Comparison of neural classification algorithms applied to land cover mapping

Christian Thiel[1], Ferdinando Giacco[2], Friedhelm Schwenker[1] and Günther Palm[1]

[1] Institute of Neural Information Processing, University of Ulm, 89069 Ulm, Germany, christian.thiel@uni-ulm.de
[2] Department of Physics, University of Salerno, Via S. Allende, 84081 Baronissi (SA), Italy giacco@sa.infn.it

Abstract. We compared the performance of several supervised classification algorithms on multi-source remotely sensed images. Apart from the Multi-Layer Perceptron, K-Nearest-Neighbour and Radial Basis Function network approaches, we looked more in detail at the Support Vector Machine classifier, which recently showed promising results in our setting. In particular, it is able to provide meaningful answers for the analysis of mixed pixels. They correspond to areas on the ground that comprise more than one distinct class, representing a major challenge for the interpretability of the final land-cover maps. To assess their impact, we performed a rejection-based analysis, allowing classifiers to refuse answers on pixels they can not associate mainly with one class.
The experimental results lead to the conclusion that the 1vs1 SVM approach with a linear kernel (using Bradley-Terry coupling) has to be preferred over all other classification algorithms examined, both in terms of accuracy as well as ease of visual interpretation.

1 Introduction

Recently, a new kind of algorithms to produce land cover mappings has caught the eye of the remote sensing community: Support Vector Machines [1,2]. These SVMs calculate distances between samples in a higher-dimensional so-called kernel space, and therefore have an advanced ability to discriminate between classes. Also, they are especially known to work with a limited number of training data, which is especially important in our context, since the labelling of ground pixels is expensive. To evaluate the usefullness of SVMs in land cover mapping, we compared their performance to a number of well-known supervised classification algorithms: Radial Basis Function Networks, Multi-Layer Perceptrons, and K-Nearest-Neighbour. A special focus of this work is to explore the behaviour of the approaches when presented with so-called mixed pixels. Those are areas on the ground which can not be subdivided into smaller parts due to main sensor resolution, but which comprise several different land cover classes. They are abundant in remote sensing applications and present a challenge because assigning them to one class in the map leads to misinterpretations [3]. Moreover, few classifiers are capable of producing correct mixed answers. Because an encompassing analysis of the nature of the mixed answers is not feasible, we performed a rejection-based study. That is, the algorithms were allowed to reject an increasing number of test pixels, related to the degree of mixing in the answer to each pixel. Looking at the resulting accuracy on the not-rejected test set, and the distribution of rejected pixels, we were able to draw conclusions on the suitability of the different classification algorithms. The most important result is the superiority of the 1vs1 linear SVM approach, when Bradley-Terry coupling is used instead of the usual simple voting.

2 Satellite data

The area of interest is a coastal plain in the southern part of Italy, located in the alluvial plain of the Salerno Gulf. The area is densely inhabited for the fertility of the land since Greek-Roman times. Land use is primarily agricultural, but during the last sixty years an urbanisation phenomenon occurred, giving rise to a very indented and complex landscape. Consequently, the principal types of land covers are agricultural fields (both fallow fields and crop covered ones), rural fabrics (greenhouses), sea water, a coniferous wood strip along the coastline, and small urban areas made up of discontinuous fabric mixed with vegetation[3].

For this study, we take into account two types of multi-spectral satellite imagery[4]: one captured by the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) on NASA's Terra satellite, and the other captured by IKONOS 2, a commercial earth observation satellite which offers high resolution imagery.

From the Aster data, we used an image taken in winter 2004. One scalar value was extracted for each of the nine bands chosen: going from the visible (bands 1-3, 15 m/pixel resolution) to the short wave infrared region (bands 6-9, 30 m/pixel resolution) of the electromagnetic spectrum. All data was resized to a resolution of 15 m/pixel.

