

EVALUATION OF ONE-CLASS SVM FOR PIXEL-BASED AND SEGMENT-BASED CLASSIFICATION IN REMOTE SENSING

Andreas Christian Braun

Institute for Photogrammetry and Remote Sensing
 Karlsruhe Institute of Technology (KIT)
 Engesserstr. 7, 76128 Karlsruhe, Germany
 andreas.ch.braun@kit.edu

Commission III/3

ABSTRACT:

Despite the wide usage of standard cost support vector machines in remote sensing, a less common approach of SVM - named *one-class SVM* - has found only little attention. This seems surprising to us, as one-class SVM overcomes one important shortcoming of standard cost SVM: the problems occurring from unassigned classes present in the scene and the difficulties handling the rejection class. We present an introduction into one-class SVM for classification in remote sensing and a comparison with standard cost SVM.

1 INTRODUCTION

Within the last decade, classification of remotely sensed data using support vector machine (SVM) approaches has gained increasing attention in the remote sensing community. The generally high performance of SVMs (Vapnik and Guyon, 1995) in terms of overall classification and the high generalization performance - due to their strong theoretical basis in statistical learning theory (Vapnik, 1999) - are the key benefits of this method. Apart from the usage of the kernel method, SVMs further increase performance by employing strategies for specialized error variables (called *slack-variables*). These establish trade-offs between the training accuracy and the necessity to generalize appropriately. Various approaches for this trade-off have been proposed. The most commonly used *cost* SVM approach with the error penalization factor C , however, has one major shortcoming regarding the classification of remotely sensed data. In order to find good separation criteria, one needs to incorporate each and every land-cover class of the entire scene into the SVM training by assigning training areas. Thus, even though one might only be interested in a single landcover class, time consuming SVM training has to be performed for all classes. By doing so, the risk of confusion between classes is being raised. This drawback can be overcome by a new type of slack-variable handling - called ν -SVM - which was proposed by (Schölkopf et al., 2001). Instead of finding optimal hyperplanes which separate two classes from each other, ν -SVM can be adapted to find optimal hyperspheres that envelop the training patterns from just one class. Such a framework is called one-class SVM (Chang and Lin, 2010) and can be considered an absolute classifier. In our paper, we present a comparative outline on our first results produced with one-class SVM for the classification of both pixel-based and segmented data.

2 MATHEMATICAL FOUNDATIONS

In this section, we will give a short introduction into the principle ideas behind SVM. We will focus on the foundations of one-class SVM and its analogy to the well-established cost SVM (Burges, 1998). One-class SVM was proposed by (Schölkopf et al., 2001) to estimate the support of unknown distributions in high-dimensional feature spaces. The general idea behind one-class SVM - and the most important difference to cost SVM, at the same time - is, that it is not designed to separate two classes from each other, but to precisely describe the distribution of one

single class. Given a training set of x_i instances without any class information,

$$x_i \in \mathbb{S}^\tau, \quad i = 1, \dots, l \quad (1)$$

one-class SVM constructs an enclosing hypersphere of minimum volume within a space \mathbb{H}^ψ of higher dimensionality for the given set (i.e. $\psi \geq \tau^1$). This is accomplished by introducing a non-linear feature mapping $\Phi : \mathbb{S}^\tau \rightarrow \mathbb{H}^\psi$. Note that just like cost SVM, one-class SVM solves this task, substituting Φ with kernel functions. To find this optimal enclosing hypersphere, the origin is used as the only reference point outside the class (cf. Fig. 1). In Eq.3, it can be shown that the distance from the origin to the hypersphere can be maximized by minimizing the norm of vector w and the parameter $\rho \in [0, 1]$. Once w and ρ are found, the decision function for new and unlabeled points is given by

$$f(x) = \text{sgn}(w \cdot \Phi(x_i - \rho)) \quad (2)$$

Thus, one-class SVM delivers a decision function which takes the value +1 in a region which includes the majority of training data and takes the value -1 for only very few data (considered non-supporting training instances). Introducing a parameter ν delivers a trade-off between the fraction of non-supporting patterns and the size of the positive region. The parameter $\nu \in (0, 1]$ serves at the same time as an upper bound on the fraction of non-supporting patterns and as a lower bound on the number of support vectors (Chen et al., 2005).

