One-Class Support-Vector Machines for the Classification of Bioacoustic Time Series

Andreas Sachs, Christian Thiel and Friedhelm Schwenker
Department of Neural Information Processing
University of Ulm
89069 Ulm
Germany
Email: *firstname.lastname*@uni-ulm.de

Abstract

*Support Vector Machines (*SVM*) have become a widespread method in machine learning applications. In this paper we studied the* one-class SVM *method, whose goal is to describe the data from a single class by a set of support vectors. One-class SVMs can be used to construct multiple classifier systems (*MCS*) utilising the individual one-class data descriptions together with strategies to combine the classifier decisions and to resolve classifier conflict situations when a new unseen pattern has to be classified. Here the one-class SVM approach has been applied to a classification problem appearing in bioacoustic monitoring, where the species of a singing insect has to be determined.*

1 Introduction

Support vector machines have become a popular method in pattern classification and were originally developed for the discrimination of two-class problems [11]. One variant, the so called one-class classification method, which was very well investigated by D. Tax [10] and others (see [7]), uses only the data of a single class to construct a model of the data. With this model it is possible to make a decision for a new, hitherto unseen object about its membership to the so called target class. This method is similar to unsupervised learning algorithms like *Gaussian mixture models (GMM)* [4] or *k-means* clustering [6].

In Section 2 the one-class support-vector variant for learning of multi-class problems is introduced and in Section 3 the bioacoustic monitoring problem is described, including a brief introduction into the feature extraction procedure applied to the cricket sounds. Subsequently we will evaluate and discuss the numerical experiments.

2 One-Class Support-Vector-Learning for Multi Class Problems

SVM learning from examples of a single class is also called support-vector-data-description (SVDD) [10]. This method is closely related to clustering or density estimation. SVDD has the goal to learn a covering hypersphere (with centre $\mathbf{a}$ and radius R) of the training data (Figure 1).

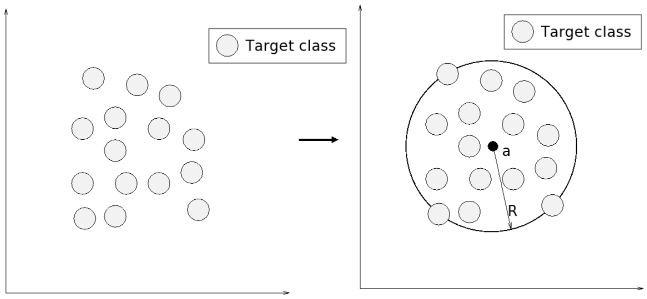

Figure 1. Data of a single class (left panel) is covered by the circle with centre $\mathbf{a}$ and radius R (right panel). This circle defines a boundary separating the dataset from the rest of the input space.

Assuming the hypersphere covers the whole set of training data points $\mathbf{x}_i \in X$ the empirical error is equal to 0. In close relation to the binary SVM approach [11] the structural error is defined through

$$E_{struct}(R, \mathbf{a}) \;=\; R^2 \tag{1}$$

Thus the primal optimisation problem is defined by

$$\min R^2 \tag{2}$$

with one constraint for each training pattern $\mathbf{x}_i$:

$$||\mathbf{x}_i - \mathbf{a}||^2 \leq R^2 \quad i = 1, \dots, N \tag{3}$$

Using the Karush-Kuhn-Tucker theory to solve problem (2) with respect to the constraints (3) the Lagrangian is given by

$$L(R, \mathbf{a}, \alpha) \;=\; R^2 - \sum_{i=1}^{N} \alpha_i (R^2 - ||\mathbf{x}_i - \mathbf{a}||^2)\,, \tag{4}$$

with $\alpha_i \geq 0$ for $i = 1, \dots, N$. This can be rewritten to

$$L(R, \mathbf{a}, \alpha) \;=\; R^2 - \sum_{i=1}^{N} \alpha_i (R^2 - (\mathbf{x}_i\mathbf{x}_i - 2\mathbf{a}\mathbf{x}_i + \mathbf{a}\mathbf{a})) \tag{5}$$

To create the appropriate dual problem the partial derivatives of (5) have to be set to 0:

$$\frac{\partial L(R, \mathbf{a}, \alpha)}{\partial R} = 2R - 2R \sum_{i=1}^{N} \alpha_i \overset{!}{=} 0$$

and

$$\frac{\partial L(R, \mathbf{a}, \alpha)}{\partial a} = -2 \sum_{i=1}^{N} \alpha_i (\mathbf{x}_i - \mathbf{a}) \overset{!}{=} 0$$

