Radial basis function neural networks and temporal fusion for the classification of bioacoustic time series

Friedhelm Schwenker\textsuperscript{a,}\textsuperscript{*}, Christian Dietrich\textsuperscript{a}, Hans A. Kestler\textsuperscript{a}, Klaus Riede\textsuperscript{b}, Günther Palm\textsuperscript{a}

\textsuperscript{a}Department of Neural Information Processing, University of Ulm, D-89069 Ulm, Germany
\textsuperscript{b}Zoological Institute of Scientific Research and Museum, Alexander Koenig, D-53113 Bonn, Germany

Received 9 July 2001; accepted 8 May 2002

Abstract

In this paper, we discuss radial basis function (RBF) classifiers and classifier fusion architectures to categorize bioacoustic time series. The proposed methods are evaluated on a data base of cricket songs of 35 different species recorded in Thailand and Ecuador. Local characteristic features are extracted from the recordings and then categorized by RBF classifiers. These local decisions are then combined through temporal fusion. Three different fusion techniques (Averaging, Percentiles and Voting) are considered. We present and discuss the classification results for the proposed classifier architectures on this data set of 35 species.

\textcopyright\ 2002 Elsevier Science B.V. All rights reserved.

Keywords: Classifier fusion; Temporal fusion; Bioacoustics; Radial basis function networks

1. Introduction

The classification of sounds of species is a fundamental challenge to study animal vocalizations [4,20]. The automatic recognition of animal vocalizations would be rather valuable for biological research and environmental monitoring applications. Many of these studies are based on manual inspection and labeling of sound spectra, which relies on agreement between human experts. Recently, pattern recognition approaches

\textsuperscript{*} Corresponding author. Tel.: +49-731-502-4159; fax: +49-731-502-4156.

E-mail addresses: schwenker@neuro.informatik.uni-ulm.de (F. Schwenker), dietrich@neuro.informatik.uni-ulm.de (C. Dietrich), kestler@neuro.informatik.uni-ulm.de (H.A. Kestler), k.riede.zfmk@uni-bonn.de (K. Riede), palm@neuro.informatik.uni-ulm.de (G. Palm).

0925-2312/03/$-see front matter \textcopyright\ 2002 Elsevier Science B.V. All rights reserved.

PII: S0925-2312(02)00621-5
including machine learning techniques, statistical hidden Markov models or artificial neural networks have been used for automatic classification of animal vocalizations [6,11,14].

Grigg et al. [6] implemented a software system based on Quinlan’s C 4.5 decision tree algorithms [17] in order to classify the vocalizations of 22 species of frogs occurring in northern Australia. Kogan and Margoliash discussed in [11] dynamic time warping and hidden Markov models [18] for the automated classification of bird songs (four zebra finches (Taeniopygia guttata) and four indigo buntings (Passerina cyanea)). In Murray et al. [14], Kohonen’s self-organizing neural network [12] has been used to analyse vocalizations of false killer whales (Pseudorca crassidens). In Dietrich et al. [2,3] k-nearest-neighbor classifiers are utilized for the categorization of cricket songs from 28 species.

Bioacoustic song analysis already forms an integral part within species descriptions of “new” Orthoptera [9]. Cricket songs recorded from a certain tropical region in Ecuador could be classified qualitatively within a parameter space of carrier frequency and pulse intervals [20]. By catching voucher specimens, it could be shown that each cluster of feature vectors is indeed produced by a distinct species [15]. This data set has been also used for the numerical evaluation presented in this paper. At present, a specimen-based multimedia database is set up, bringing together data from different sound archives within the “virtual phonothek” (http://www.dorsa.de). Besides the mentioned features used by traditional, descriptive bioacoustics, other signal parameters can be analyzed in order to classify animal sounds. Examples for such parameters are the periodogram, the energy contour and the frequency contour.

The automated classification of bioacoustic time series is the topic of this paper. Radial basis function (RBF) networks are used for the classification of characteristic local features extracted from the time series. These local decisions are combined through classifier fusion techniques in order to categorize the whole time series. Sounds of crickets (Grylloidea) of 35 different species from Thailand and Ecuador are analyzed and classified. These songs were recorded in the field or at the lab under rather different conditions, thus the automatic classification is not an easy task.