$$\min_{w, \xi, \rho} \frac{1}{2} \|w\|^2 - \rho + \frac{l}{\nu l} \sum_{i=1}^l \xi_i \quad (3)$$

$$\begin{aligned} \text{subject to } & w^\top \Phi(x_i) \geq \rho - \xi_i \\ & \xi_i \geq 0, \quad i = 1, \dots, l \end{aligned}$$

A solution is found by solving a quadratic optimization problem (see Eq.3). Usually, this problem is solved in its dualized form (given in Eq.4), derived by the Lagrangian.

$$\min_{\alpha} \frac{1}{2} \alpha^\top Q \alpha \quad (4)$$

$$\text{subject to } 0 \leq \alpha_i \leq \frac{1}{\nu l}, \quad i = 1, \dots, l$$

¹ $\psi = \tau$ for the linear kernel $K(x_i, x_j) = \langle x_i, x_j \rangle \quad i, j \in 1, \dots, l$

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha = 1$$

Where $Q_{i,j} = K(x_i, x_j) \equiv \Phi(x_i)^T \Phi(x_j)$ $i, j \in 1, \dots, l$ represents a kernel function which performs the transformation $\Phi(x_i)$ into the higher dimensional space implicitly (Burges, 1998). These kernels have proprietary parameters (like γ for the width of the gaussian bells in the radial basis function kernel, cf. Eq.5) which need to be adjusted together with ν by performing a grid search. Computation time for SVM grid search is time-consuming and prolongates drastically with the amount of training information. This explains the numerous intents to reduce necessary training data to a minimum. As we can see, one-class SVM is

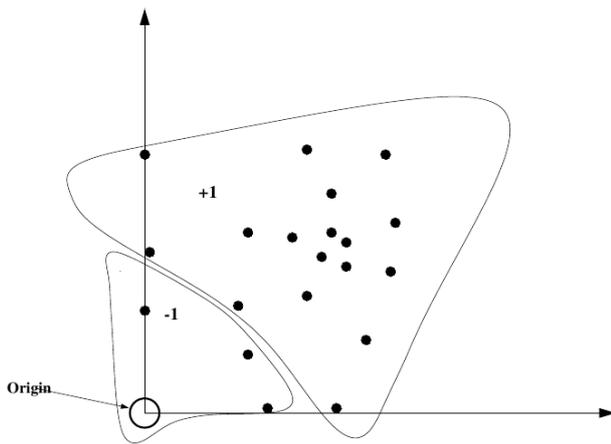


Figure 1: Principle of one-class SVM from (Manevitz and Yousef, 2001) (*the origin is labeled as the only point in class "-1" while the training samples are labeled "+1", a small proportion of the class is considered non-supporting patterns*)

an absolute classifier that makes use of the same mathematical foundations like cost SVM, but is adapted to detect only single classes employing the ν -strategy for error penalization.

3 RESULTS AND DISCUSSION

In this section, we will present classifications of HyMap, QuickBird, Landsat and LiDAR data, we obtained using the one-class SVM classifier. As the name suggests, we first used one-class SVM to detect a single class only in subsection 3.1. Cost SVM is a relative classifier, it finds separating planes between different classes. Each class present in the scene needs to be represented in the training data set in order to achieve meaningful results. This might be inconvenient if one is only interested in a single class. One-class SVM can bridge this gap, making it a valuable alternative to cost SVM for remote sensing. After that, we will give results with more than one class for Landsat and HyMap in subsection 3.2. For HyMap, we classified with a pixel-based approach while we preprocessed the Landsat scene with a mean-shift segmentation (Comaniciu, 2002) to perform a segment-based classification. We compare our results to classifications using standard cost SVM.

$$K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|}{2\gamma^2}\right) \quad (5)$$

We used the LibSVM 2.91 implementation by (Chang and Lin, 2010) for Matlab R2009a to perform our classifications. In order to find a well suited SVM model, for each class in every classification, we realized a grid search. We used a radial basis function kernel (cf. Eq.5) with $\gamma \in [2^{-5}, 2^{15}]$ and $\nu \in (0, 1]$.