This leads to

$$\sum_{i=1}^{N} \alpha_i = 1 \tag{6}$$

and

$$\mathbf{a} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i \tag{7}$$

Inserting conditions (6) and (7) into the Lagrangian (5), the dual optimisation problem becomes

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i \mathbf{x}_i \mathbf{x}_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \mathbf{x}_i \mathbf{x}_j \tag{8}$$

with the constraints

$$\sum_{i=1}^{N} \alpha_i = 1 \tag{9}$$

and

$$\alpha_i \geq 0 \qquad i = 1, \dots, N. \tag{10}$$

Those patterns $\mathbf{x}_j$ located at the boundary of the hypersphere are called the *support vectors* of the data set (see Figure 1) and are characterised by Lagrange multipliers $\alpha_j > 0$. For the decision function f determining the class membership, the radius R and centre $\mathbf{a}$ have to be calculated. While $\mathbf{a}$ is given directly by (7), radius R can be calculated by choosing arbitrarily a support vector $\hat{\mathbf{x}}_k$ (with Lagrange multipliers $\alpha_k > 0$). Taking into account the Karush-Kuhn-Tucker conditions

$$\alpha_i(||\mathbf{x}_i - \mathbf{a}||_2^2 - R^2) = 0, i = 1, \dots, N$$

it follows that

$$R^2 \;=\; \hat{\mathbf{x}}_k \cdot \hat{\mathbf{x}}_k - 2\mathbf{a} \cdot \hat{\mathbf{x}}_k + \mathbf{a} \cdot \mathbf{a}$$

and then, using equation (7):

$$R^2 = \hat{\mathbf{x}}_k \cdot \hat{\mathbf{x}}_k - 2 \sum_{i=1}^{N} \alpha_i \mathbf{x}_i \cdot \hat{\mathbf{x}}_k + \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{11}$$

The decision function f for a new sample z, determining if z is within the hypersphere, can be written as

$$f(\mathbf{z}, \mathbf{a}, R) = I(||\mathbf{z} - \mathbf{a}||_2^2 \leq R^2) \tag{12}$$

with

$$I(A) = \begin{cases} 1, & \text{if A true} \\ -1, & \text{otherwise} \end{cases}$$

An important feature of support vector machines is the usage of kernel functions K. A Mercer kernel function (see [11]) transforms (implicitly) the data points of the input space X to a high dimensional Hilbert space H (sometimes called feature space) and the calculation of the dot product in H can be written as:

$$K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_H \quad \mathbf{x}), \mathbf{y}) \in X$$

The transformation $\phi : X \rightarrow H$ need not be computed explicitly, and $\langle \cdot, \cdot \rangle_H$ is the dot product defined on H. Applying this to the decision function (12), the values of $\phi(\mathbf{a})$ and $\phi(R)$ are computed as follows:

The squared Euclidian distance $||\phi(\mathbf{x}_i) - \phi(\mathbf{a})||_2^2$ is equal to

$$\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_i) - 2\phi(\mathbf{a}) \cdot \phi(\mathbf{x}_i) + \phi(\mathbf{a}) \cdot \phi(\mathbf{a}) =$$
$$K(\mathbf{x}_i, \mathbf{x}_i) - 2 \sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i, \mathbf{x}_i) + \sum_{i,j=1}^{N} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{13}$$

and the radius R_ϕ of the transformed data $\phi(\mathbf{x}_i)$ is equal to

$$K(\mathbf{x}_k, \mathbf{x}_k) - 2 \sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i, \mathbf{x}_k) + \sum_{i,j=1}^{N} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{14}$$

As already mentioned before, $\mathbf{x}_k$ is an arbitrarily chosen support vector with $\alpha_k > 0$.

As kernel function the Gaussian kernel will be used

$$K(\mathbf{x}, \mathbf{z}) = e^{-\frac{||\mathbf{x}-\mathbf{z}||^2}{2\sigma^2}}, \qquad \sigma^2 > 0 \tag{15}$$

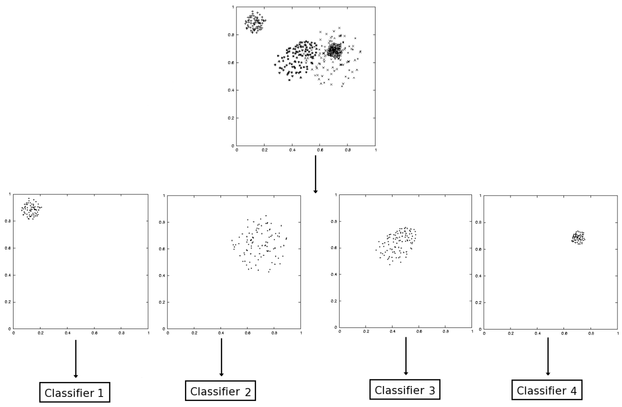

Figure 2. Learning a single one-class classifier for every single class of a multiclass problem.

