

AUTOMATIC REFINEMENT OF TRAINING DATA FOR CLASSIFICATION OF SATELLITE IMAGERY

Torsten Büschenfeld and Jörn Ostermann

Institut für Informationsverarbeitung, Leibniz Universität Hannover
Appelstraße 9A, 30167 Hannover, Germany – {bfeld, ostermann}@tnt.uni-hannover.de

KEY WORDS: Classification, Imagery, Land Cover, Learning, Satellite, Training

ABSTRACT:

In this paper, we present a method for automatic refinement of training data. Many classifiers from machine learning used in applications in the remote sensing domain, rely on previously labelled training data. This labelling is often done by human operators and is bound to time constraints. Hence, selection of training data must be kept practical which implies a certain inaccuracy. This results in erroneously tagged regions enclosed within competing classes. For that purpose, we propose a method that removes outliers from training data by using an iterative training–classification scheme. Outliers are detected by their newly determined class membership as well as through analysis of uncertainty of classified samples. The sample selection method which incorporates quality of neighbouring samples is presented and compared to alternative strategies. Additionally, iterative approaches tend to propagate errors which might lead to degenerating classes. Therefore, a robust stopping criterion based on training data characteristics is described. Our experiments using a support vector machine (SVM) show, that outliers are reliably removed, allowing a more convenient sample selection. The classification result for unknown scenes of the accordant validation set improves from 70.36% to 79.12% on average. Additionally, the average complexity of the SVM model is decreased by 82.75% resulting in similar reduction of processing time.

1 INTRODUCTION

Today, the increasing amount of image data originating from sensors like satellites provides a broad basis for several applications in *Geographic Information Systems* (GIS). Evaluation, however, often demands more manpower than available. Hence, (semi-)automatic systems based on computer vision and machine learning algorithms are of great interest with respect to these applications (Förstner, 2009). With regard to this, methods involving pixel-wise and object-wise classification as well as segmentation are proposed for land cover classification (Helmholz et al., 2010). A comprehensive review in (Mountrakis et al., 2011) shows that a lot of research has been done in the area of support vector machine classification, recently. Supervised methods like SVMs still demand for user interaction in the training process. The selection of samples (i. e. training data) is crucial and directly influences the classification. With real world data, optimal sample selection is neither possible nor practical for the human operator. This particularly applies for small enclosures of dissimilar regions within competing classes (e. g. bushes and tree groups within settlement). These outliers are incorrect samples that consequently reduce classification quality. To address this problem, we propose a method to automatically optimise training data for SVM classification with respect to correctness while concurrently reducing the complexity of the derived model.

1.1 Related Work

Only little research has been done in this area, recently. (Tolba, 2010) describes a method to locate outliers on a low-dimensional manifold that was mapped from a higher dimensional space of training samples. (Xu et al., 2006) directly modify the standard soft margin principle to suppress outliers during training. Surveys on outlier detection methodologies are given by (Chandola et al., 2009) and (Escalante, 2005). A survey of (Hodge and Austin, 2004) points out decision tree methods for supervised machine learning approaches. (John, 1995) uses iterative training and pruning of misclassified labels to keep inliers. (Brodley and Friedl, 1996) extend this idea for land cover classification. A

consensus voting scheme is employed to filter results from an ensemble of classifiers to eliminate mislabelled samples. For SVM classification, further methodologies can be roughly categorised into *online learning* and *batch learning* based techniques.

In online learning, samples are added one at a time. This can be utilised for *active learning* where new samples are consecutively queried for annotation by the algorithm, while updating the trained model. While primarily being focused on large data sets, the algorithm queries for critical or important samples and thus avoids outliers. In (Laskov et al., 2006), *incremental support vector learning* is utilised for active learning. (Li and Sethi, 2006) propose confidence-based active learning to optimise the training, by only processing uncertain samples (which hold most information).

