \Pr(s = 1 \mid \boldsymbol{x}, \ \eta = 1)} \tag{1}$$

where c is a constant (Li, Guo, and Elkan 2011b).

Thus, in order to obtain the desired model $\Pr(y = 1 \mid \boldsymbol{x})$, we need to estimate the constant c. Let O denote a set of the 'prototypical positive' pixels. Li et al. proved that the predicted value $\Pr(s = 1 \mid \boldsymbol{x}, \eta = 1)$ of any prototypical positive pixel is equal to c (Li, Guo, and Elkan 2011b). In practice, we average the predicted probabilities of multiple prototypical positive pixels to obtain a more reliable estimator of c:

$$c = \frac{1}{n} \sum_{\boldsymbol{x} \in O} \Pr(s = 1 \mid \boldsymbol{x}, \ \eta = 1) \tag{2}$$

where n is the cardinality of O.

In summary, PBL allows us to obtain the desired model $\Pr(y = 1 \mid \boldsymbol{x})$ indirectly based on positive and unlabelled data through the following procedure. The naive model $\Pr(s = 1 \mid \boldsymbol{x}, \eta = 1)$ is first trained using positive and unlabelled data. The constant c is then estimated from prototypical positive data as shown in Equation (2). Finally, the naive model is used to derive the desired model $\Pr(y = 1 \mid \boldsymbol{x})$ using Equation (1). More details about the PBL algorithm can be found in Li, Guo, and Elkan (2011b).

2.3. *Model implementation*

We used our own Matlab (version R2011b, MathWorks) implementation of MLC (Richards and Jia 1999). Under the assumption that the distribution of each class can be represented using a multivariate normal, we can infer the conditional probability of a test sample being positive (or negative). Each multivariate normal distribution is characterized by a mean vector and a covariance matrix which are estimated from the training data. According to the maximum likelihood criterion, a test sample is classified as positive if $\Pr(y = 1 \mid \boldsymbol{x})$ is larger than $\Pr(y = 0 \mid \boldsymbol{x})$, and negative otherwise.

We implemented SVM and OCSVM using the LIBSVM SVM library developed by Chang and Lin (2015). We used Gaussian radial basis function (RBF) kernels and performed grid search to find the optimal free parameter values (Li and Guo 2014). The original training set was randomly divided into two subsets: 75% for training and 25% for validation. For SVM, the penalty parameter C and RBF kernel width γ were chosen from the following ranges: $C = 2^{-5}, 2^{-3}, \ldots, 2^{15}$; $\gamma = 2^{-15}, 2^{-13}, \ldots, 2^3$. For OCSVM, the RBF kernel width γ and rejection fraction v were chosen from $\gamma = 2^{-4}, 2^{-3}, \ldots, 2^{10}$; $v = 0.01, 0.02, \ldots, 0.5$ (Li and Guo 2014). Please note that, during training, SVM is provided with both positive and negative data, and so we can use κ to measure the cross-validation accuracy. For OCSVM, negative data are not available during training, so we used F_{pb} (F-measure based on positive and background data), an accuracy measure for one-class data, to measure the cross-validation accuracy (Li and Guo 2014).

We implemented BPNN using Matlab (version R2011b, MathWorks). We obtained a probabilistic output by using a mean-squared-error objective function and log sigmoid transfer function. The other parameters were set to default values. To improve the performance of the model, the original training set was randomly divided into two subsets: 75% for training and 25% for validation. We trained the network 10 times and the individual outputs were averaged to generate the final prediction. Since BPNN is trained using positive and negative data, the output is the posterior probability $\Pr(y = 1 \mid \mathbf{x})$. Binary classification of a test sample is achieved by applying a decision threshold of 0.5.

PBL requires that a binary classifier that can estimate posterior probability be trained on the labelled positive and unlabelled data. Since neural networks can accurately estimate posterior probability (Richard and Lippmann 1991; Hung et al. 1996; Palocsay, Stevens, and Brookshire 2001), we used the same BPNN and training procedure as above to implement PBL. This BPNN is trained using labelled positive and unlabelled data and results in the naive model $\Pr(s = 1 \mid \mathbf{x}, \eta = 1)$. The predicted values of the positive data in the training set were then averaged to estimate the constant c. Please note that the constant c can be estimated either from the training set or from the validation set, and both methods produce similar results according to our test. The naive model $\Pr(s = 1 \mid \mathbf{x}, \eta = 1)$ was used to calibrate the desired model $\Pr(y = 1 \mid \mathbf{x})$ though Equation (1). Binary classification of a test sample is achieved by applying a decision threshold of 0.5.

3. Results

Scene one: Figure 5 shows the predicted binary maps of each land type in scene one for each classifier with a training set size of 1000 pixels. Ten random realizations of the training set produce similar results so we only present one. Overall, PBL provides the best visual results that have the least 'salt-and-pepper' effects particularly for urban, grass, and shadow. SVM and BPNN provide relatively good results while OCSVM and MLC produce the worst results with obvious over-prediction of the positive classes. The quantitative evaluations are provided in Table 1 including the mean and standard deviation of OA and κ. Generally, PBL results in the best performance (highest values of OA and κ), followed by BPNN and SVM, whereas OCSVM and MLC perform the worst. The average κ values over 10 random realizations of the training sets with different sample sizes are shown in Figure 6. The κ value for PBL is relatively stable over different sample sizes, and is the highest among different methods. The κ values for SVM, OCSVM,
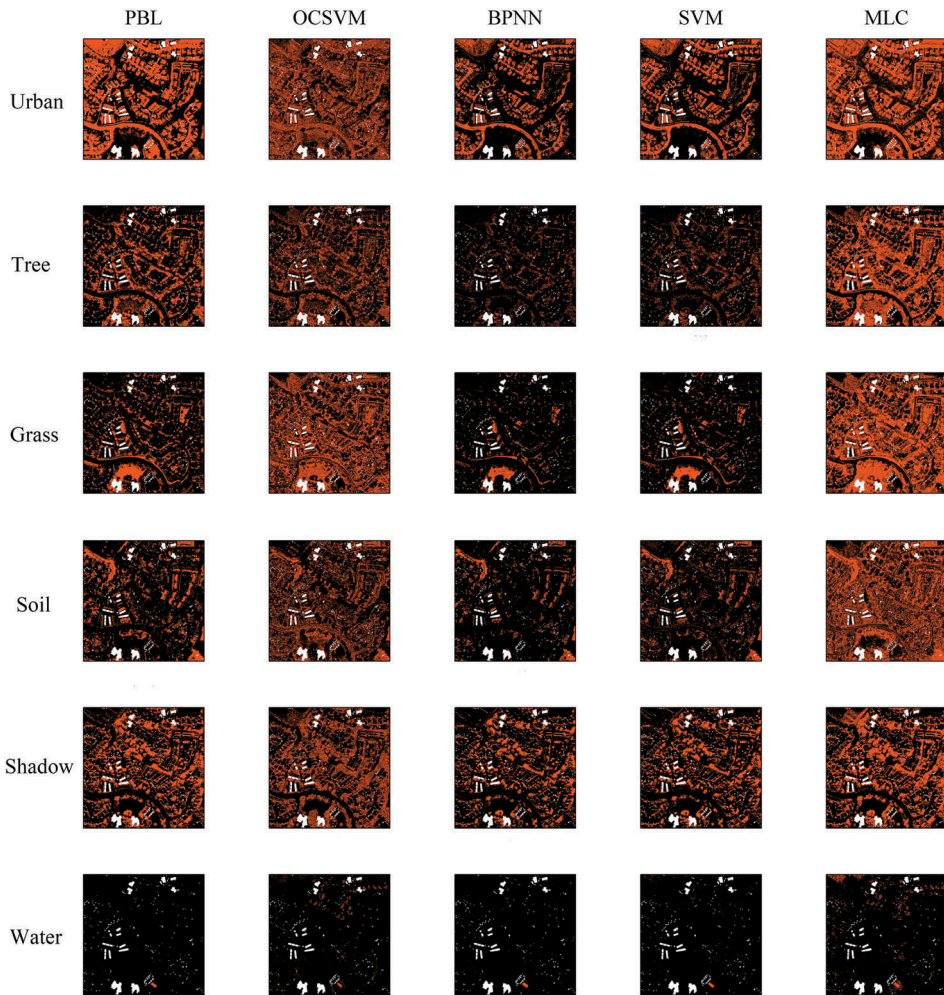
Figure 5. Predicted binary maps of each land type for scene one. Training set size: 1000. Red: positive class. Black: negative class. White: masked pixels.