The described one-class methodology can be applied to multiclass problems in the following way. It is assumed that data points from k different classes are given (in Figure 2 data from four classes). Let $\Omega = \{1, \dots, k\}$ denote the set of classes. For each single class $t \in \Omega$ a one-class classifier f_t is constructed, more precisely k centres $\mathbf{a}_t$ and corresponding radii R_t are calculated. To determine the class membership of a new unseen pattern $\mathbf{z}$ all decision functions are evaluated in parallel leading to class memberships $f_t(\mathbf{z})$, $t \in \Omega$:

$$f_t(\mathbf{z}, \mathbf{a}_t, R_t) = I(||\mathbf{z} - \mathbf{a_t}||_2^2 \le R_t^2). \qquad (16)$$

In the classification phase of an unseen input pattern $\mathbf{z}$ three possible resulting situations have to be distinguished:

1. If exactly one classifier indicates its responsibility for the new pattern, the resulting class label is the class membership of this classifier.
2. If more than one classifiers raises a claim for this pattern, a so called *conflict situation* has arrived.
3. On the other hand if no classifier matches, we characterise this result as an *outlier situation*.

The problem with such unclear (**2** and **3**) classification assignments can be solved in two ways: Rejecting the classification of the new pattern or finding the corresponding class label with an outlier-/conflict processing strategy. In the following we will introduce two different strategies to solve such problems.

2.1 Nearest-Centre Strategy

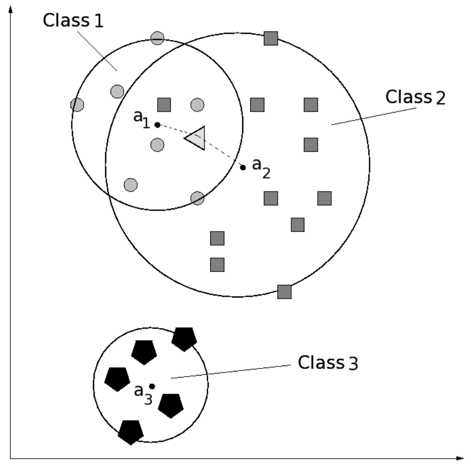

Figure 3. Nearest-Centre Strategy: Pattern (triangle) is classified to the SVM whose centre is closest. Here a conflict situation is depicted.

Basically, a new pattern $\mathbf{z}$ will be assigned to the classifier, whose centre $\mathbf{a_i}$ has the shortest distance to $\mathbf{z}$. A conflict situation appears if the candidate set $\mathcal{C}$ given by $I(z) := \{t \;:\; f_t(z) = 1\} \subset \Omega$ is of cardinality > 1. Then the decision function is

$$f(\mathbf{z}) = \arg\min_{l \in \mathcal{C}} ||\mathbf{z} - \mathbf{a}_l||. \qquad (17)$$

This situation is illustrated in Figure 3, only the centres of the classifiers from the candidate set $I(\mathbf{z})$ are evaluated. Because the candidate set $I(\mathbf{z})$ is empty for an outlier z, in this case the $\arg\min$ is taken over $\mathcal{C} = \Omega$.

2.2 Nearest-Support-Vector Strategy

The class label of a new pattern $\mathbf{z}$ is estimated by comparing the distance to the support vectors of the involved classifiers (Figure 4 shows a conflict situation). The classifier with the nearest support vector determines the class membership of $\mathbf{z}$. Assuming that $SV(t) := \{\hat{\mathbf{x}}_1^t, \dots, \hat{\mathbf{x}}_{i_t}^t\}$ are the support vectors, and i_t the number of support vectors of classifier t.

The decision function is then defined through

$$f(\mathbf{z}) = \arg\min_{l \in \mathcal{C}} ||\hat{\mathbf{x}}_i^l - \mathbf{z}|| \qquad (18)$$

where the minimum is taken over all support vectors of classes from the candidate set. Again, in the conflict situation $\mathcal{C}$ is equal to $I(\mathbf{z})$, and in the outlier situation the candidate set is set to Ω.