In batch learning, all samples are available at once. The main focus of several proposed approaches lies on training set reduction while keeping cluster boundaries. (Bakır et al., 2005) first train independent SVM on subsets of the training data. These are used to classify the training data and then to identify uncertain samples and discard all others. In a second step, a final SVM is trained on the remaining samples. (Wagstaff et al., 2010) propose to use probability estimates of a low complexity SVM to determine uncertain regions for high-accuracy classification. (Wang et al., 2005) describe two approaches based on confidence and Hausdorff distance to remove unneeded samples.

1.2 Contribution

As stated in the previous paragraph, most batch learning based methods focus on training set reduction. Samples are removed, such that only those at boundaries remain, since these will most likely lead to support vectors. However, uncertain samples at boundaries might also originate from outliers that unnecessarily increase the complexity of the model. In our approach, we pick up the idea of using uncertainty information, but use it to remove outliers.

In an iterative training–classification scheme, we identify and remove misclassified samples. In contrast to outlier detection meth-

ods like those in (John, 1995) or (Brodley and Friedl, 1996), our approach takes uncertainty of the executing classifier into account. Uncertainty of classified samples is evaluated by employing probability estimates (Wu et al., 2003). Highly uncertain samples, which primarily represent transitions between distinct textures in the image, are removed to further improve the refinement. An underrated problem of sample removal is degeneration of classes. In order to avoid underrepresented classes to vanish from training data, we describe a robust stopping criterion for the iterative refinement. Our experiments show that manual sample selection can be kept practical, since small enclosures of competing classes are sorted out automatically. We show that both, quality and processing time of classification improve even with low quality training data. It is important to note that, in the first instance, outliers are removed which would otherwise lead to a wrong model and thereby unnecessarily increased model complexity. It is not the primary aim to search for the most trivial samples in order to simplify the model.

In the following section, we will first show the methodology of the proposed approach in detail. Subsequently, the experimental setup is outlined. Finally, results are shown which illustrate the benefit of our approach.

2 ITERATIVE TRAINING DATA REFINEMENT

Without loss of generality, examples and explanations will refer to a four class problem. Training samples were chosen from the following classes

- Cropland/Grassland (c_1)
- Forest (c_2)
- Industry (c_3)
- Settlement (c_4)

which nearly cover the entire area of the classified scenes. Figure 1 shows a typical, manually selected training set (T01) representing the aforementioned classes with patches of c_4 , c_3 , c_1 , c_2 , c_4 from left to right. Additionally, class membership is represented by a label image.

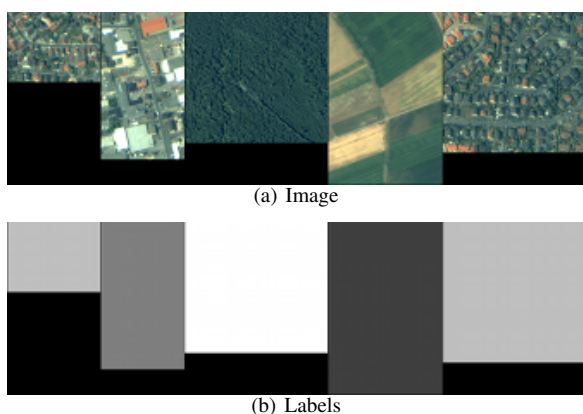


Figure 1: Training set T01 (a) with the accordant labels (b). Black are invalid regions not used as samples, grey values represent labels. Classes from left to right: c_4 , c_3 , c_2 , c_1 , c_4

2.1 Base System

The iterative refinement proposed in this paper is built around a common base system which consists of feature extraction and training/classification as explained in the following subsections.

2.1.1 Feature Extraction Radiometric and statistical features are extracted within a local $N \times N$ neighbourhood for each pixel. This is done for all available spectral image bands (e. g. R, G, B, NIR) and in different levels of detail and finally composes the feature vector. Here, the feature vector dimension d_f , considering two scales, two features and four bands, adds up to $d_f = 16$.