Secondly, textural features extracted from Ikonos images were introduced. This was done in order to add intra-pixel spatial information to the Aster spectral data. Our textural characteristics are based on the well known Grey-Level Co-occurrence Matrix (GLCM), widely used in land-cover mapping [5]. The GLCMs were computed on two different Ikonos data sets: the panchromatic band (a black and white imagery of 1 m/pixel resolution, sensitive to all visible radiation) and the band ratio between near-infrared and red (4 m/pixel resolution, resized to 1 m/pixel), which in remote sensing literature is considered as a reasonable way to avoid shadows. A moving window of 15x15 Ikonos pixels is used in the computation of the GLCM, since a window of such dimensions covers the same spatial area as one Aster pixel. Among the several statistical measures which can be extracted from the GLCM to describe specific textural characteristics of

[3] For our study, we hence defined the following $l = 7$ classes: vegetated agricultural fields, buildings, pine forest, urban green, sea shore, not vegetated agricultural fields, and water.

the image [6], we chose the following two: the Correlation function computed on the Ikonos panchromatic band and the Energy function computed on the Ikonos band ratio (see above). These particular choices for the statistical measures provide the best classification performances for our dataset.

Summing up, our data vectors are made up of 11 components, the first 9 standing for the spectral information (taken from Aster bands) and the last 2 representing textural measures extracted from Ikonos images.

3 Rejection of pixels

The supervised classifiers we consider take a training set T, consisting of pairs of training samples z in the feature space $\mathbb{R}^N$ with associated training labels y that detail to which of the l classes the sample belongs:

$$T = \{(z_\mu, y_\mu\ )|\ \mu = 1, \ldots, M,\ y_\mu \in \{1, \ldots, l\}\ \}$$

Being trained on the data set T, a classifier can now be presented with an hitherto unseen sample, and will output an answer $o \in [0,1]^l$, $\sum_{d=1}^{l} o_d = 1$, that reflects to what degree the classifier thinks the sample belongs to each class. This answer can be interpreted as a probability.

In a rejection setting, a classifier is allowed to reject a test sample presented to him, that is to refuse to take a decision about the class of the sample. In classical machine learning settings, this is done to increase the classification accuracy on the not-rejected samples. An answer which the classifier says he is not sure about, where this can be deducted from the structure of the answer [7], or where different classifiers disagree[8], is rejected. In a remote sensing context, there is a problem with so-called mixed pixels. For those, the classifier's output o is not predominantly in favour of one class, but gives probabilities to multiple ones. These pixels can not be marked as clearly belonging to a certain class. An example might be an area containing both buildings and vegetation. The goal is to detect and flag mixed pixels.

The rejection method we decided to use is based on the maximum probability entry of the answer o to a sample:

$$\texttt{reject if } (\max_d o_d < \mathit{threshold}) \tag{1}$$

4 Classifiers

The goal of this article is to compare the performance of several supervised classification algorithms on satellite images. In the following, the classifiers we used will be presented briefly. For an in-depth introduction to each of the architectures, as well as their relations, please refer to [9].

The Multi Layer Perceptron (**MLP**, [10]) is a network of perceptrons, in our case with an input layer, a hidden layer, and an output layer which yields the mixed answer for each pixel, the probabilities. In the hidden layer with a total of 70 neurons the hyperbolic tangent sigmoid transfer function is applied, in the output layer a linear transfer function is used. Optimisation was accomplished using backpropagation [11] with a maximum of 20 epochs.

When presented with a test sample x the weighted K-Nearest-Neighbour algorithm (**KNN**, [12]) employed looks for the $K = 200$ nearest samples n_i in the training data, and answers with a weighted mean of their labels. The weighting function w is a Gaussian:

$$w_i(x) = exp\left(-\frac{\|x - n_i\|^2}{\sigma}\right)$$

The spread $\sigma > 0$ was set for each sample in relation to the median of all its K neighbour distances.

We also used the very powerful Radial Basis Function Network classifier (**RBF**, [13]), a network reconstructing the decision surfaces using Gaussian nodes. The number of kernels in the RBF network was set to 69 using a simple heuristic formula, their position determined by running a fuzzy c-means algorithm [14] on the training data. The individual variance of the kernels was set based on an an experimental observation of Breiman [15], which allows to have only one parameter to optimise for the whole net.

Support Vector Machines (**SVMs**) have become a popular method in pattern classification and were originally developed for the discrimination of two-class problems [16]. They work by projecting the data into a higher dimensional feature space using kernel functions, then finding the hyperplane that separates the two classes while providing the widest margin in which no sample points lie (see [17] or [9] for an introduction). The kernels we used are of linear nature, or again Radial Basis Functions. Being in a multi-class setting, we had to pay special attention to the possibilities of extending the originally binary SVMs. There has been quite some research activity in this field recently [18], and architectures like 1vs1 or 1vsAll are widely used nowadays (see [19] for a comparison). In a recent work of ours [20], we dealt with the issue of accepting, and more importantly producing, soft labels. Below, we will briefly present the solutions we decided to explore in the current application, and elaborate a bit on the 1vs1 case where no ready procedure existed.