and MLC are also relatively stable, except for the water class. By contrast, BPNN is more sensitive to the sample size, with larger sample sizes resulting in higher κ values.

Scene two: Figure 7 shows the predicted binary maps in scene two again with a training set size of 1000 pixels. The corresponding quantitative evaluations are provided in Table 2. Again, PBL performs the best followed by BPNN and SVM, whereas MLC and OCSVM perform the worst. For example, PBL obtains a κ value of 0.8 for the urban class, whereas OCSVM only achieves 0.24; the κ values for BPNN, SVM, and MLC are 0.77, 0.76, and 0.67. Figure 8 shows the average κ values over 10 random realizations of the training sets with different sample sizes. Similar to scene one, PBL is shown to be a robust method that results in the highest accuracies for all cases. The accuracy of the other methods is lower and less stable than that of PBL. BPNN is shown to be the most sensitive to training sample size.

Table 1. Mean and standard deviation of overall accuracy (OA) and kappa coefficient (*κ*) for scene one.

	PBL		BPNN		SVM		OCSVM		MLC	
Class	OA(%)	*κ*	OA(%)	*κ*	OA(%)	*κ*	OA(%)	*κ*	OA(%)	*κ*
Urban	87.93	0.76	78.43	0.51	84.14	0.65	66.41	0.34	80.99	0.62
	(0.84)	(0.02)	(5.7)	(0.14)	(0.79)	(0.05)	(1.32)	(0.02)	(2.66)	(0.05)
Tree	85.28	0.58	84.86	0.24	86.51	0.46	78.43	0.39	67.21	0.32
	(1.35)	(0.02)	(0.93)	(0.09)	(0.67)	(0.05)	(2.67)	(0.02)	(1.78)	(0.02)
Grass	88.88	0.53	91.85	0.39	91.88	0.52	74.37	0.23	59.49	0.18
	(1.07)	(0.02)	(0.87)	(0.16)	(0.54)	(0.02)	(5.42)	(0.03)	(4.66)	(0.04)
Soil	90.39	0.5	92.77	0.41	90.09	0.46	80.5	0.24	64.31	0.17
	(0.9)	(0.02)	(0.43)	(0.15)	(1.92)	(0.02)	(3.33)	(0.24)	(3.94)	(0.03)
Shadow	90.97	0.77	91.01	0.74	91.31	0.46	78.39	0.47	82.87	0.61
	(1.09)	(0.02)	(0.49)	(0.02)	(0.64)	(0.01)	(1.81)	(0.02)	(2.27)	(0.04)
Water	99.97	0.88	99.9	0.67	99.73	0.56	98.83	0.16	97.35	0.07
	(0.01)	(0.05)	(0.04)	(0.08)	(0.22)	(0.28)	(0.96)	(0.06)	(0.61)	(0.02)

Training sample size: 1000. Values in parentheses are standard deviations.

Scene Three: Figure 9 shows the predicted binary maps in scene three with a training set size of 1000 pixels. Figure 10 and Table 3 show the corresponding quantitative evaluations. Consistent with the previous scenes, PBL still produces the highest values of OA and *κ*. The accuracies of PBL with a sample size of 200 is even higher than the other methods with a sample size of 1000. For example, the *κ* value for PBL with a sample size of 200 is 0.67, whereas the *κ* values for OCSVM, BPNN, SVM, and MLC with a sample size of 1000 are 0.54, 0.34, 0.49, and 0.55, respectively. Again, BPNN is the most sensitive to training sample size.