2.1.2 SVM Training and Classification For training and classification, all feature vectors are handed to an SVM (Vapnik, 1998). (Burges, 1998) gives a comprehensive introduction to support vector machines. Our implementation is based on LIBSVM (Chang and Lin, 2001). For our tests, the common *Radial Basis Function* (RBF) kernel was used. No explicit parameter tuning was done, general parameters with respect to robust classification in all our scenes were chosen. This implies a good generalisation biased against overfitting.

2.2 Iterative Refinement

The general method of our approach is shown in Figure 2. The exemplary training data set T01 consists of five manually selected sample patches of the four classes $c_1 - c_4$ and their labels (Figure 1). Larger images of the pictograms for intermediate results are depicted in Figure 3.

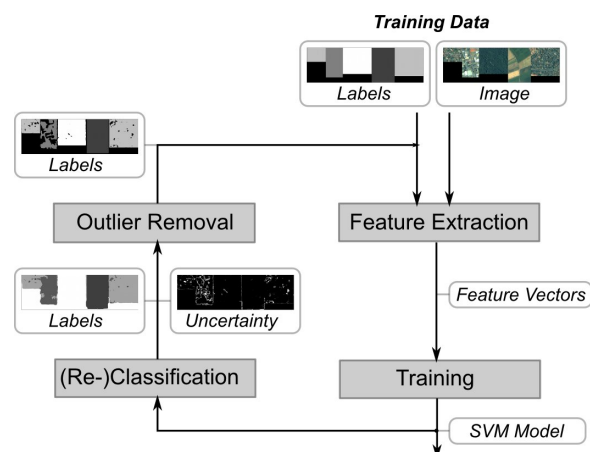


Figure 2: Training data refinement scheme. Larger images of the pictograms showing intermediate results are depicted in Figure 3

The iterative approach consists of four main parts and is repeated until convergence:

1. Feature Extraction
2. Training
3. (Re-)Classification
4. Outlier Removal

At first, features are extracted from given samples as stated in section 2.1.1. Feature vectors are then passed to the SVM for training, leading to the SVM model. In typical applications, this model is directly used for classification of unknown scenes.

In this paper's contribution, additional steps are introduced to optimise the model. After training, a reclassification of the training data is done (Figure 3(a)). As can be seen, there are misclassifications compared to the original labels. Due to robust, generalising parameters, the training is not prone to overfitting. Hence, misclassified samples can safely be treated as outliers and exempted from further consideration. Additionally, uncertain samples are excluded as well. Therefore, probabilities for class memberships are estimated by pairwise coupling (Wu et al., 2003) for each

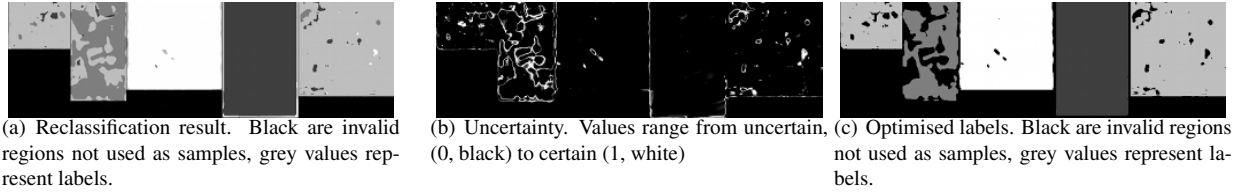


Figure 3: Intermediate results of the refinement process shown in Figure 2. Reclassification result (a) and uncertainty (b) are used to refine the label image resulting in (c).

sample. The *first best to second best* ratio defines the uncertainty U of each sample (Figure 3(b)). Given a threshold T_u , uncertain pixels with $U > T_u$ are excluded. This modified label image (Figure 3(c)) serves as basis for the next iteration.

2.2.1 Sample Selection Figure 4 shows how samples are selected from label image. Grey represents a label of a certain class while black marks invalid samples. These might originate from the initial label image or were removed during the refinement process. The boxes indicate the $N \times N$ neighbourhood used for feature extraction, a dot marks the center position. As shown, samples are only used in case S_3 where the entire local neighbourhood does not cover any invalid regions.