The first step of the whole classification task is the feature extraction [3]. For this, the locations of pulses (see Fig. 4) are determined and different features from the time and frequency domain are extracted automatically from the time series and then used for the categorization of the sound patterns. Two types of feature extraction can be distinguished:

1. **Global features**: These features are based on the information of the whole time series, e.g. the mean frequency, mean energy, etc.
2. **Local features**: These are derived within local time windows $W^t$ of the whole time series. In this type of feature extraction, a set of features is calculated within the window $W^t$. The window is then moved by a time step $\Delta t$ into $W^{t+\Delta t}$ and the next set of features is calculated.

We focus on the automated classification of bioacoustic time series based on local features, i.e. on sequences of locally derived feature vectors. These feature vectors serve
as inputs for a radial basis function (RBF) network. For each local feature vector a soft decision is calculated through the RBF classifier. Then all these local decisions are combined to an overall decision. This type of decision fusion is called temporal fusion. Temporal fusion is the combination of a sequence of decisions calculated in different parts of the time series into a single decision. A large number of combining schemes for temporal classifications exists: combining local class decisions, voting the soft classifier outputs, and combining the real valued outputs of each class [26]. The reader may consult the article by Kittler et al. [10] for an overview on classifier combination. We present RBF classification results of all three types of fusion schemes.

2. Classification fusion of local features

In this section, we present the DCT-Architecture (Data fusion, Classification, Temporal fusion) which is applied to classify local features, extracted from local time windows. Here, a window $W_t$ covering a small part of the time series is moved over the whole time series. For each window $W_t$, $t = 1, \ldots, T$ a set of $p$ features $F_t^i \in \mathbb{R}^{d_i}$, $i = 1, \ldots, p$ and $d_i \in \mathbb{N}$, is extracted from the time series. The situation is illustrated in Fig. 1.

The classification of the whole time series is determined through the classification of the concatenated feature vectors (which may be called data fusion) and temporal fusion over the whole time series (see Fig. 2). The following steps are applied.

1. **Data fusion of feature vectors (D-step):** Here, the extracted features $F_t^1, \ldots, F_t^p$ in the time window $W_t$ are simply concatenated into a single feature vector $F_t^c = (F_t^1, \ldots, F_t^p) \in \mathbb{R}^P$, with $P = \sum_{i=1}^p d_i$.
2. **Classification (C-step):** The combined feature vector $F_t^c$ is classified into $C_t^c \in \Delta$ using a classifier mapping $c^c : \mathbb{R}^P \to \Delta$

\[
C_t^c := c^c(F_t^c),
\]
where $\Delta$ is defined by

\[
\Delta := \left\{ (q_1, \ldots, q_l) \in [0,1]^l \mid \sum_{i=1}^l q_i = 1 \right\}.
\]

Here, $l$ is the number of class labels.
3. **Temporal fusion of decisions over the whole time series (T-step):** Here

\[
C^o = \mathcal{F}(C_t^1, \ldots, C_t^T)
\]

is the classification result for the whole set of time windows $W_t$, $t = 1, \ldots, T$.

For the integration over time three different paradigms may be considered: majority vote (which counts only the local class decisions), voting, and combining the local class decisions utilizing a fusion mapping like average or any other linear combinations [26].

In the context of this paper, we consider three different fusion mappings for $T$ decision vectors $C_t^1, \ldots, C_t^T \in \Delta$:
Fig. 1. A set of \( p \) features \( F_{t1}, \ldots, F_{tp} \) is extracted from a local time window \( W_t \) located at time \( t \).

Fig. 2. Data fusion, Classification, Temporal fusion.

**Averaging:** Here the average of the \( T \) classification results is calculated by

\[
\mathcal{F}(C^1, \ldots, C^T) := \frac{1}{T} \sum_{i=1}^{T} C^i.
\]  

**Percentiles:** For this fusion method, for each class, \( j \in \{1, \ldots, l\} \) the classifier outputs \( C_j^i \in \mathbb{R} \) are sorted in decreasing order, e.g. there is a sequence \((\tau_j^i)_{i=1}^T\) of indices, such that \( C_j^{\tau_1} \geq \cdots \geq C_j^{\tau_T} \). Then for \( p \in [0, 1] \) the fusion mapping is defined by

\[
\mathcal{F}_p(C^1, \ldots, C^T)_j := C_j^{\lceil pT \rceil}.
\]

For \( p = 0 \), \( p = 1 \), and \( p = \frac{1}{2} \) the fusion mapping \( \mathcal{F}_p \) is the minimum, maximum and median fusion mapping [13].