4. Discussion

One-class classification can be performed by either one-class or binary classifiers. Binary classifiers have been well studied and widely applied in remote sensing, and they are generally perceived as being more accurate than one-class classifiers (Li, Guo, and Elkan 2011b; Baldeck and Asner 2015; Mack et al. 2016). However, binary classifiers require a larger and exhaustively labelled training set which requires more effort to collect and the labelling of classes that the user is not interested in. Further, the training set will be incomplete if not all land types are identified, which can result in significantly decreased accuracy (Guo et al. 2012). In contrast, one-class classifiers do not require labelled negative data for training and so the labelling cost can be reduced or focused entirely on the target class. They are less likely to suffer from the problem of incomplete training data because only the target class needs to be labelled. The novel contribution of this work is that we compare the binary and one-class classifiers under the same labelling cost, while previous researchers compare the both approaches under unequal labelling costs, which helps us better understand the performances of different classifiers. In the scenario of unequal labelling costs, the number of labelled samples of each land type is the same or adjusted by the class frequency, and the size of labelled samples for binary classifiers is larger than one-class classifiers (details shown in Figure 11). According to previous research, binary classifiers generally outperform one-class classifiers when the training samples are ample and complete (Li, Guo, and Elkan 2011b; Guo et al. 2012; Mack et al. 2016). With the sampling scenario in Figure 11(b), the binary classifier BPNN
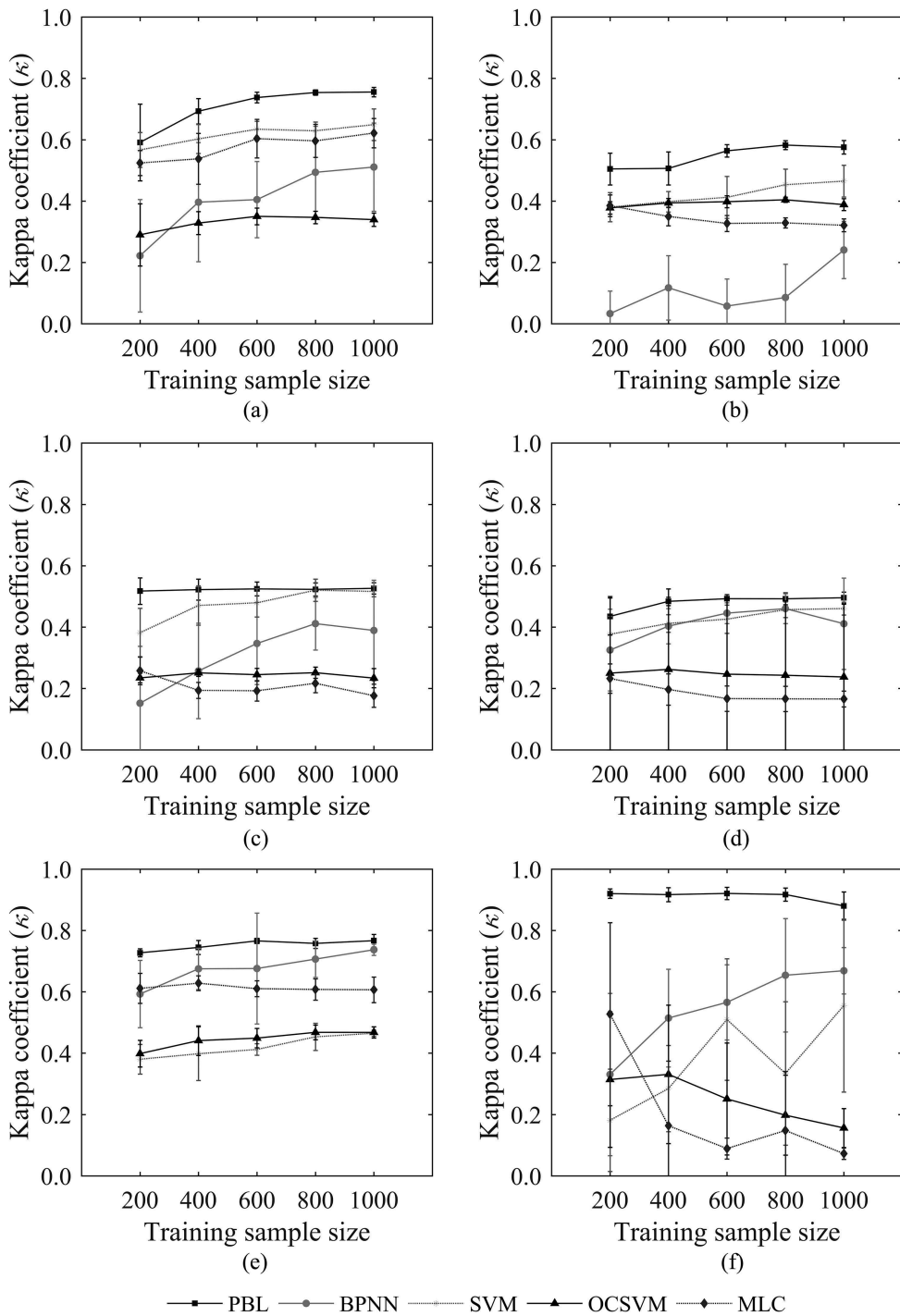
Figure 6. The average and standard deviation of kappa coefficients (κ) obtained by different methods with different training sample sizes for scene one. (a) Urban. (b) Tree. (c) Grass. (d) Soil. (e) Shadow. (f) Water.
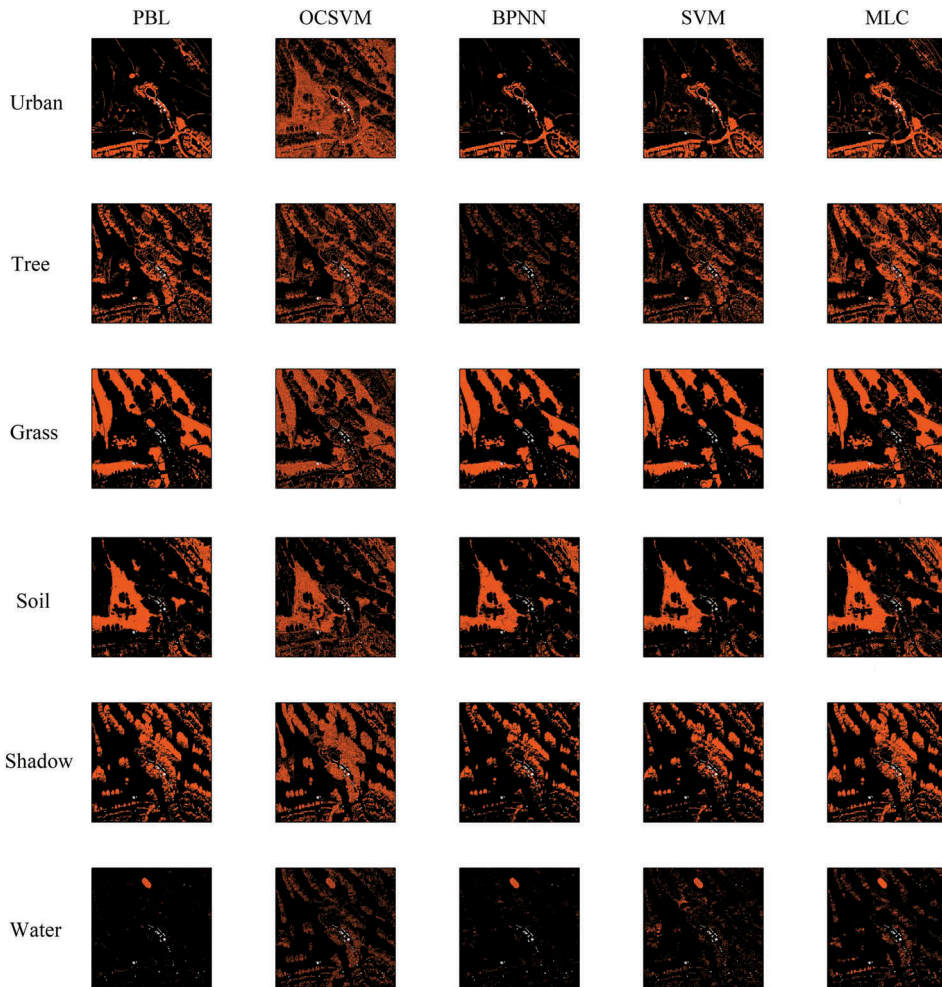
Figure 7. Predicted binary maps of each land type for scene two. Training set size: 1000. Red: positive class. Black: negative class. White: masked pixels.