In the current case, we have samples of $l = 7$ different classes in the data. The 1vsAll approach (also called One vs Rest) now builds l different SVMs, each of which is able to separate one specific class from all the others. Presented with a new sample x, each SVM$_i$ will answer with the distance $d_i(x),\ i = 1 \ldots l$, that this sample has to its hyperplane. To transfer this distances to soft output answers o_i that can be interpreted as probabilities, we make use of a sigmoid function (*Fermi* like), as recommended in [21]:

$$o_i(\ d_i(x)\ ) = 1/(1 + \exp(-A_i^T d_i(x) + B_i)), \quad i = 1, \ldots, l$$

The parameters $A_i \in \mathrm{I\!R}^N$ and $B_i \in \mathrm{I\!R}$ are estimated for each SVM$_i$ to minimise the mean squared error on the training data between the original label and the sigmoid output, using a batch gradient descent technique. Details can be found in our article [20] mentioned above.

The solution is not so straightforward in the 1vs1 approach. Here, a SVM$_i$ is built for every pair of classifiers (resulting in $\binom{l}{2} = 21$ machines in our case). To get the desired l-dimensional soft output, the technique employed in most cases today, for example in [1], is as follows: using an indicator function, transform each of the answers d_i into a vote for one of the two classes distinguished by the current machine i. Then sum those votes per class, and normalise[4] the resulting soft label. This method does not have a bad performance, but some limitations (detailed in section 5) as it does deliberately not take into account the distance information provided by the values d_i. To heal this issue, one can proceed similarly to the 1vsAll case and use a sigmoid, preferably with the same parameter for all pairs, to transfer the distances to soft answers, which can then be summed up and normalised.

But even then, the class-pair information provided by the SVM$_i$ is not used. This can be addressed by the technique of pairwise coupling, based on the statistical Bradley-Terry model [22]. It uses initial estimations for the pairwise probabilities (derived via sigmoid from the distances d_i), and in an iterative process produces soft labels that take into account the coupled distance information. In our experiment, this results in answers with a much lower entropy than without coupling. An introduction to and many interesting theoretical conclusions on pairwise coupling in classification can be found in [23].

A short note on choosing the (universal) scaling parameter for the sigmoid transfer function in the 1vs1 SVMs: unfortunately, there is no perfect choice valid for all data sets. To really get those 2-3 percent points in accuracy increase, and ensure an appropriate hillyness of the coupled answers, the parameter has to be determined experimentally. In our case, 2 turned out to be the best choice. On the other hand, when looking at the rejection versus accuracy curves, the plots for different choices of this parameter were very similar.

5 Results

One aspect in the comparison of the different algorithms is to look at their accuracy on the spatially separate test set. That is, in the following we will concentrate on what portion of the pixels in the test data is correctly classified, allowing for first conclusions. The initial accuracies (that is, when no test sample points are rejected) of the different classifiers are shown in Tab. 1, and here the SVM based architectures clearly are to be preferred. As mentioned above, the goal of our experiments was to assess the behaviour of the classifiers once they were allowed to reject an (increasing) portion of the test samples. This is plotted in Fig. 1 for all the classifiers. As the rejection rate is not a direct parameter of the algorithms, we simply raised the rejection threshold from 0 to 1 in steps of 0.01, and noted the respective rates. Looking at the upper graph, the SVM

[4] With normalise we mean to make the values of each label to sum up to one.

Table 1. Accuracy of the different supervised learning algorithms studied, when no samples were allowed to be rejected.