**Voting:** Here each classifier output \( C^j = (C^j_1, \ldots, C^j_T) \) is sorted in decreasing order, e.g. there is a sequence \((\tau_j^i)_{i=1}^T\) of indices such that \( C_j^{\tau_1} \geq \cdots \geq C_j^{\tau_T} \). With \( R_j^i \) we denote the rank of class \( j \) in this sequence, e.g. \( R_j^{\tau_i^1} = R_j^{\tau_i^T} = j \). Based on the rank \( R_j^i \) a pay off or fusion mapping \( \mathcal{F}_{\beta, \theta} \) with \( \beta > 0 \) and \( \theta \in [0, 1] \) is given through

\[
\mathcal{F}_{\beta, \theta}(C^1, \ldots, C^T)_j := \frac{1}{T} \sum_{i=1}^{T} \left( 1 - \frac{1}{1 + \exp(-\beta(\frac{R_j^i-1}{T-1} - \theta))} \right).
\]
Here $\beta$ determines the slope and $\theta$ the location of the inflection point of the signum function. In Fig. 3 the applied voting functions are shown.

3. Radial basis function network classifiers

The RBF network is a neural network type which is used in many applications [8]. RBF is a well established method for the approximation and interpolation of multivariate functions, as well as for classification [1,21]. Recent developments in theory and applications of RBF networks can be found in [7,8,24]. A RBF network which classifies an input feature vector $x^\mu \in \mathbb{R}^d$ into one of $l$ different classes consists of $d$ input neurons, $l$ output neurons and a layer of $k$ nonlinear RBF neurons [22,23,16]. For the classification task, the target vector $y^\mu \in \{0, 1\}^l$ is encoded as a binary vector of length $l$ with exactly a single one. The index $i$ with $y^\mu_i = 1$ indicates that the input feature vector $x^\mu$ should be categorized to class $i \in \{1, \ldots, l\}$. The output of the $j$th RBF neuron is the value of the RBF $\varphi_j(x)$. In this paper, the Gaussian function is used $\varphi_j(x) = \exp(-\|x - c_j\|^2/2\sigma_j^2)$ where $\| \cdot \|$ denotes the Euclidean norm, $x \in \mathbb{R}^d$ the input vector, $c_j \in \mathbb{R}^d$ a so-called center or prototype vector and $\sigma_j \in \mathbb{R}$ a scaling parameter (vector or matrix valued scaling parameters $\sigma_j$ may also be used).
value of the $p$th output neuron is then given as a linear combination of the RBFs:
$$y_p = \sum_{j=0}^{k} w_{jp} \varphi_j(x).$$
Here $w_{jp}$ is the weight from the $j$th hidden neuron to the $p$th output neuron and $\varphi_0 \equiv 1$. The index of the output neuron with the maximal output specifies the class determined by the RBF network for the input $x^\mu$. In this paper, the centers $c_j$, the scaling parameters $\sigma_j$ of the basis functions and the output weights are adapted through a three phase learning procedure (see [25] for details on this three phase learning scheme for RBF networks). In the first learning phase the centers are initialized by Kohonen’s learning vector quantization (LVQ) method [12] and scaling parameters are initialized through a real-valued width parameter. Then the output weights are calculated separately in a second learning phase by a gradient descent optimization algorithm. Minimizing the mean squared error between target and net output. In the final third learning phase all network parameters, namely $c_j$, $\sigma_j$, and $w_j$ are adapted simultaneously in a backpropagation-like training procedure.

4. Application

We present results achieved by testing the algorithms on a dataset which contains sound patterns from 35 different cricket species. The dataset contains recordings of 3 or 4 individuals per species. The recordings are provided from Ingrisch [9], Nischk [15] and Heller. The recordings are from Thailand (used by Ingrisch [9]) and from Ecuador (used in the doctoral thesis by Nischk [15]). The sound patterns are stored in the WAV-format (sampling rate 44.1 kHz, 16 Bit sampling accuracy).

The cricket songs consist of sequences of sound patterns (chirps). Based on these chirps (sequences of so-called pulses, see Fig. 4a) the crickets are classified [15]. Therefore, we determine the on- and off-sets of the single pulses through the signal’s energy and two thresholds (see Fig. 4b) which is called end point detection in [19].

Fig. 4. Signal parameters of the species *Gryllus campestris* (time window 2000 ms). Fig. (b) shows only 500 ms. (a) Filtered cricket waveform. (b) The signal’s energy and the threshold functions for signal segmentation. (c) Distances between pulses.
As in speech recognition systems the endpoint detection is used to get the relevant parts of the speech signal, in bioacoustics this algorithm is applied to determine the on- and off-sets of single pulses. Details of the segmentation algorithm are given in [3].