consistently outperforms PBL in predicting the spatial distribution of a virtual species (Li, Guo, and Elkan 2011b). With the sampling scenario in Figure 11(a), classifying all the land types in one time using the binary classifier BPNN outperforms classifying the land types one-by-one using the one-class classifier PUL, given that all of the land types have been exhaustively labelled in the training set; however, PUL outperforms BPNN when the training set is incomplete, i.e. one of the land types is not sampled (Guo et al. 2012). With the same labelling cost scenario, our experimental results show that the one-class classifier, PBL, is the most robust method, consistently outperforming all other methods. With these findings, we therefore confirm that both additional labelled and unlabelled data are responsible for the improvement of classification accuracy. Although classification accuracy increases with training sample size, PBL with a small sample size (e.g. 200) is shown to outperform other methods with a larger sample size (e.g. 1000). This is because PBL can take advantage of 200 labelled positive and additional 5000 unlabelled

Table 2. Mean and standard deviation of overall accuracy (OA) and kappa coefficient (κ) for scene two.

Class	PBL OA(%)	κ	BPNN OA(%)	κ	SVM OA(%)	κ	OCSVM OA(%)	κ	MLC OA(%)	κ
Urban	95.36	0.8	95	0.77	94.68	0.76	72.54	0.24	92.12	0.67
	(0.37)	(0.01)	(0.18)	(0.01)	(0.41)	(0.02)	(3.25)	(0.02)	(0.31)	(0.01)
Tree	88.59	0.63	87.69	0.44	87.57	0.53	83.38	0.47	82.49	0.51
	(0.53)	(0.01)	(0.7)	(0.09)	(0.97)	(0.03)	(2.02)	(0.03)	(1.61)	(0.02)
Grass	94.45	0.87	93.16	0.83	94.04	0.86	82.73	0.58	90.82	0.79
	(0.17)	(0)	(1.21)	(0.03)	(0.45)	(0.01)	(0.52)	(0.02)	(1.34)	(0.03)
Soil	94.3	0.82	93.67	0.79	93.73	0.8	84.59	0.55	91.06	0.74
	(0.2)	(0)	(0.48)	(0.02)	(0.68)	(0.02)	(2.54)	(0.03)	(1.3)	(0.03)
Shadow	93.36	0.8	92.67	0.75	93.15	0.78	82.55	0.54	90.14	0.73
	(0.39)	(0.01)	(0.9)	(0.04)	(0.44)	(0.02)	(1.34)	(0.02)	(0.98)	(0.02)
Water	99.55	0.65	99.13	0.49	96.9	0.2	91.61	0.07	93.76	0.13
	(0.14)	(0.06)	(0.28)	(0.08)	(0.52)	(0.03)	(3.1)	(0.01)	(2.33)	(0.04)

Training sample size: 1000. Values in parentheses are standard deviations.

data for training, with the cost of labelling 200 samples; but the binary classifiers only take advantage of 167 labelled positive and 833 labelled negative data for training, with the cost of labelling 1000 samples. Thus, PBL is shown to be a good option for one-class classification in remote sensing with respect to both accuracy and cost.

The solid performance of PBL results from its ability to exploit unlabelled data during training. Although the class memberships of unlabelled data are not known, this data provides information on the distribution of the covariates that is still useful for model training (Castelli and Cover 1996; Li, Guo, and Elkan 2011b). Large sets of unlabelled data can be collected without manual labelling effort making it more likely that the training set contains samples of all land types. Positive-unlabelled one-class classifiers, such as PBL, are thus able to achieve good classification performance even when trained with a limited number of labelled positive samples (Elkan and Noto 2008; Li, Guo, and Elkan 2011a). Positive-only one-class classifiers, such as OCSVM, do not exploit unlabelled data. OCSVM is also sensitive to free parameters that are difficult to tune (Manevitz and Yousef 2002), and appropriate model selection could significantly improve the accuracy (Mack and Waske 2017). We used a common, empirical approach (cross-validation and grid search) to tune the two free parameters (γ and v) of OCSVM but it is possible we were not able to find the optimal values. Please note that the model selection was based on a validation set where each class contains the same number of samples, but the accuracy assessment for scenes one and two was based on the whole labelled image where the population prevalence of each class is different (e.g. the area of urban is larger than water). In order to test whether the ratio of samples will affect the model selection, we tried to adjust the number of samples in the validation set according the population prevalence of each class, but we found that it does not improve the results. Whatever the reason, OCSVM does not perform well in our experiments. This is consistent with previous results which show positive-unlabelled methods generally outperform positive-only methods (Liu et al. 2003; Elkan and Noto 2008; Li and Guo 2010; Li, Guo, and Elkan 2011a).