As ν is a lower bound for the number of free support vectors (FSV), (Chen et al., 2005) consider the number of FSV needed as a validation method for the performance of a certain parameter combination. Unfortunately, we obtained heavily overfitted SVM training models by doing so. According to (Tran et al., 2005) the $\xi\alpha\rho$ -estimate is a good choice for the task, but as LibSVM 2.91 does not deliver all the parameters needed, we refused to implement it and decided to use the leave-one-out error as described in (Fukanaga and Hummels, 1989) instead. Leave-one-out worked well but was computationally very expensive. The LiDAR data set with a single class (3339 training pixels) took more than 32 hours performing grid search.

3.1 DETECTING URBAN TREES

To show the capabilities of one-class SVM to detect classes of interest without any information on the other classes present in the dataset, we first used it to classify one single class only. As we work on urban vegetation, we took *urban trees* as our first class of interest. We used three datasets showing the university campus of KIT². In each scene, we assigned a different training area for *urban trees*. The first was a HyMap scene (see Fig.2), taken 2003, the second one a QuickBird scene of 2005 (see Fig.4) and the last one a LiDAR scene (see Fig.6) from 2002. The only information we used was the first-pulse/last-pulse information. Apart from their height, the criterion that distinguishes trees from other pixels is their significant difference between first- and last-pulse. As the same applies to roof-edges, confusion between building edges and trees had to be expected. The classification results are shown in Fig.5 to Fig.7. The result for QuickBird (cf. Fig.5) seems most convincing. Visually, we did not find regions of pixels to be falsely classified as trees or trees left out (false-negatives). In the HyMap result (cf. Fig.3) one can see that at several places, pixels belonging to other classes have been classified as forest. For instance lawn in the north of the castle, gravel in front of the castle and some borders of the park lawn. On the other hand, there is quite some loss in the dense forest in the north-east. The most probable explanation that came to our minds is that some training pixels in the forest might have been a mixture between trees and other surface cover types (like soil or grass). With a ground resolution of 4×4 m, especially in the forest, this is quite possible. Finally, visual results for the LiDAR scene (cf. Fig.7) are in general satisfying as well. Unfortunately, the edges of many roofs have been classified as tree pixels as well (false-positives) due to the shortcoming of the data mentioned above. As one can see, tree areas without a significant difference between first- and last-pulse (like the alley trees in the south of the castle) have been left out as well. These misclassifications might be the reciprocal effect of the same shortcoming. After classification, we defined control areas for *urban trees* on the screen, using our knowledge of the scene. We computed quality measures for the class *urban trees* (results are given in table 1). As one can see, true-positive and true-negative rates are considerably high for all of the scenes. The only exception is the true-positive rate for LiDAR which is somewhat lower. We believe that this is due to the confusion with building edges, mentioned above. On the other hand, it becomes clear, that for all three scenes, false-negative rates are quite high. This indicates, that one-class SVM does not classify pixels falsely to a high extend, but misses quite many pixels which are later considered as rejection class. As we have seen, one-class SVM could successfully be employed to detect one single class in various datasets without introducing training information for the other classes. A task that could not have been fulfilled with cost SVM as it is usually applied.

²Karlsruhe Institute of Technology, Karlsruhe, Germany



Figure 2: HyMap Scene (Channels: R=126, G=10, B=1)