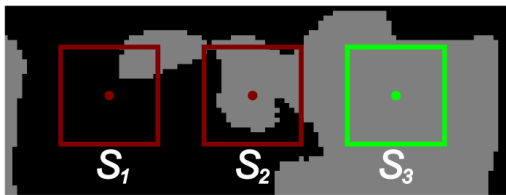


Figure 4: Training sample selection. Boxes $S_1 - S_3$ indicate $N \times N$ neighbourhood from feature extraction with center (dot). Class labels are depicted in grey, invalid area is black. S_1 and S_2 are not valid due to invalid center position and neighbourhood, respectively. Only S_3 is a valid sample.

2.2.2 Convergence Characteristics Gradually removing outliers entails the risk of degenerating classes. A class might be underrepresented due to a low number of samples or many outliers. Additionally, low separability in feature space can lead to a heavy bias towards other classes. Hence, it is essential to keep track of characteristic details of the refinement process. In our case, one such detail appears at the beginning of the iteration.

In Figure 5, the number of samples for four different classes is plotted over 75 iterations. The local minimum at iteration start is significant as it already indicates a converging class. Usually, the initial selection of samples contains a large number of outliers. A lot of them are removed during the first iteration. Therefore, the number of samples drops. Afterwards, a new SVM model is trained based on the improved sample set. This leads to a better representation of the data and consequently to less uncertainty and outliers. This implies, that the number of removed samples is lower, i. e. the number of samples increases.

Accordingly, two factors can cause a missing minimum at iteration start. First, the initial set doesn't contain any or only few outliers. In this case, the number of samples will hardly drop at all. Second, the number of samples monotonously drops. This indicates, that the SVM model did not improve for this class, causing degeneration.

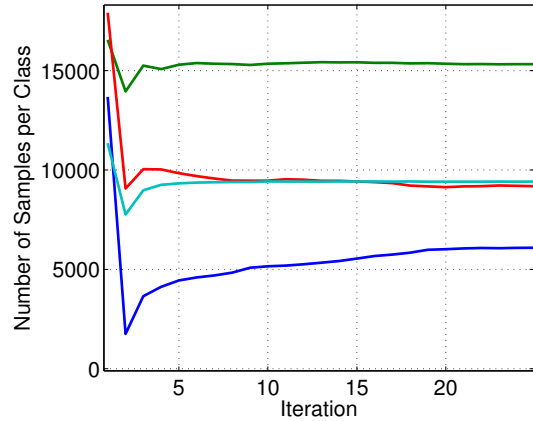


Figure 5: Convergence characteristics showing the number of samples per class (class is colour coded) for 25 steps of iteration. Each of the four graphs belongs to one of the classes $c_1 - c_4$. A significant detail is the minimum at the beginning of the iteration.

3 EXPERIMENTS

In the following, the setup for experiments is outlined, which consists of image data, training subsets, and validation subsets with reference. At last, the evaluation procedure is described.

3.1 Image Data

Input data for our tests originate from the IKONOS satellite. The images are ortho-rectified and consist of the four spectral bands red (R), green (G), blue (B), and near infra red (NIR) with 8 bit colour depth per band. The spatial resolution is 1 m. The scenes cover areas from Hildesheim/Germany and Weiterstadt/Germany.

3.2 Validation Set and Reference

Representative scenes from city and environs of Hildesheim and Weiterstadt were manually referenced with pixel accuracy, both originating from the same IKONOS image data. Classification result and reference are compared pixel-wise to obtain the rate of correct detection. The reference features more classes than our classification system. Thus, classes are mapped if necessary, e. g. *inner city* and *suburban* areas are combined to *settlement*. In many applications, line objects like streets are treated separately from area objects since training with the same classifier is not efficient. Hence, line objects are ignored in our evaluation to not falsify the results.