Note that PBL is not a classifier but a framework that uses a naive model Pr($s = 1 \mid \boldsymbol{x}$, $\eta = 1$) to calibrate the desired model Pr($y = 1 \mid \boldsymbol{x}$). Even though the naive model for our implementation of PBL has the same network structure as the binary BPNN classifier that we consider, PBL is shown to outperform the binary classifier. This can be explained by the
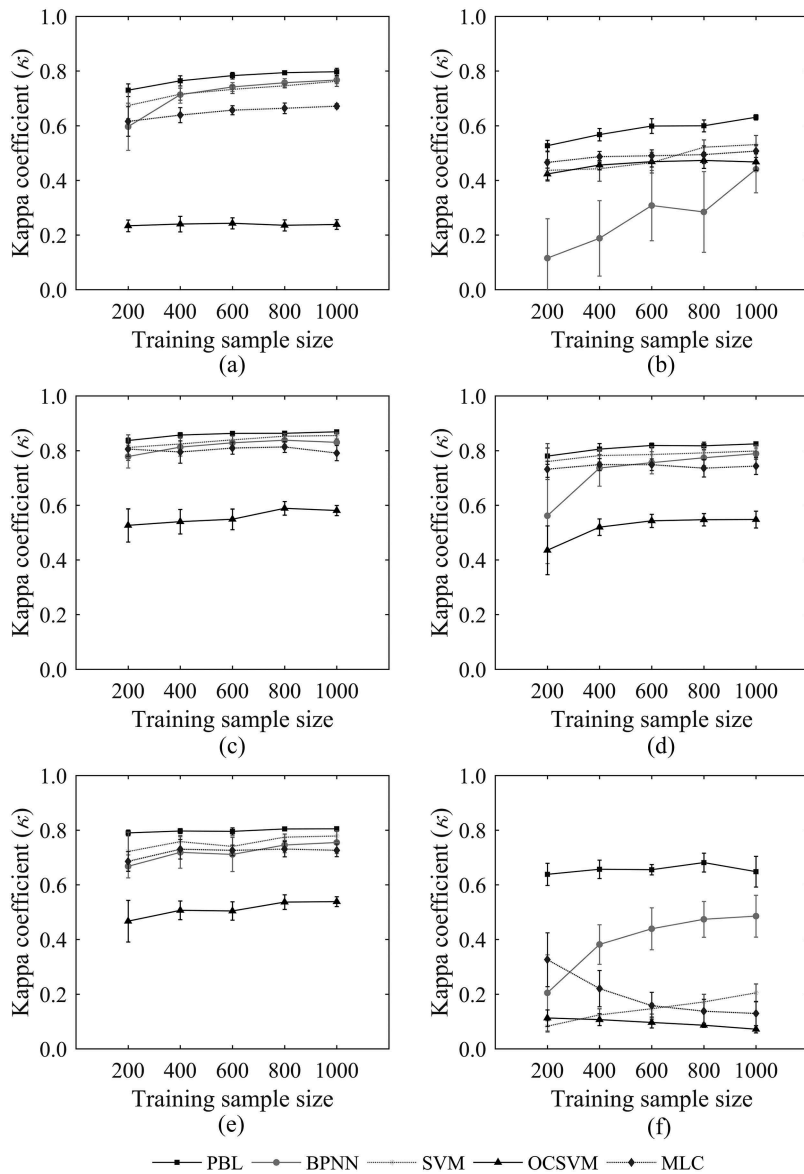
Figure 8. The average and standard deviation of kappa coefficients (κ) obtained by different methods with different training sample sizes for scene two. (a) Urban. (b) Tree. (c) Grass. (d) Soil. (e) Shadow. (f) Water.

fact that BPNN is very sensitive to training set size. Neural networks can model complex, non-linear relationships but they require large amounts of training data (Bishop 2006). When BPNN is used as a binary classifier, it sees only 33 and 167 positive samples when the training set size is 200 and 1000. PBL, and thus the associated naive BPNN model, not only sees significantly more positive labelled samples in these two cases, 200 and 1000 respectively, but also sees 5000 unlabelled samples. This explains why PBL outperforms the binary classifiers even when it is trained with a smaller labelled set.
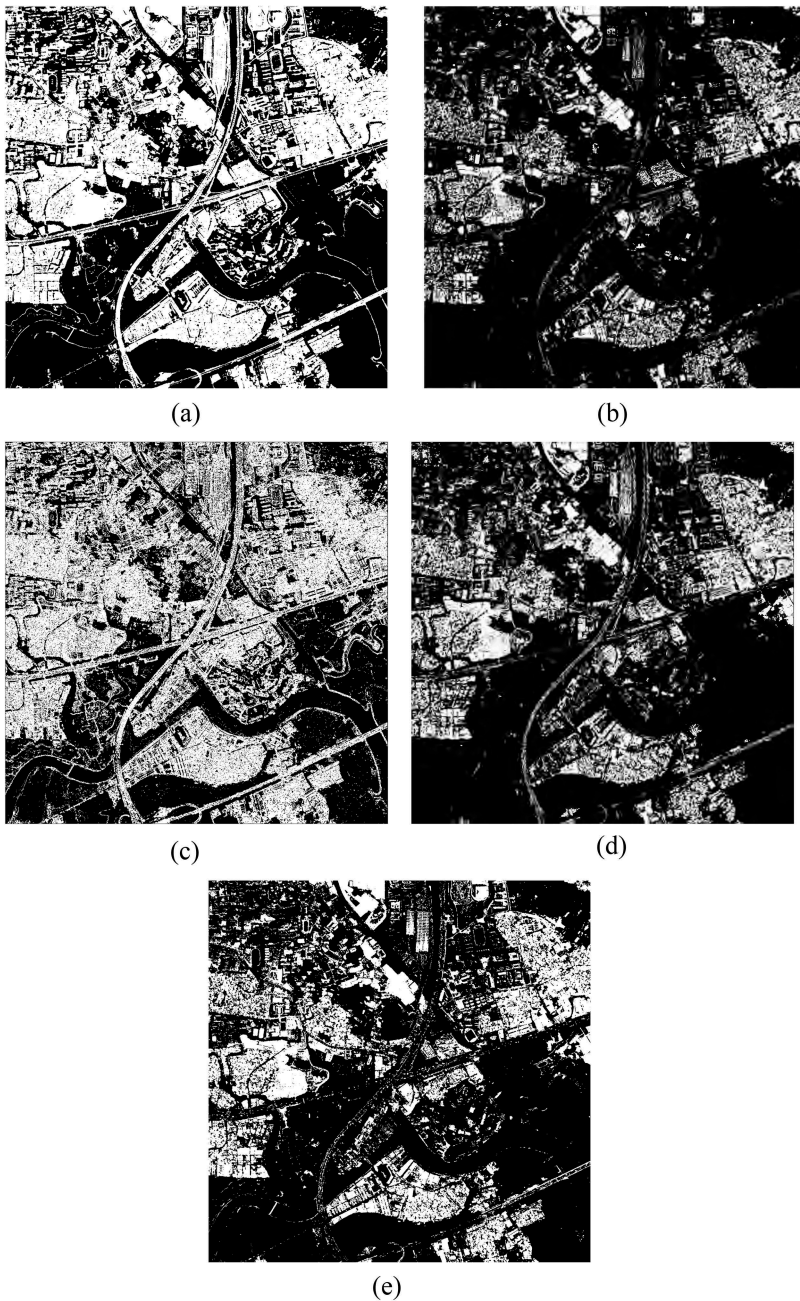
Figure 9. Predicted binary maps of urban class for scene three. Training set size: 1000. White: positive class. Black: negative class. (a) PBL. (b) BPNN. (c) OCSVM. (d) SVM. (e) MLC.