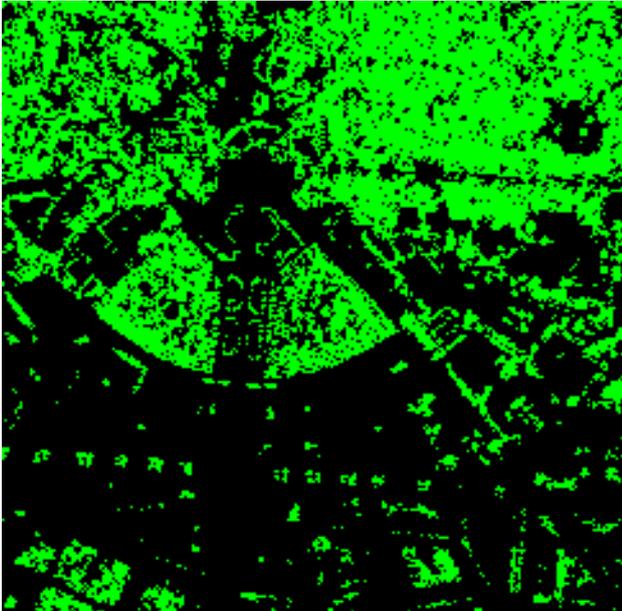


Figure 3: *urban trees* classified in HyMap scene

	QuickBird	HyMap	LiDAR
true-positives	100	98.9	87.1
true-negatives	82.7	63.0	66.9
false-positives	0	1.1	12.9
false-negatives	17.3	37.0	33.1

Table 1: True-positive, true-negative, false-positive & false-negative rates [%] for class *urban trees*

3.2 DETECTING NEAR-NATURAL SURFACE COVER - A COMPARISON OF METHODS

After these first results, we performed classifications with more than one class. The task was to find the coverage of natural and near-natural surface cover. We refer to surface cover types like (urban) forests, lawn, agricultural surfaces and open soils. At first, we classified a HyMap scene on a pixel-based level. After that, we segmented a Landsat scene, using mean-shift segmenta-



Figure 4: QuickBird Scene (Channels: R=3, G=2, B=1)

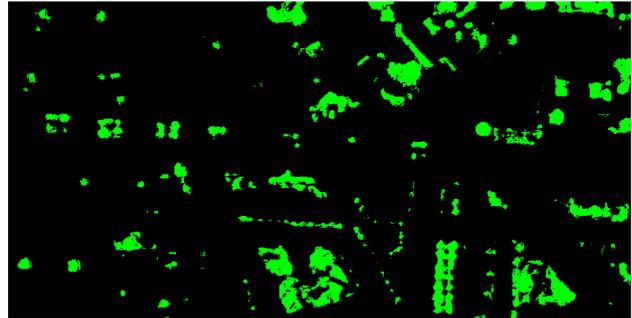


Figure 5: *urban trees* classified in QuickBird scene

tion (Comaniciu, 2002) to present results for segment-based classification. In the HyMap scene shown in Fig.2, we assigned training areas for five landcover classes (*C1:trees*, *C2:watered lawn*, *C3:dry lawn*, *C4:uncovered soils*, *C5:alleys*) We trained and classified for these five classes with one-class SVM (see Fig.9 for result). For comparison, we also classified using standard cost SVM. As cost SVM strongly depends on a representation of all landcover classes in the training data set, we assigned six additional classes for typical urban surfaces (like roofs, streets etc.). We trained according to the one-against-all strategy (Rifkin and Klatau, 2004) and classified with these 11 classes (see table 2). To ensure a visual comparison with the result of one-class SVM in Fig.9, we set the color of the additional classes to black in the classified output (see Fig.8). There are three very obvious

NO.	CLASS	COLOR
C1	urban trees	
C2	lawn (watered)	
C3	lawn (dry)	
C4	uncovered soils	
C5	alleys	
C6	*roofs (red)	
C7	*roofs (bright)	
C8	*roofs (patina)	
C9	*roofs (grey)	
C10	*streets	
C11	*bitumen	

Table 2: Landcover classes for (near-)natural classes in HyMap scene (* additional class for cost SVM, not present in one-class SVM training data set)

differences between the results in Fig.8 and 9. Firstly, the one-class result has a higher amount of scattered rejection class pixels in areas clearly covered by assigned landcover classes (e.g. within the forest). This finding confirms the observations given in table 1 and discussed in section 3.1. The reason for this effect might be, that these pixels are very similar to the pixels considered as non-supporting patterns and labeled -1 during training



Figure 6: LiDAR Scene (Shown : First-Pulse)