3 Bioacoustics of Crickets

The one-class SVM approach is applied to the automatic classification of insect songs. The objective is to classify

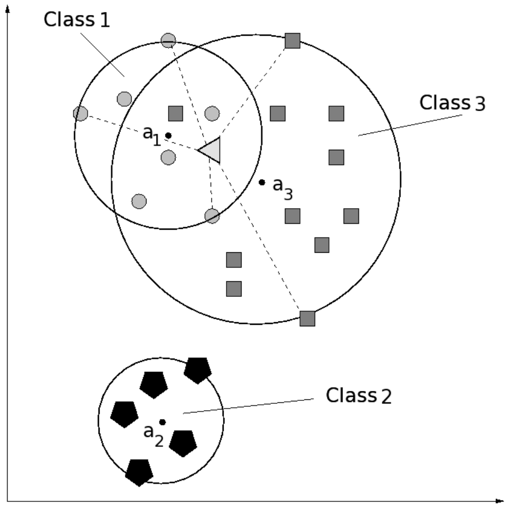

Figure 4. Nearest-Support-Vector Strategy: Pattern (triangle) is classifier to the SVM that contains the support vector closest to the new pattern. Here a conflict situation is depicted.

extracted feature vectors from sound recordings of crickets. In Figure 5 a part of a cricket song is depicted. Crickets are members of the family of Gryllidae. They produce characteristic sound patterns by stridulation their legs and wings [8] which can be perceived by humans.

3.1 Feature Extraction

Signal preprocessing methods and feature extraction techniques applied to the cricket sounds were introduced in [2]. It is well known (see[2, 1, 9] and references therein) the most discriminative features are derived from the temporal structure of the so called pulses (see Fig.5). These features are the mean length (feature L), the average frequency (feature F) the pulse distances (feature T) of consecutive pulses. From these single features combined features (by vector concatenation leading to features LF, LF, FT and LFT) have been used as inputs to the classifiers.

In the following we briefly describe the feature processing. It is assumed that n pulses have been detected in the insect song given through the onsets $\lambda = (\lambda_1, \ldots, \lambda_n)$ and offsets $\mu = (\mu_1, \ldots, \mu_n)$.

- Pulse length (L)
  The mean pulse length L_i for a set of d consecutive pulses is determined from the onsets λ and offsets μ by

$$L_i = \frac{1}{d} \sum_{j=1}^{i+d-1} (\mu_j - \lambda_j), \quad i = 1, \ldots, n-d$$

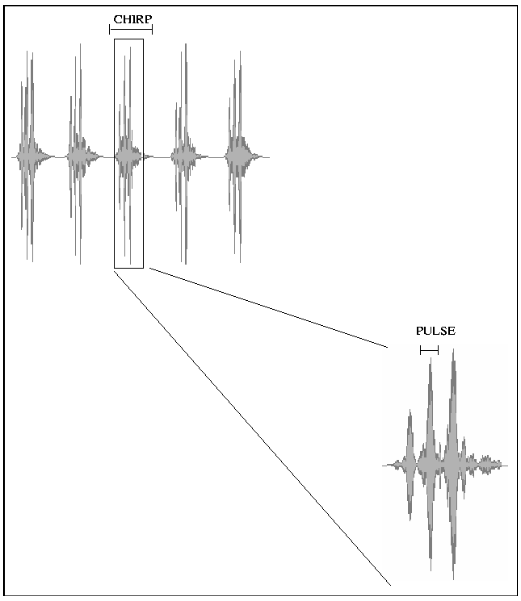

Figure 5. A sound pattern of a cricket which belongs to the genus *Gryllus bimacalatuts* **consists of a sequence of chirps which each contain a series of pulses. The temporal structure of the pulses are the most discriminative features to classify the species of an singing insect.**

- Frequency contour (F)
  Let f be the average frequency contour of the k-th pulse. Then the average frequency contour over d pulses is

$$F_i = \frac{1}{d} \sum_{j=1}^{i+d-1} f_k, \quad i = 1, \ldots, n-d$$

- Temporal structure of pulses (T) by d-gram coding
  Let the distance between pulse i and $i+1$ is equal to $\delta_i = \lambda_{i+1} - \lambda_i$. The d-gram vector is then

$$\Delta_i = (\delta_i, \ldots \delta_{i+d}) \quad i = 1, \ldots, n-d$$

For each type of feature, the feature vector is created by sliding a window of the width d, yielding to streams of feature vectors $L = (L_1, \ldots, L_{n-d})$, $F = (F_1, \ldots, F_{n-d})$, and $\Delta = (\Delta_1, \ldots, \Delta_{n-d})$. Concatenation of types of features results in a new feature vector, e.g. the LFT vector is simply $((L_1, F_1, \Delta_1), \ldots, (L_{n-d}, F_{n-d}, \Delta_{n-d}))$.