SVM is another state-of-the-art binary classifier, especially for classifying high-dimensional data given small training sample sizes (Huang, Davis, and Townshend 2002; Pal and Mather 2005). Our experiments show that larger sample sizes result in slightly higher accuracy for SVM but the effect is less significant compared to BPNN. SVM is
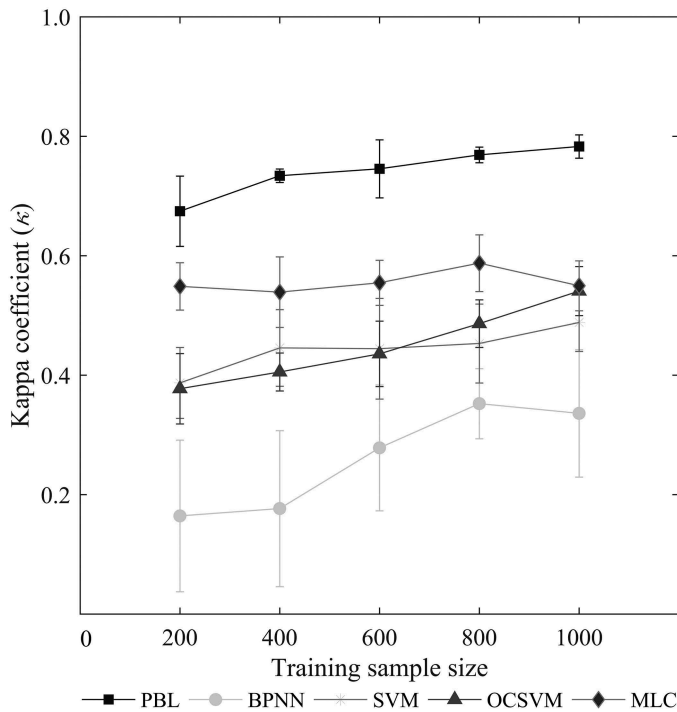
Figure 10. The average and standard deviation of kappa coefficients (κ) of urban class obtained by different methods with different training sample sizes for scene three.

Table 3. Mean and standard deviation of overall accuracy (OA) and kappa coefficient (κ) of urban class for scene three.

Sample size	PBL		BPNN		SVM		OCSVM		MLC	
	OA(%)	κ	OA(%)	κ	OA(%)	κ	OA(%)	κ	OA(%)	κ
200	83.72	0.67	58.22	0.16	69.35	0.39	68.87	0.38	77.44	0.55
	(2.94)	(0.06)	(6.35)	(0.13)	(2.97)	(0.06)	(2.95)	(0.06)	(1.98)	(0.04)
400	86.7	0.73	58.83	0.18	72.28	0.45	70.27	0.41	76.95	0.54
	(0.57)	(0.01)	(6.53)	(0.13)	(3.21)	(0.06)	(1.59)	(0.03)	(2.96)	(0.06)
600	87.28	0.75	63.91	0.28	72.22	0.44	71.78	0.44	77.73	0.55
	(2.43)	(0.05)	(5.27)	(0.11)	(4.21)	(0.08)	(2.74)	(0.05)	(1.89)	(0.04)
800	88.44	0.77	67.62	0.35	72.65	0.45	74.31	0.49	79.38	0.59
	(0.66)	(0.01)	(2.93)	(0.06)	(3.31)	(0.07)	(2)	(0.04)	(2.38)	(0.05)
1000	89.15	0.78	66.8	0.34	74.41	0.49	77.04	0.54	77.48	0.55
	(0.97)	(0.02)	(5.33)	(0.11)	(2.42)	(0.05)	(2.05)	(0.04)	(2.09)	(0.04)

Values in parentheses are standard deviations.

shown to outperform BPNN for small sample sizes but the gap narrows as the sample size increases. When compared to PBL, SVM is shown to be less accurate and less robust, again probably due to the smaller amount of training data (positive and overall) that it sees. SVM is also sensitive to free parameters that are difficult and time consuming to tune. Similar to the model selection of OCSVM, we tried to adjust the number of samples of each class in the validation set, but the results are not improved. The other binary classifier we consider, MLC, is also widely used in remote sensing (Jeon and Landgrebe 1999; Huang, Davis, and Townshend 2002; Pal and Mather 2005). MLC is a parametric
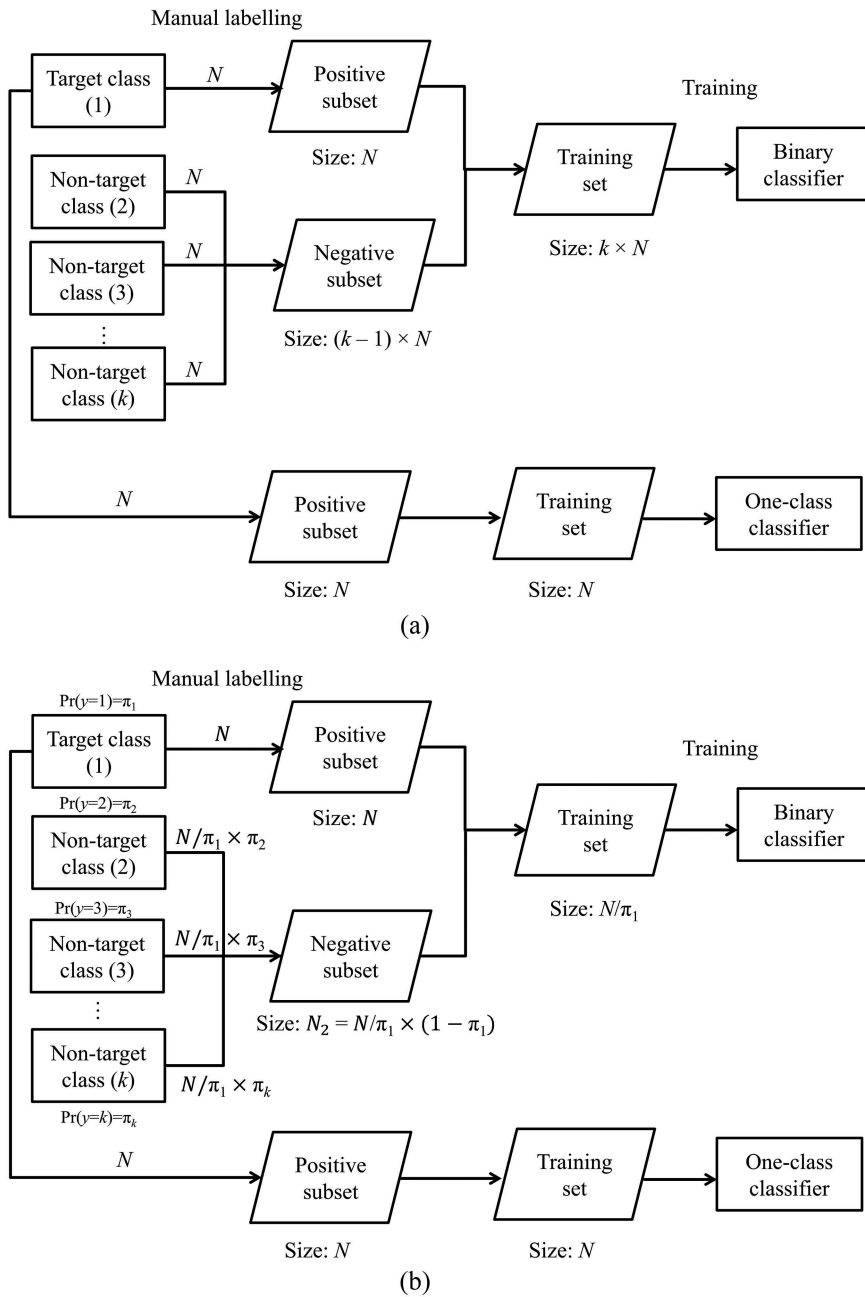
Figure 11. Flow chart of the sampling scheme for one-class and binary classifiers with unequal labelling costs. For binary classifiers, the number of labelled samples of each land type is the same (a) or adjusted by the class frequency (b). k refers to the number of land types.