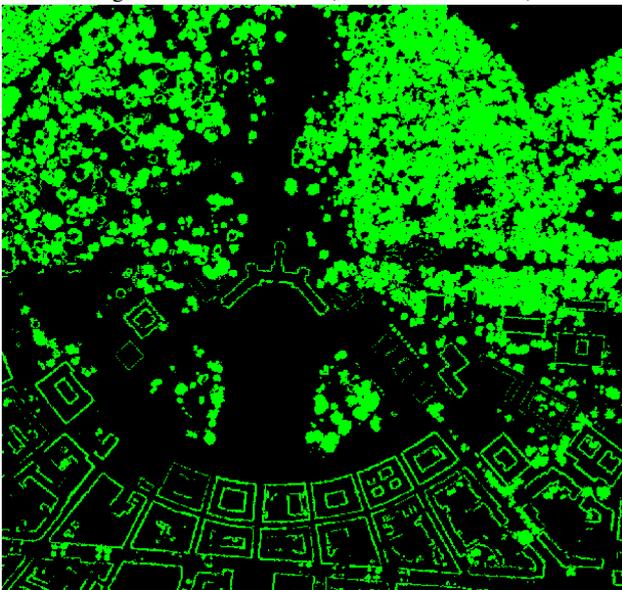


Figure 7: urban trees classified in LiDAR scene

(cf. Fig.1). However, these misclassifications are similar to those discussed in subsection 3.1 and might be related to inhomogeneous ground coverage in the forests as well. Moreover, areas covered by *C5:alleys* come out less clearly in the result for one-class SVM. Obviously, pixels have been confused with the class *C1:trees* due to the inherent similarity of trees in the *C5:alleys* and other trees (covered by class *C1:trees*). For the standard cost support vector machine, LibSVM 2.91 offers probability outputs as described in (Wu et al., 2004) that can be used to decide the class membership in inconclusive cases (i.e. more than one positive class label for a pixel). Unfortunately, they are not provided for one-class SVM yet, so we could not use probability outputs as a tie breaking strategy. Finally, in the result of the one-class SVM much more pixels have been classified as *C1:trees* or *C5:alleys* in the urbanized area (south of the scene). We believe that e.g. for the urban trees besides the main street (running from east to west in the lower part in Fig.2) these are correctly detected. Also for the vegetated courtyards in the south-east, we agree with that finding. Apart from visual evaluation, we evaluated our results

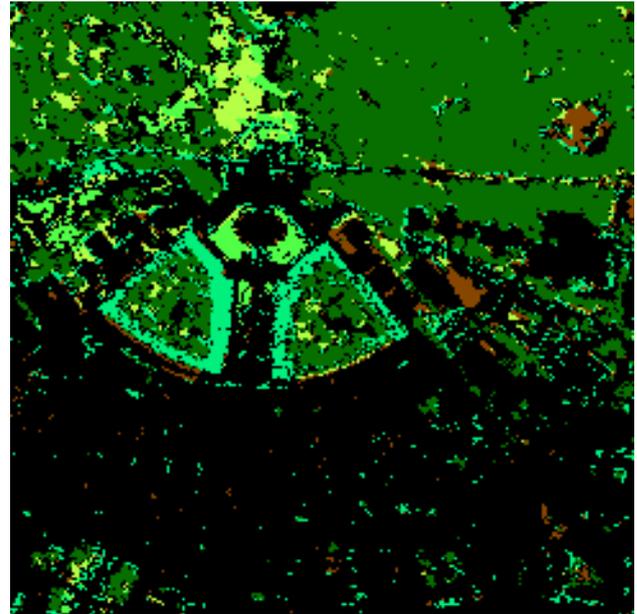


Figure 8: Result of cost SVM classification for HyMap scene (classes marked with "*" in table 2 set to black)