3.2 Method

The multiple classifier system was trained with recordings of 28 cricket species with 3-12 individuals per species.

The complete dataset consists of 195 individuals. The temporal context of analysis windows was not taken into account except the averaging and d-gram coding of consecutive pulses as explained in Section 3.1. The whole dataset was reduced by choosing 70 feature vectors randomly per individual. Then, the classifiers were trained as explained in Section 2. After training, the classifiers were used to determine the class membership of unknown individuals. To that end, the feature vectors extracted from the unseen recording were tested by evaluating the k decision functions. The result was a vector of classifier decisions. A majority voting scheme was then applied to determine the final class membership.

For statistical evaluation a 10-fold cross validation for the sequences is used. It is ensured that an individual is used either for training or for testing. The validation process was repeated five times and the final results were calculated by the mean.

The one-class SVMs were evaluated by using the Gaussian kernel. Varying the variance of the Gaussian σ^2 from 0.005 to 0.08 produced different rates for conflict- and outlier situations (Figure 6)

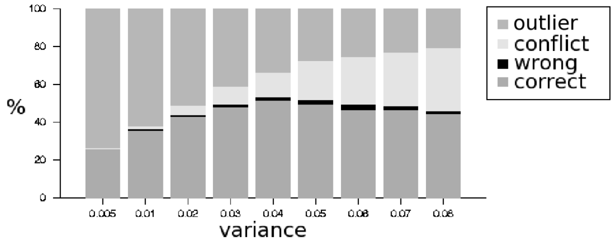

Figure 6. Classification results achieved by the one-class svm(in %) for unique decisions (correct/wrong), outlier and conflict situation. LFT features extracted form the cricket songs were used (Kernel function: Gaussian kernel with different variances σ^2).

Taking a look at Figure 6, we see that the fraction of unique classifier scores (exactly one classifier indicates his responsibility for the class membership) varies depending on the variance σ^2 of the Gaussian kernel and has its maximum at $\sigma^2 = 0.04$. While the fraction of outlier situations is quite high with low variances (σ^2 = 0.005) and lower with higher variance (σ^2 = 0.08), the fraction of conflict situations develops contrarily to that. It was decided to fix the variance at $\sigma^2 = 0.020$ in the classification experiments, resulting in a percentage of unique classifier response of approximately 43% and a percentage of conflict- and outlier situations of around 57%.

Overview					
feature	correct	wrong	conflict	outlier	sv
L	2.77	0.72	82.87	13.64	24.86
F	5.13	0.82	92.51	1.54	2.74
T	24.82	4.92	51.69	18.56	22.07
LF	34.46	3.59	37.44	24.51	20.67
LT	30.56	3.38	18.26	47.79	36.19
FT	**44.31**	1.03	18.67	36.00	21.80
LFT	41.95	2.05	4.00	52.00	33.08

Table 1. Results of the one-class SVM for the four different classifier situations correct/wrong/conflict/outlier (in %) measured for the 7 different feature combination. In the rightmost column (sv) the fraction (in %) of data points that were identified as support vectors is given.

Conflict- and Outlier processing (error rates)		
feature	*nearest-centre*	*nearest-support-vector*
L	67.38	65.95
F	74.49	77.88
T	32.41	31.59
LF	34.26	27.08
LT	24.92	22.46
FT	15.59	13.03
LFT	16.62	**8.31**

Table 2. Error rates (%) for all possible feature combinations of the one-class svm utilising *nearest-centre* **and** *nearest-support-vector* **conflict- and outlier processing strategies.**

4 Results

Now we will take a look at the classifier scores for the different features as introducted in section 3.1. In cases where not exactly one classifier answered, the distinction between conflict- and outlier situation was made (irrespective of the correctness of the classification result). The results are displayed in Table 1 with an additional column for the mean percentage of training samples that were used as support vectors ($\alpha_i > 0$, see Equation 12).

To achieve an unique decision the *nearest-centre* and the *nearest-support-vector* methods introduced in section 2 have been applied. The results are shown in Table 2.