3.3 Training Sets

Training data for each class was manually selected with a graphical tool. The data originates from the same geographical region, yet not overlapping with the according validation set. We

Set	Scene	Nr. of Classes	Sample Selection
T01 – T03	Weiterstadt	4	manual
T04 – T05	Weiterstadt	4	GIS
T06 – T07	Hildesheim	6	manual
T08 – T11	Hildesheim	4	manual

Table 1: Training data sets from Weiterstadt/Germany and Hildesheim/Germany. Sample selection for two scenes was automatically done using a GIS.

extracted several training sets with different characteristics like sample size and quality. Additionally, map data from a German GIS, the ATKIS¹ was rasterised with the same spatial resolution as the image data and mapped to accordant labels. Hence, it was used to automatically generate training data. Since ATKIS contains errors and is generalised at a higher level of detail compared to our manually referenced scenes, it is a demanding stress test for our system. Table 1 lists all training data sets. For scene Hildesheim, two sets with six classes were created.

3.4 Evaluation

Each training data set of Table 1 was optimised with the proposed approach. For evaluation, four characteristic values are extracted for each iteration step:

1. Number of samples per class
2. Number of support vectors per class (representing model complexity)
3. Contour length of labels (representing training selection convenience)
4. Classification result in validation set with respect to reference

While items 1 – 3 are gathered within the training process, the overall classification result is not available for training refinement. It is calculated by using the SVM model from the current iteration step to classify the validation set. Thus, quality improvement is shown throughout refinement and coherence to training characteristics can be identified. Even though a stop criterion is used, it was disabled for graphs displayed in this section. A sufficient number of 75 iterations was used for evaluation to show the characteristics of the refinement process.

The contour lengths of original and optimised labels are compared as a measure for convenience of training data selection. The longer the contours are, the more effort has to be put into sample selection by a human operator. Hence, a higher contour length after refinement equals more convenience for initial sample selection. All contours are one pixel wide. Contour pixels were simply counted using an 8 neighbourhood. Since contours of manually selected labels are oriented horizontally and vertically, this method is perfectly reasonable as a lower boundary which does not abet our approach.

4 RESULTS

Figure 6 shows a characteristic result of the refinement process (set T02) with respect to evaluation items defined in section 3.4. The four graphs illustrate the convergence of the process. In Figure 6(a), the number of samples of all but one class directly drop to the characteristic minimum as discussed in section 2.2.2. The other class of type Cropland/Grassland (c_1) does not contain

¹Amtlich topographisch-kartographisches Informationssystem (Authoritative Topographic Cartographic Information System)

Set	Result / %	Refined Result / %	Absolute Improvement / %
T01	66.75	73.18	6.43
T02	71.33	79.49	8.16
T03	71.19	78.33	7.15
T04	61.63	74.36	12.73
T05	61.90	72.50	10.60
T06	77.23	85.14	7.91
T07	67.28	79.37	12.09
T08	75.80	81.53	5.74
T09	81.54	84.58	3.04
T10	65.06	83.40	18.34
T11	74.24	78.39	4.15
Avg	70.36	79.12	8.76

Table 2: Classification results of validation set for training data sets T01 – T11, see Table 1 for reference.

a significant number of outliers, resulting in immediate convergence. This is also due the homogeneity of this class (see c_1 in Figure 3(a) for instance). As can be seen, the number of support vectors rapidly decreases which argues for a lower complexity of the model (Figure 6(b)). The overall contour length (Figure 6(c)) increases as expected due to outlier removal. Furthermore, a local maximum stands out significantly. It relates to the characteristic local minima of the number of samples but does not turn out to be as consistent. The last plot depicts the overall classification of the validation set for each iteration. Here, the initial training set leads to 71.33% correct classifications. It increases nearly monotonously until convergence, with a correct rate of 79.49%. This trend is observable in all test sets. The classification results are listed in Table 2.

From scene Weiterstadt, test sets T04 and T05 show the biggest improvement. These were trained with the GIS data which is very prone to outliers. The highest improvement was achieved for a test set of scene Hildesheim (T10). It is a very small test set which does not represent the scene quite well. These observations validate our approach of refining poor training data resulting in an average improvement from 70.36 % to 79.12 %

The impact on a human operator is estimated by evaluation of contour length as described in section 3.4. On average, an initial sample selection with 35.6 % shorter contours is compensated if training data is refined.