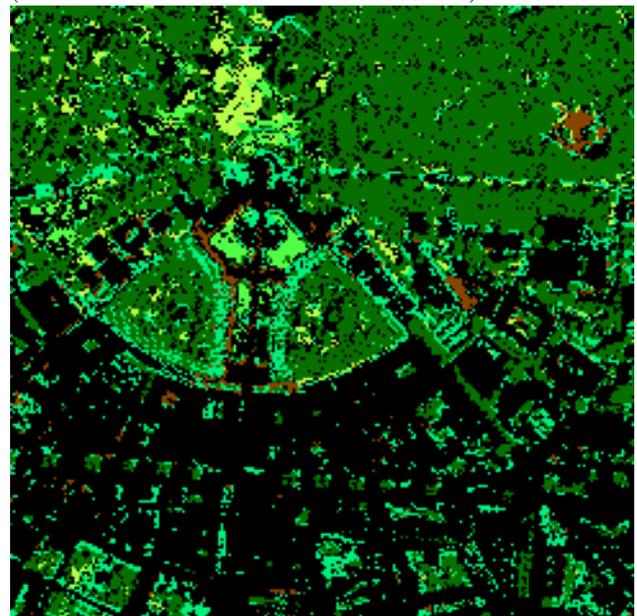


Figure 9: Result of one-class SVM classification for HyMap scene

quantitatively. By assigning control areas based on our knowledge of the city on the screen, we computed confusion matrices. These can be found completely in appendix A. The overall accuracy and the κ -statistics for the one-class result were considerably lower than for cost SVM. Except for the producer accuracy of *C3:dry lawn*, user- and producer accuracies were all lower for one-class SVM. User- and producer accuracy are especially lower for *C5:alleys*. There are several possible reasons for these findings. First of all, the number of training pixels might have been too low. Due to fact, that the computational expense of the leave-one-out error increases significantly with the number of training pixels, we used very small training areas. This caused that too many pixels have been cut of as non-supporting patterns during training. For instance, *C5:alleys* is a class that consists of two different materials, vegetation and gravel. Possibly, gravel pixels

have been considered non-supporting pixels which would explain why so many alley pixels have been classified as *C1:trees* (26 of 62 control pixels). Another problem might be that the leave-one-out error is not the method of choice for one-class SVM (Tran et al., 2005). Lastly, these findings agree with the general experience, that absolute classifiers tend to perform worse compared to relative classifiers. After that, we classified a scene taken from the Landsat sensor in 1993. We segmented the scene using the mean-shift, the result of this is shown in Fig. 10. Just like for HyMap, we



Figure 10: Landsat Scene after segmentation (Channels: R=3, G=2, B=1)

defined 10 training areas for all classes present in the scene which were all used to train the cost SVM. Afterwards we classified the image and set classes *C5* to *C10* to black for visualization. For one-class SVM, only four classes (*C1* to *C4*) were used for training and classification (see table 3). Classification results for cost

NO.	CLASS	COLOR
C1	dark forest	dark green
C2	light forest	light green
C3	agriculture	dark brown
C4	pasture	yellow
C5	*river	black
C6	*lake	black
C7	*city	black
C8	*industry	black
C9	*clouds	black
C10	*shadows	black

Table 3: Landcover classes for (near-)natural classes in Landsat scene (* additional class for cost SVM, not present in one-class SVM training data set)

SVM are given in Fig. 11, whereas the result for one-class SVM is shown in Fig. 12. As one can see, the discrimination between *C1:dark forest* and *C2:light forest* could be accomplished better by cost SVM. One-class SVM failed to recognize *C2:light forest* in many areas and classified it as *C1:dark forest* or assigned it to rejection class. After classification, we again used control areas (see appendix B) to assess the quality of the results. Again, cost SVM achieved slightly higher accuracies than one-class SVM. The reasons for this should be the ones discussed above. How-