To compare the classification results of our multi classifier system using one-class support vector machines to other commonly used classification methods, the training

K-nearest-neighbour (error rates)			
		k	
feature	1	3	5
L	48.72	50.36	50.97
F	65.54	68.51	68.31
T	22.60	22.07	22.95
LF	21.89	22.05	21.89
LT	18.53	18.97	19.18
FT	9.74	9.44	10.46
LFT	**9.59**	9.85	9.95

Table 3. Error rates (%) for the K-nearest-neighbour classifier for $K = 1, 3, 5$

.

LVQ network (error rates						
	Number of cluster centres per class					
feature	1	3	5	10	15	20
L	94.77	89.64	85.64	73.23	63.59	60.41
F	89.33	86.46	82.67	77.74	72.62	72.51
T	95.59	88.72	72.62	43.38	36.82	35.17
LF	62.26	28.51	25.54	24.41	25.44	26.97
LT	95.18	68.31	42.46	26.87	26.36	26.36
FT	85.23	35.90	21.33	16.41	17.13	20.21
LFT	84.10	25.23	15.69	**12.41**	17.54	18.46

Table 4. Error rates (%) for the learning vector quantisation neural network for $K = 1, 3, 5, 10, 15, 20$ prototypes per class. Classification of new unseen patterns is through the 1-nearest neighbour rule applied to the prototypes.

and testing data is evaluated with the k-nearest-neighbour classifier [3] and the learning vector quantisation network [5]. The results are shown in Table 3 and Table 4.

5 Discussion

The rate of unique classifier responses varies from 3.49% (feature L) to 45.34% (feature FT). This means that the hyperspheres of the one-class SVMs overlap (conflict) or do not cover the whole input space (outlier) in a significant number of cases. It is worth noting that the feature that performed best at this stage, FT, did by far not need the highest number of support vectors. Classifiers working on combined features have a higher accuracy, because the input variables have a higher dimension and hence can provide more discriminative information. In all feature combinations more than half of the samples were in a conflict- or outlier situation, solving those correctly was crucial. Both strategies used resulted in a very significant boost of the classification performance (bringing the error from 44% down to 8%), with the nearest-support-vector strategy always performing best, yielding an error rate of only 8.3% on the LFT feature. The reference classifiers did not do so good, the LVQ network achieved only an error rate of 12.4%, while the K-nearest-neighbour came close to our approach with 9.6%. Interestingly, it was the 1-nearest-neighbour method that did so well, which does not take account any but the nearest training sample, similarly to the nearest-support-vector strategy. Most likely, that approach worked better because the important (support) vectors had been preselected.

6 Conclusion

Our one-class-support vector machine performed best in the experiment, with the K-nearest-neighbour method being the closest competitor. The strategies to resolve conflict- and outlier situations turned out to be indispensable for the success of the architecture. A method to still improve the performance of the system could be to let it work separately on the different features, and then combine the decisions with an appropriate fusion rule, majority vote in the simplest case.

Acknowledgement

The authors would like to thank Dr. Klaus Riede (Zoological Research-Museum, Bonn, Germany) for providing the data. This work was partially supported by DFG (German Research Society) grant SCHW 623/3-2.

References

[1] C. Dietrich, G. Palm, K. Riede, and F. Schwenker. Classification of bioacoustic time series based on the combination of global and local decisions. *Pattern Recognition*, 37(12):2293–2305, 2004.

[2] C. R. Dietrich. *Temporal Sensorfusion for the Classification of Bioacoustic Time Series*. PhD thesis, University of Ulm, 2003.

[3] K. Fukunaga. *Introduction to statistical pattern recognition*. Academic Press, New York, 2nd edition, 1990.

[4] T. Hastie and R. Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society series B*, 58:158–176, 1996.

[5] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, and K. T. ko la. Lvq-pak: The learning vector quantization program package. Technical Report A30, Helsinky University of Technology, 1996.

[6] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *5th berkeley symposium on mathematical statistics and probability*, pages 281–298, 1967.

[7] L. Manevitz and M. Yousef. One-class svms for document classification. *J. Mach. Learn. Res.*, 2:139–154, 2002.

[8] K. Riede. Acoustic monitoring of orthoptera and its potential for conversation. *Journal of Insect Conservation*, 2:217–223, 1998.

[9] F. Schwenker, C. Dietrich, H. Kestler, K. Riede, and G. Palm. Radial basis function neural networks and temporal fusion for the classification of bioacoustic time series. *Neurocomputing*, 51:265–275, 2003.

[10] D. M. J. Tax. *One-class classification; Concept-learning in the absence of counter-examples*. PhD thesis, Delft University of Technology, June 2001.

[11] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.