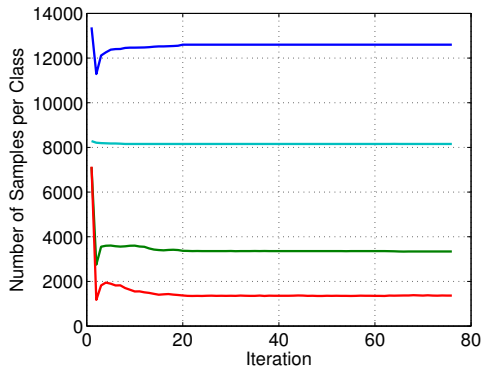
Additionally, processing time for classification is extensively reduced on account of a less complex SVM model. The number of support vectors which directly correlates to processing time is reduced by 82.75 % on average.

4.1 Comparison of Sample Selection Strategies

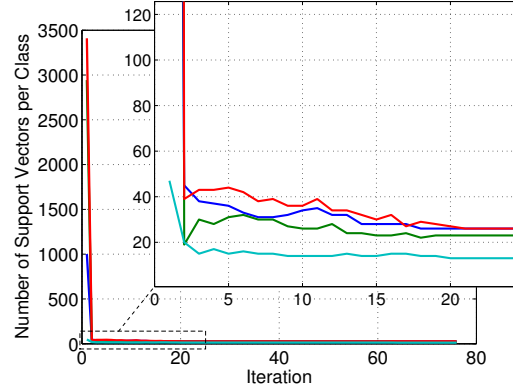
A comparison of sample selection strategies is given in Figure 7. Results clearly support our approach which outperforms alternative strategies. Allowing only clean samples (S_3 , see Figure 4) and removing uncertain samples leads to a considerable improvement of the classification result. Here, test set T01 was optimised with different strategies for sample selection, tested with and without removal of uncertain samples.

4.2 Comparison to Slack Variables

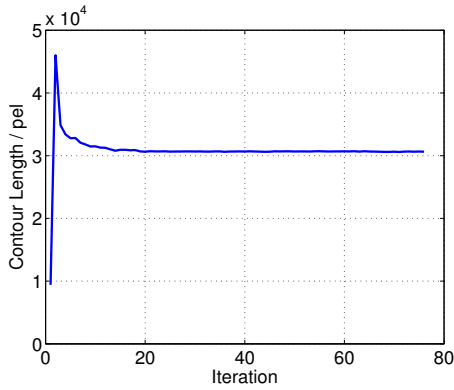
To handle outliers in training data, slack variables were introduced to SVMs (Cortes and Vapnik, 1995), which allow for a certain amount of feature vectors to be located on the wrong side of the hyperplane. Costs C are used to these penalise outliers. In



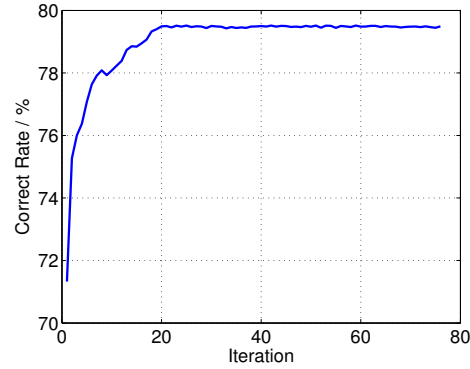
(a) Number of samples per class (class is colour coded) of training set T02 for each iteration.



(b) Number of support vectors per class (class is colour coded) of training set T02 for each iteration with magnification of important section.



(c) Contour length of trained label image of training set T02 for each iteration.



(d) Overall classification result of validation set for each iteration.

Figure 6: Results of the refinement process for each of 75 iterations. Four characteristics as discussed in section 3.4 are plotted.