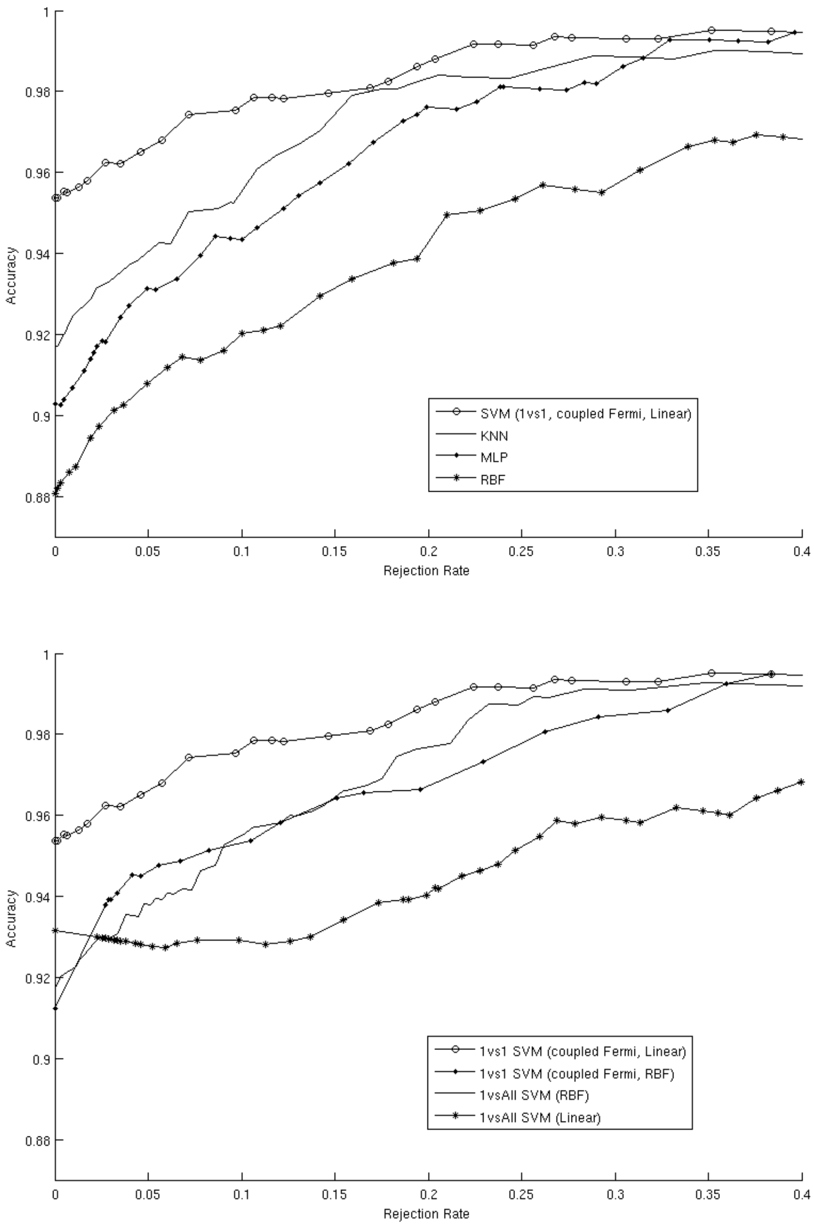
Classifier	Initial Accuracy
RBF	0.88
MLP	0.90
KNN	0.92
SVM, 1vsAll, RBF	0.92
SVM, 1vsAll, Linear	0.93
SMV, 1vs1, coupled Fermis, RBF	0.91
SMV, 1vs1, coupled Fermis, Linear	0.95
SVM, 1vs1, coupled votes, RBF	0.93
SVM, 1vs1, coupled votes, Linear	0.95

steadily provides the highest accuracy on the test set. The RBF has the worst initial performance, and also the slowest increase of accuracy. Interestingly, the relatively simple KNN algorithm has a good initial accuracy, and raises very fast, so that at a level of rejection of 15% it reaches the performance of the champion SVM. Hence, being much easier to implement, the KNN solution might be to preferred in some applications. Not shown in the graph, the first algorithm to reach a perfect accuracy of 1 is the MLP, at the price of rejecting 43% of the samples. The accuracies of the different SVM architectures and kernels, also plotted against a rising rejection rate, can be found in the lower part of Fig. 1. The results do allow only a few conclusions: the 1vs1 architecture with coupled Fermis is the clear winner, always yielding the highest accuracy. And the approaches based on RBFs exhibit a similar behaviour for both the 1vs1 and 1vsAll architecture.

It must be stressed again that the performance of SVMs is highly dependent on selecting the appropriate kernel and its parameters for each data set. Without empirical proof, based solely on experience gained in a range of applications, it seems to us that when properly tuned, the 1vs1 architecture (with coupled Fermis) will always provide a higher accuracy then the 1vsAll scheme.