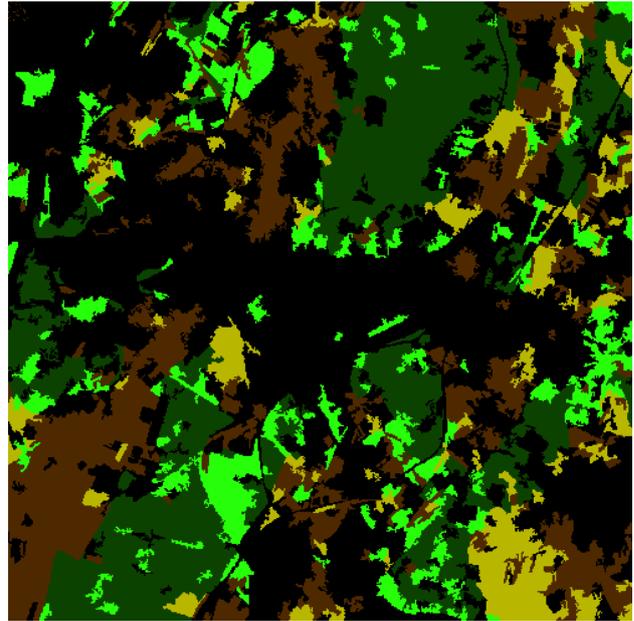


Figure 11: Result of cost SVM classification for Landsat scene (classes marked with "*" in table 3 set to black)

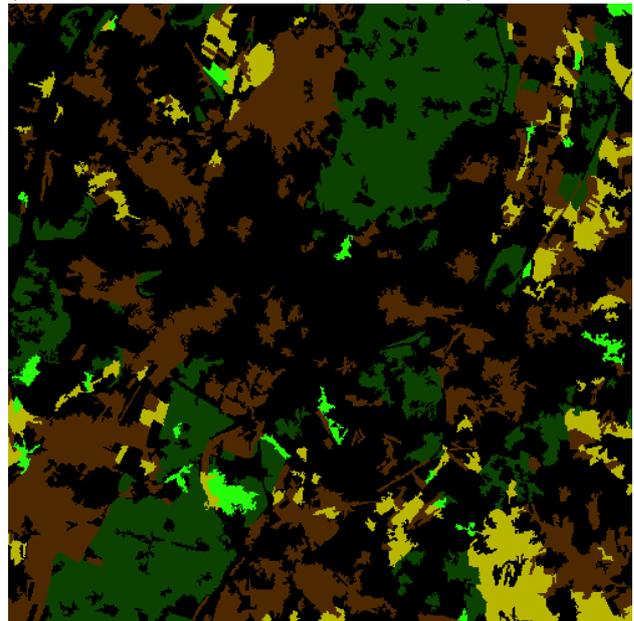


Figure 12: Result of one-class SVM classification for Landsat scene

ever, the advantage of cost SVM in terms of accuracy is not so significant than for the HyMap scene. As one can see, both classifiers have confused some segments of the classes *C1:dark forest* and *C2:light forest*. The reason for this might be the obvious similarity of the both classes, given only seven channels. As could be shown, one-class SVM can be employed in a row of classification steps for various classes to produce an output for more than one class. Accuracy rates are admittedly lower than using cost SVM. Further research is needed to mitigate this loss of accuracy.

4 CONCLUSIONS AND FUTURE WORK

We present our first results for an initial comparison between one-class SVM to standard cost SVM. We compared the two classifiers for a variety of data from different sensors and classification

approaches. One-class SVM can in general be applied to all data types used with reasonably accurate results. However, for HyMap and Landsat, classification accuracy of one-class SVM was lower than for cost SVM when used to detect multiple classes. Furthermore, classes that should be separable could not be separated properly. One hindering for the enhancement of classification accuracy was the lack of probability outputs for one-class SVM. In case of ambiguous decisions, we had to leave pixels in the rejection class, thus lowering accuracy. Another shortcoming was the lack of a utile and precise validation method. Of course, it should not be forgotten that absolute classifiers generally achieve a lower accuracy than relative classifiers. Hence, in our future work, we will compare one-class SVM to other one class methods like support vector domain description (SVDD) (Tax and Duin, 1999) or neural networks. We will also implement the $\xi\alpha\rho$ -estimate to improve performance. Despite the fact, that one-class SVM yielded lower accuracies, it appears to us, that it is a promising approach for remote sensing as it shares the conveniences of standard cost SVM without some of its shortcomings. We will concentrate further research efforts onto mitigating the drawbacks described and the application of one-class SVM to remote sensing tasks.

REFERENCES

Burges, C.-J.-C., 1998. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* 2, pp. 121–167.

Chang, C.-C. and Lin, C.-J., 2010. LIBSVM: A Library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

Chen, P.-H., Lin, C.-J. and Schölkopf, B., 2005. A Tutorial on ν -Support Vector Machines.