classifier based on the Bayes' theorem, with the assumption that the classes have multivariate normal distributions (Huang, Davis, and Townshend 2002; Ahmad and Quegan 2012). Our experiments show that MLC generally performs poorly with a few

exceptions. This again is likely due to the training sample size although it could also be that the multivariate normal assumption is not valid for our problem.

One-class classification is a common problem in many fields, such as image classification, text classification, species distribution modelling, wildland search and rescue incident modelling, and landslide susceptibility modelling (Liu et al. 2003; Elkan and Noto 2008; Li, Guo, and Elkan 2011a; Li, Guo, and Elkan 2011b; Vorpahl et al. 2012; Doherty et al. 2014). As opposed to learning from positive and negative data, learning a classifier from positive and unlabelled data is a challenging but important problem (Elkan and Noto 2008; Ward et al. 2009). To address this problem, different methods have been developed in the literature. We consider PBL as one example to demonstrate that positive-unlabelled methods can be a better choice than the traditional binary classifiers for one-class remote sensing classification. There are other promising positive-unlabelled methods which could also be considered. Another limitation of this study is that the methods are evaluated using just three scenes of remotely sensed data sets. Future work might consider additional positive-unlabelled methods using a broader range of remotely sensed data.

5. Conclusion

We compare one-class and binary classifiers for detecting a single land type in high-spatial-resolution imagery given the same amount of manual labelling effort. PBL, a positive-unlabelled one-class classifier, is shown to provide the best results compared to three binary classifiers, BPNN, SVM, and MLC, as well as a positive-only one-class classifier, OCSVM. PBL does not require labelled negative training samples and thus the entire labelling budget can go towards positive samples. This allows it to outperform the binary classifiers. And, while OCSVM also does not require labelled negative training samples, it does not exploit unlabelled data like PBL. It is also sensitive to free parameters. In summary, practitioners could consider positive-unlabelled classifiers such as PBL for performing one-class classification in remote sensing especially when there is limited labelled training data.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Xiaoping Liu 🔴 http://orcid.org/0000-0003-4242-5392

References

Ahmad, A., and S. Quegan. 2012. "Analysis of Maximum Likelihood Classification on Multispectral Data." *Applied Mathematical Sciences* 6 (129): 6425–6436.

Baldeck, C. A., and G. P. Asner. 2015. "Single-Species Detection with Airborne Imaging Spectroscopy Data: A Comparison of Support Vector Techniques." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8 (6): 2501–2512. doi:10.1109/JSTARS.2014.2346475.

Bischof, H., W. Schneider, and A. J. Pinz. 1992. "Multispectral Classification of Landsat-Images Using Neural Networks." *IEEE Transactions on Geoscience and Remote Sensing* 30 (3): 482–490. doi:10.1109/36.142926.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. New York, NY: Springer-Verlag New York, Inc.

Castelli, V., and T. M. Cover. 1996. "The Relative Value of Labeled and Unlabeled Samples in Pattern Recognition with an Unknown Mixing Parameter." *IEEE Transactions on Information Theory* 42 (6): 2102–2117. doi:10.1109/18.556600.

Chang, C., and C. Lin. 2015. "A Library for Support Vector Machines." Accessed 21 September 2015. Available online at: https://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Chen, X., D. Yin, J. Chen, and X. Cao. 2016. "Effect of Training Strategy for Positive and Unlabelled Learning Classification: Test on Landsat Imagery." *Remote Sensing Letters* 7 (11): 1063–1072. doi:10.1080/2150704X.2016.1217437.

Doherty, P. J., Q. Guo, W. Li, and J. Doke. 2014. "Space-Time Analyses for Forecasting Future Incident Occurrence: A Case Study from Yosemite National Park Using the Presence and Background Learning Algorithm." *International Journal of Geographical Information Science* 28 (5): 910–927. doi:10.1080/13658816.2014.890202.

Elkan, C., and N.Noto 2008. "Learning Classifiers from Only Positive and Unlabeled Data." In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 213–220. Las Vegas, Nevada, USA: ACM.

Foody, G. M. 2004. "Supervised Image Classification by MLP and RBF Neural Networks with and without an Exhaustively Defined Set of Classes." *International Journal of Remote Sensing* 25 (15): 3091–3104. doi:10.1080/01431160310001648019.

Foody, G. M., and A. Mathur. 2004. "A Relative Evaluation of Multiclass Image Classification by Support Vector Machines." *IEEE Transactions on Geoscience and Remote Sensing* 42 (6): 1335–1343. doi:10.1109/TGRS.2004.827257.

Foody, G. M., A. Mathur, C. Sanchez-Hernandez, and D. S. Boyd. 2006. "Training Set Size Requirements for the Classification of a Specific Class." *Remote Sensing of Environment* 104 (1): 1–14. doi:10.1016/j.rse.2006.03.004.

Guo, Q., W. Li, D. Liu, and J. Chen. 2012. "A Framework for Supervised Image Classification with Incomplete Training Samples." *Photogrammetric Engineering & Remote Sensing* 78 (6): 595–604. doi:10.14358/PERS.78.6.595.

Huang, C., L. S. Davis, and J. R. G. Townshend. 2002. "An Assessment of Support Vector Machines for Land Cover Classification." *International Journal of Remote Sensing* 23 (4): 725–749. doi:10.1080/01431160110040323.

Hung, M. S., M. Y. Hu, M. S. Shanker, and B. E. Patuwo. 1996. "Estimating Posterior Probabilities in Classification Problems with Neural Networks." *International Journal of Computational Intelligence and Organizations* 1 (1): 49–60.