In 1vs1 SVMs, the single most common method is to use only the votes of the individual classifiers for coupling, but not the transformed distances. And as our results, and experiments in [23] show, the performance in terms of accuracy is nearly identical to the one achieved using coupled sigmoids. Yet, in the sample-rejection scenario, we observed a strange behaviour in the machines with coupled votes: the rejection rate would go from nearly 0 to around 80% at a certain threshold, which makes this architecture useless in this context. Examining the answers to the 628 test samples closer, we found that there are only 79 different variants of the vote matrix, leading to 77 different coupled answers, with only 13 different maxima. As ten of those maxima even lie in the narrow range of [0.74 0.76], it is immediately clear that a smooth scaling of the tradeoff accuracy versus rejection rate is not possible. Whereas coupled transformed distances lead to completely individual answers, forcing the SVMs to answer in hard votes in the experiment limited the variance in the answers to a tiny fraction of the

Fig. 1. Accuracy of various supervised classifiers, at certain rates of rejection of test samples. The upper plot shows performances across several different architectures (including the best SVM), the lower one across different SVM approaches.

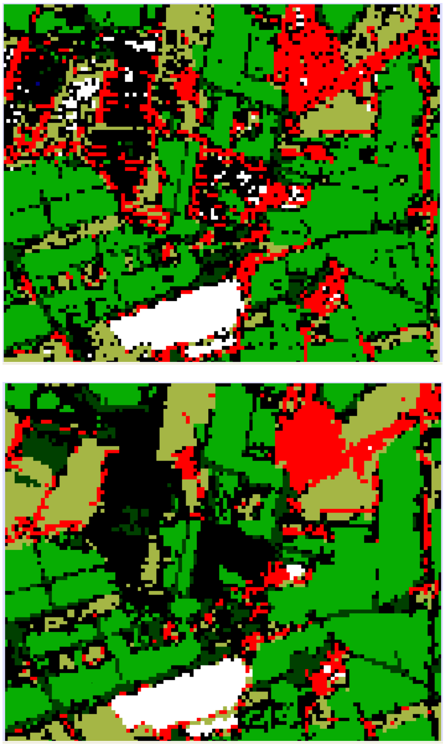


possible answer space, which contains 2^{21} different elements. Hence, 1vs1 SVMs based on coupled votes should not be used for this purpose.

A visual analysis of the answers on the whole dataset essentially confirms the effectiveness of the 1vs1 SVM approach. The distribution of the rejected pixels is very sparse and indented for the RBF, MLP, KNN, and 1vsAll SVMs. Whereas with the 1vs1 SVMs, both using a linear and a RBF kernel, the rejected pixels are gathered in more compact regions. Using the RBF kernel, however, the 1vs1 SVMs, like the non-SVM classifiers, exhibit a clear overestimation of the urban green class.

To allow the reader a visual comparison, the land cover mappings produced by a 1vs1 SVM and a 1vsAll SVM are depicted in Fig. 2. The differences in the distribution of rejected (black) pixels mentioned above is clearly visible. Also, in the 1vsAll case, one can observe an overestimation of the greenhouses (white pixels on the map) to the detriment of not vegetated lands (brown pixels on the map), an error which we found, despite good overall accuracies on the test set, common to all the classifiers except for the 1vs1 SVM architecture.

Fig. 2. The figure shows a portion of the final map provided by the 1vsAll SVM (RBF kernel, upper map), and the 1vs1 SVM (Linear kernel, lower map). Black pixels are spread in a compact way with the 1vs1 architecture, enabling an easier interpretation in successive analysis. Note that this figure appears in colour in the electronic version of this article.



6 Conclusions

In this work, we compared the capabilites of different supervised classification algorithms to produce meaningful land-cover mappings. Of special interest was the performance of the as yet little explored Support Vector Machine approach, as well as the behaviour of all classifiers when being presented with mixed pixels that contain several land-cover classes on the ground.

The results yielded insights into the comparative performance of different SVM architectures, leading us to the conclusion that the 1vs1 SVMs are to be preferred for best classification accuracies, but are also sensitive to the choice of kernel. The performance of SVM approaches using a RBF kernel is rather stable.