Comaniciu, D., 2002. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(5), pp. 603–618.

Fukunaga, K. and Hummels, D.-M., 1989. Leave-one-out Procedures for Nonparametric Error Estimates. *Transactions on pattern analysis and machine intelligence* 2(4), pp. 421–423.

Manevitz, L.-M. and Yousef, M., 2001. One-Class SVMs for Document Classification. *Journal of Machine Learning Research* 2, pp. 139–154.

Rifkin, R. and Klatau, A., 2004. In Defense of One-vs-All Classification. *Journal of Machine Learning Research* 5, pp. 101–141.

Schölkopf, B., Platt, J.-C., Shawe-Taylor, J., Smola, A.-J. and Williamson, R.-C., 2001. Estimating the Support of a High-Dimensional Distribution. *Neural Computation* 13, pp. 1443–1471.

Tax, D.-M.-J. and Duin, R.-P.-W., 1999. Support Vector Domain Description. *Pattern Recognition Letters* 20, pp. 1191–1199.

Tran, Q.-A., Xing, L. and Duan, H., 2005. Efficient performance estimate for one-class support vector machine. *Pattern Recognition Letters* 26, pp. 1174–1182.

Vapnik, V. and Guyon, I., 1995. Support Vector Networks. *Machine Learning* 20, pp. 273–297.

Vapnik, V.-N., 1999. An Overview of Statistical Learning Theory. *IEEE Transactions on Neural Networks* 10(5), pp. 988–999.

Wu, T.-F., Lin, C.-J. and Weng, R.-C., 2004. Probability Estimates for Multi-Class Classification by Pairwise Coupling. *Journal of Machine Learning Research* 5, pp. 975–1005.

APPENDIX

A CONFUSION MATRICES FOR CLASSIFICATIONS OF HYMAP

	C1	C2	C3	C4	C5	U.A.
C1	191	0	0	0	0	100
C2	0	88	1	2	0	96.7
C3	0	20	118	0	2	84.2
C4	0	0	0	90	0	100
C5	0	0	0	0	67	100
P.A.	100	81.4	99.1	97.8	97.1	
O.A.	95.6					
κ	0.94					

Table 4: Confusion matrix for cost SVM classification of HyMap scene (C1-C5: classes (see table 2) U.A.: user accuracy [%], P.A.: producer accuracy [%], O.A.: overall accuracy [%] κ : cappa coefficient)

	C1	C2	C3	C4	C5	U.A.
C1	177	5	0	0	0	97.2
C2	5	89	0	1	3	90.8
C3	3	21	104	2	1	79.3
C4	3	0	0	55	12	78.5
C5	26	0	0	0	36	58.0
P.A.	82.7	77.3	100	94.8	69.2	
O.A.	84.9					
κ	0.8					

Table 5: Confusion matrix for one-class SVM classification of HyMap scene (C1-C5: classes (see table 2) U.A.: user accuracy [%], P.A.: producer accuracy [%], O.A.: overall accuracy [%] κ : cappa coefficient)

B CONFUSION MATRICES FOR CLASSIFICATIONS OF LANDSAT

	C1	C2	C3	C4	U.A.
C1	138	5	0	0	96.5
C2	0	157	0	0	100
C3	0	0	205	0	100
C4	2	0	0	130	98.4
P.A.	98.5	96.9	100	100	
O.A.	98.9				
κ	0.9				

Table 6: Confusion matrix for cost SVM classification of Landsat scene (C1-C4: classes (see table 2) U.A.: user accuracy [%], P.A.: producer accuracy [%], O.A.: overall accuracy [%] κ : cappa coefficient)

	C1	C2	C3	C4	U.A.
C1	138	0	0	0	100
C2	12	54	0	0	81.8
C3	0	0	205	0	100
C4	2	0	0	130	98.4
P.A.	90.7	100	100	100	
O.A.	97.4				
κ	0.9				

Table 7: Confusion matrix for one-class SVM classification of Landsat scene (C1-C4: classes (see table 2) U.A.: user accuracy [%], P.A.: producer accuracy [%], O.A.: overall accuracy [%] κ : cappa coefficient)