Jensen, J. R. 2000. *Remote Sensing of the Environment: An Earth Resource Perspective*. Upper Saddle River, NJ: Prentice Hall.

Jeon, B., and D. A. Landgrebe. 1999. "Partially Supervised Classification Using Weighted Unsupervised Clustering." *IEEE Transactions on Geoscience and Remote Sensing* 37 (2): 1073–1079. doi:10.1109/36.752225.

Kavzoglu, T., and P. M. Mather. 2003. "The Use of Backpropagating Artificial Neural Networks in Land Cover Classification." *International Journal of Remote Sensing* 24 (23): 4907–4938. doi:10.1080/0143116031000114851.

Li, W., and Q. Guo. 2010. "A Maximum Entropy Approach to One-Class Classification of Remote Sensing Imagery." *International Journal of Remote Sensing* 31 (8): 2227–2235. doi:10.1080/01431161003702245.

Li, W., and Q. Guo. 2014. "A New Accuracy Assessment Method for One-Class Remote Sensing Classification." *IEEE Transactions on Geoscience and Remote Sensing* 52 (8): 4621–4632. doi:10.1109/TGRS.2013.2283082.

Li, W., Q. Guo, and C. Elkan. 2011a. "A Positive and Unlabeled Learning Algorithm for One-Class Classification of Remote-Sensing Data." *IEEE Transactions on Geoscience and Remote Sensing* 49 (2): 717–725. doi:10.1109/TGRS.2010.2058578.

Li, W., Q. Guo, and C. Elkan. 2011b. "Can We Model the Probability of Presence of Species without Absence Data?" *Ecography* 34 (6): 1096–1105. doi:10.1111/j.1600-0587.2011.06888.x.

Lin, J., X. Liu, K. Li, and X. Li. 2014. "A Maximum Entropy Method to Extract Urban Land by Combining MODIS Reflectance, MODIS NDVI, and DMSP-OLS Data." *International Journal of Remote Sensing* 35 (18): 6708–6727. doi:10.1080/01431161.2014.960623.

Liu, B., Y. Dai, X. Li, W. S. Lee, and P. S. Yu. 2003. "Building Text Classifiers Using Positive and Unlabeled Examples." In *Proceedings of the Third IEEE International Conference on Data Mining*, 179–186. IEEE Computer Society.

Liu, X., L. Ma, X. Li, B. Ai, S. Li, and Z. He. 2014. "Simulating Urban Growth by Integrating Landscape Expansion Index (LEI) and Cellular Automata." *International Journal of Geographical Information Science* 28 (1): 148–163. doi:10.1080/13658816.2013.831097.

Mack, B., R. Roscher, S. Stenzel, H. Feilhauer, S. Schmidtlein, and B. Waske. 2016. "Mapping Raised Bogs with an Iterative One-Class Classification Approach." *ISPRS Journal of Photogrammetry and Remote Sensing* 120: 53–64. doi:10.1016/j.isprsjprs.2016.07.008.

Mack, B., R. Roscher, and B. Waske. 2014. "Can I Trust My One-Class Classification?" *Remote Sensing* 6 (9): 8779–8802. doi:10.3390/rs6098779.

Mack, B., and B. Waske. 2017. "In-Depth Comparisons of MaxEnt, Biased SVM and One-Class SVM for One-Class Classification of Remote Sensing Data." *Remote Sensing Letters* 8 (3): 290–299. doi:10.1080/2150704X.2016.1265689.

Manevitz, L. M., and M. Yousef. 2002. "One-Class SVMs for Document Classification." *Journal of Machine Learning Research* 2: 139–154.

Munoz-Mari, J., L. Bruzzone, and G. Camps-Valls. 2007. "A Support Vector Domain Description Approach to Supervised Classification of Remote Sensing Images." *IEEE Transactions on Geoscience and Remote Sensing* 45 (8): 2683–2692. doi:10.1109/TGRS.2007.897425.

Pal, M., and P. M. Mather. 2005. "Support Vector Machines for Classification in Remote Sensing." *International Journal of Remote Sensing* 26 (5): 1007–1011. doi:10.1080/01431160512331314083.

Palocsay, S. W., S. P. Stevens, and R. G. Brookshire. 2001. "An Empirical Evaluation of Probability Estimation with Neural Networks." *Neural Computing & Applications* 10 (1): 48–55. doi:10.1007/s005210170017.

Phillips, S. J., M. Dud´ık, and R. E. Schapire. 2004. A Maximum Entropy Approach to Species Distribution Modeling. In *Proceedings of the Twenty-First International Conference on Machine Learning*, 655–662. Banff, Alberta, Canada.

Richard, M. D., and R. P. Lippmann. 1991. "Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities." *Neural Computation* 3 (4): 461–483. doi:10.1162/neco.1991.3.4.461.

Richards, J. A., and X. Jia. 1999. *Remote Sensing Digital Image Analysis: An Introduction*. Edited by D. E. Ricken and W. Gessner. Berlin: Springer-Verlag New York, Inc.

Schölkopf, B., J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. 2001. "Estimating the Support of a High-Dimensional Distribution." *Neural Computation* 13 (7): 1443–1471. doi:10.1162/089976601750264965.

Tax, D. M. J. 2001. "One-Class Classification, Concept-Learning in the Absence of Counter-Examples." Dissertation, Delft Univ. Technol., Delft.

Vorpahl, P., H. Elsenbeer, M. Märker, and B. Schröder. 2012. "How Can Statistical Models Help to Determine Driving Factors of Landslides?" *Ecological Modelling* 239: 27–39. doi:10.1016/j.ecolmodel.2011.12.007.

Wan, B., Q. Guo, F. Fang, Y. Su, and R. Wang. 2015. "Mapping US Urban Extents from MODIS Data Using One-Class Classification Method." *Remote Sensing* 7 (8): 10143–10163. doi:10.3390/rs70810143.

Ward, G., T. Hastie, S. Barry, J. Elith, and J. R. Leathwick. 2009. "Presence-Only Data and the EM Algorithm." *Biometrics* 65: 554–563. doi:10.1111/j.1541-0420.2008.01116.x.

Xu, X., X. Liu, X. Li, Q. Xin, Y. Chen, Q. Shi, and B. Ai. 2016. "Global Snow Cover Estimation with Microwave Brightness Temperature Measurements and One-Class in Situ Observations." *Remote Sensing of Environment* 182: 227–251. doi:10.1016/j.rse.2016.05.010.

Zhu, C., B. Liu, Q. Yu, X. Liu, and W. Yu. 2012. A Spy Positive and Unlabeled Learning Classifier and Its Application in HR SAR Image Scene Interpretation. In *2012 IEEE Radar Conference*, 7–11 May 2012.