Overall, the best results are provided by the 1vs SVM with a linear kernel: it yields the highest accuracy on the test set, as well as a distribution of rejected pixels on the output map that is much more clear and interpretable than with all other approaches. Looking at the rejection behaviour, it is clear that the fuzzy answers in the 1vs1 architecture have to be produced using a sigmoid transfer function and Bradley-Terry coupling, instead of the usual votes.

For the non-SVM approaches, the relatively simple K-Nearest-Neighbour has the highest classification accuracy.

Our future research in this area will entail incorporating spectral and textural information from higher-resolution images to further improve the meaningfulness of mixed SVM answers, allowing for a richer presentation of the land-cover map.

References

1. Gidudu, A., Hulley, G., Marwala, T.: Image classification using svms: One-against-one vs one-against-all. In: Proccedings of the 28th Asian Conference on Remote Sensing. (2007)
2. Bruzzone, L., Marconcini, M.: An Advanced Semisupervised SVM Classifier for the Analysis of Hyperspectral Remote Sensing Data. In: Proceedings of the 12th SPIE International Symposium on Remote Sensing / Image and Signal Processing for Remote Sensing XII. Volume 6365., International Society for Optical Engineering (2006) 63650Y–1–6350Y–12
3. Xua, M., Watanachaturaporna, P., Varshneya, P.K., Arorab, M.K.: Decision tree regression for soft classification of remote sensing data. Remote Sensing of Environment **97**(13) (2005) 322–336
4. Giacco, F., Pugliese, L., Scarpetta, S., Marinaro, M.: Application of Neural Unsupervised Methods to Environmental Factor Analysis of Multi-spectral Images with spatial information. In Sablatnig, R., Scherzer, O., eds.: Proceedings of Signal Processing, Pattern Recognition, and Applications, SPPRA, ACTA press, Zürich (2008)
5. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural Features for Image Classification. IEEE Transactions on Systems, Man, and Cybernetics **SMC-3**(6) (1973) 610–621
6. Mather, P.M.: Computer Processing of Remotely-Sensed Images. Wiley (1999)
7. Schürmann, J.: Pattern Classification, a unified view of statistical and neural approaches. John Wiley & Sons (1996)

8. Thiel, C.: Using Dempster-Shafer Theory in MCF Systems to Reject Samples. In Oza, N.C., Polikar, R., Kittler, J., Roli, F., eds.: Proceedings of the 6th International Workshop on Multiple Classifier Systems, MCS 2005. Volume 3541 of Springer LNCS. (2005) 118–127
9. Webb, A.R.: Statistical Pattern Recognition. second edn. John Wiley & Sons (2002)
10. Hornik, K., Stinchcombe, M.B., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks **2**(5) (1989) 359–366
11. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In Rumelhart, D., McClelland, J., eds.: Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations. Volume 1. MIT Press (1986) 318–362
12. Therrien, C.W.: Decision estimation and classification: an introduction to pattern recognition and related topics. John Wiley & Sons (1989)
13. Powell, M.J.D.: Radial basis functions for multivariate interpolation: A review. In Mason, J.C., Cox, M.G., eds.: Algorithms for Approximation. Clarendon Press, Oxford (1987) 143–168
14. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press (1967) 281–298
15. Breiman, L., Meisel, W., Purcell, E.: Variable Kernel Estimates of Multivariate Densities. Technometrics **19**(2) (May 1977) 135–144
16. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer (1995)
17. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2002)
18. Angulo, C., Ruiz, F.J., González, L., Ortega, J.A.: Multi-Classification by using Tri-Class SVM. Neural Processing Letters **23** (February 2006) 89–101
19. Kahsay, L., Schwenker, F., Palm, G.: Comparison of multiclass SVM decomposition schemes for visual object recognition. In: DAGM 2005. Volume 3663 of LNCS., Springer (2005) 334–341
20. Thiel, C., Scherer, S., Schwenker, F.: Fuzzy-Input Fuzzy-Output One-Against-All Support Vector Machines. Proceedings of the 11th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems KES 2007 (2007)
21. Platt, J.C.: Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In: Advances in Large Margin Classifiers, NIPS 1998, MIT Press (1999) 61–74
22. Bradley, A.R., Terry, M.E.: Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. Biometrika **39**(3/4) (1952) 324–345
23. Hastie, T., Tibshirani, R.: Classification by Pairwise Coupling. The Annals of Statistics **26**(2) (1998) 451–471