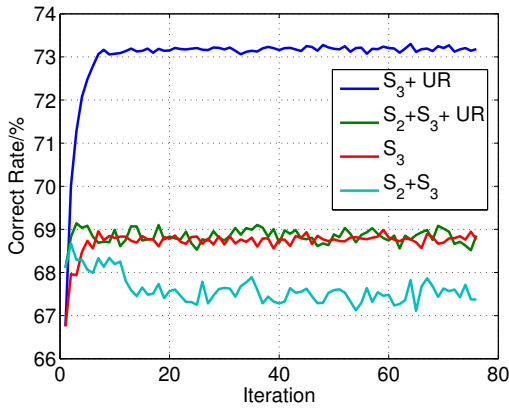


Figure 7: Comparison of sample selection strategies and impact of uncertainty information. S_2 and S_3 refer to sample selection from Figure 4. +UR indicates removal of uncertain samples from training set.

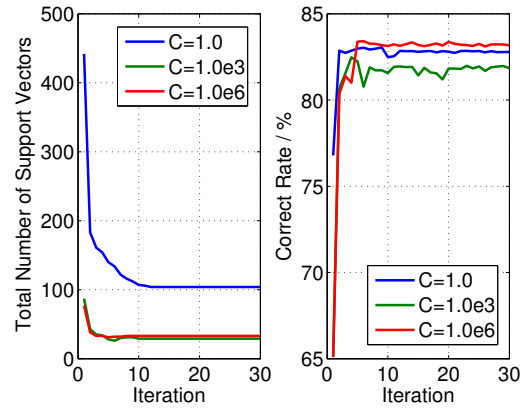
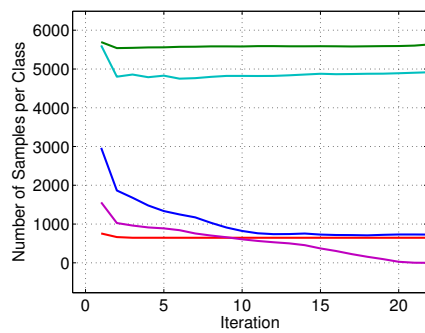


Figure 8: Comparison for different values for cost variable C when classifying with an SVM using slack variables.

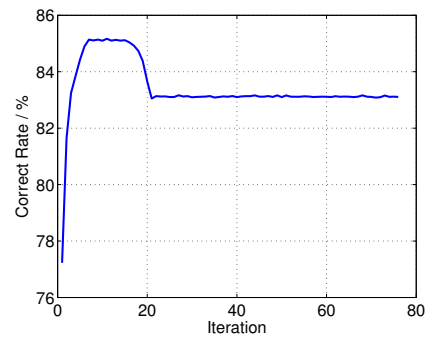
4.3 Class Degeneration

Figure 8, the classification result and the total amount of support vectors is plotted for varying values of C for set T10. This set clearly shows the important characteristics: Allowing more outliers ($C = 1.0$) helps in improving the initial classification result. Nevertheless, the model can still be refined by our approach. Additionally, the model complexity is reduced. This applies even more for higher values of C .

Within our setup, three test sets show degeneration of classes (T05, T06, T10). T06 articulately exhibits important characteristics. (Figure 9). Samples for three of the five classes do not change significantly. The number of samples for the other two rapidly drops. Their monotonous decline indicates that newly derived SVM models do not improve. This is where the stopping criterion steps in. As soon as the relative number of samples compared to the initial value reaches a threshold T_s without the



(a) Number of samples per class (class is colour coded) for each of the first 23 iterations. The graph of the sixth class of this set is omitted for better visualisation, it converges immediately.



(b) Overall classification result for each iteration.

Figure 9: Degeneration of classes. (a) shows the important section for the number of samples per class. (b) depicts the classification result.

typical local minimum in the number of samples, the iteration is stopped (iteration 9 in the example of Figure 9). Here, this holds true for both classes, even though only one degenerates. Threshold T_s is set to 30 %. However, results are not too sensitive to the choice of T_s as iteration reliably stops before degeneration and rapid loss in correct classification rate occurs.

5 CONCLUSIONS

In this paper, we presented a general method for automatic refinement of training data for SVM classification. Refinement was done with respect to sample selection convenience. We have shown, that by incorporating uncertainty into an iterative outlier detection, correct rate of classification as well as complexity of the derived model can be significantly reduced. The correct classification rate was improved from 70.36 % to 79.12% on average. The relevance for manual sample selection for a human operator is given by compensating 35.6 % shorter contours. Additionally, since poor training data is compensated, automatic training with samples from sources like GIS becomes practical. The processing time for classification is narrowed down considerably as the complexity of the SVM model is reduced by 82.75 % on average. Since internals of the classifier are not modified, the refinement is conceivable for other margin based classifiers if probability estimates are available.

REFERENCES

- Bakır, G. H., Bottou, L. and Weston, J., 2005. Breaking svm complexity with cross training. In: In Proceedings of the 17 th Neural Information Processing Systems Conference.
- Brodley, C. E. and Friedl, M. A., 1996. Identifying and eliminating mislabeled training instances. In: In AAAI/IAAI, pp. 799–805.
- Burges, C. J., 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2, pp. 121–167.
- Chandola, V., Banerjee, A. and Kumar, V., 2009. Anomaly detection: A survey. *ACM Computing Surveys* 41, pp. 1–58.
- Chang, C. C. and Lin, C. J., 2001. LIBSVM: a library for support vector machines.
- Cortes, C. and Vapnik, V., 1995. Support-vector networks. In: *Machine Learning*, pp. 273–297.
- Escalante, H. J., 2005. A comparison of outlier detection algorithms for machine learning. In: *Communications in Computing*.
- Förstner, W., 2009. Computer vision and remote sensing - lessons learned. In: *Photogrammetric Week 2009*, pp. 241–249.
- Helmholz, P., Becker, C., Breikopf, U., Büschenfeld, T., Busch, A., Grünreich, D., Heipke, C., Müller, S., Ostermann, J., Pahl, M., Vogt, K. and Ziem, M., 2010. Semiautomatic quality control of topographic reference datasets. In: *ISPRS Commission 4 Symposium*.
- Hodge, V. J. and Austin, J., 2004. A survey of outlier detection methodologies. *Artificial Intelligence Review* 22(2), pp. 85–126.
- John, G. H., 1995. Robust decision trees: Removing outliers from databases. In: *In Knowledge Discovery and Data Mining, AAAI Press*, pp. 174–179.
- Laskov, P., Gehl, C., Krüger, S. and Müller, K. R., 2006. Incremental support vector learning: Analysis, implementation and applications. *Journal of Machine Learning Research* 7, pp. 1909–1936.
- Li, M. and Sethi, I. K., 2006. Confidence-based active learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 28(8), pp. 1251–1261.
- Mountrakis, G., Im, J. and Ogole, C., 2011. Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing* 66(3), pp. 247–259.
- Tolba, A. S., 2010. Manifolds for training set selection through outlier detection. In: *Proceedings of the IEEE International Symposium on Signal Processing and Information Technology, IS-SPIT*, pp. 467–472.
- Vapnik, V. N., 1998. *Statistical learning theory*. 1 edn, Wiley.
- Wagstaff, K., Kocurek, M., Mazzoni, D. and Tang, B., 2010. Progressive refinement for support vector machines. *Data Mining and Knowledge Discovery* 20, pp. 53–69.
- Wang, J., Neskovic, P. and Cooper, L. N., 2005. Training data selection for support vector machines. In: *ICNC 2005. LNCS*, Springer, pp. 554–564.
- Wu, T.-F., Lin, C.-J. and Weng, R. C., 2003. Probability estimates for multi-class classification by pairwise coupling. In: *Neural Information Processing Systems*.
- Xu, L., Crammer, K. and Schuurmans, D., 2006. Robust support vector machine training via convex outlier ablation. In: *National Conference on Artificial Intelligence*.