The time windows $W_t$, $t = 1, \ldots, T$ are aligned at the on-sets and off-sets of the pulses and the features: pulse length, pulse distances and pulse frequency are calculated inside these windows [2]. These features serve as inputs for the RBF classifiers, and were extracted from $d$ consecutive pulses (see Eqs. (7)–(9)).

1) **Pulse length (PL):** Let $n$ be the number of pulses. Furthermore, let $\lambda_i$ be the onset of the $i$th pulse and $\mu_i$ be the offset of the $i$th pulse. Then the mean length of $d$ pulses is given by

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

(7)

where $d$ is the number of pulses used to extract one feature vector.

2) **Pulse distances (PD):** Let $\Delta = (\Delta_1, \ldots, \Delta_{n-1})$ be the distances between two pulses calculated by $\Delta_j = \lambda_{j+1} - \lambda_j$, $j \in \{1, \ldots, n-1\}$. Then the pulse distances are extracted using a $d$-tuple encoding scheme producing $n - d$ data points $T_i \in \mathbb{R}^d$

$$T_i := (\Delta_i, \Delta_{i+1}, \ldots, \Delta_{i+d-1}) \in \mathbb{R}^d, \quad i = 1, \ldots, n - d.$$

(8)

3) **Pulse frequency (PF):** Let $f_i \in \mathbb{R}$ be the frequency of the $i$th pulse extracted from the Fourier spectra of exactly this pulse. The algorithm searches the frequency band with the highest energy to extract the pulse frequency. Then the used feature is mean frequency

$$F_i := \frac{1}{d} \sum_{j=i}^{i+d-1} f_j \in \mathbb{R}, \quad i = 1, \ldots, n - d$$

(9)

of $d$ adjacent pulses.

At this point it should be emphasized that we tested several local and global features for this application, among these are LPC coefficients, time encoded signals, frequency contours, Parzen density of pulse distances, and the energy contour of single pulses (see [3] for details).

5. Results and discussion

Because of the limited data sets we utilize the cross validation method to evaluate the classifiers. The training set has been used to design the classifiers, and the test set for testing the performance of the classification task. The *leave-k-out test* with 4 cycles using 2–3 records of each species for training and 1 record of each species has been used for the classification test [1]. In the numerical experiments training and test records of the same species are drawn from different individuals.
Table 1

Classification rate (standard deviation—Std) for the three single features PL (pulse length), PD (pulse distances) and PF (pulse frequency), (avr) Average fusion, (P) Percentiles (P₀.₀₁, P₀.₂₅, P₀.₅₀, P₀.₇₅), and 12 (β, θ)-Voting schemes. For the single features average fusion is used for the temporal combination.

<table>
<thead>
<tr>
<th></th>
<th>PL</th>
<th>PD</th>
<th>PF</th>
<th>avr</th>
<th>P₀.₀₁</th>
<th>P₀.₂₅</th>
<th>P₀.₅₀</th>
<th>P₀.₇₅</th>
</tr>
</thead>
<tbody>
<tr>
<td>Err</td>
<td>90.94</td>
<td>34.65</td>
<td>92.13</td>
<td>6.59</td>
<td>10.63</td>
<td>7.41</td>
<td>6.84</td>
<td>9.74</td>
</tr>
<tr>
<td>Std</td>
<td>1.13</td>
<td>1.91</td>
<td>0.71</td>
<td>0.53</td>
<td>0.569</td>
<td>0.57</td>
<td>0.37</td>
<td>1.10</td>
</tr>
<tr>
<td>β = 500 θ = 0.020</td>
<td>β = 500 θ = 0.045</td>
<td>β = 500 θ = 0.075</td>
<td>β = 100 θ = 0.105</td>
<td>β = 100 θ = 0.025</td>
<td>β = 100 θ = 0.040</td>
<td>β = 100 θ = 0.080</td>
<td>β = 100 θ = 0.120</td>
<td></td>
</tr>
<tr>
<td>Err</td>
<td>6.26</td>
<td>11.61</td>
<td>13.56</td>
<td>15.32</td>
<td>6.52</td>
<td>7.13</td>
<td>11.37</td>
<td>12.31</td>
</tr>
<tr>
<td>Std</td>
<td>0.36</td>
<td>0.70</td>
<td>1.27</td>
<td>0.59</td>
<td>0.37</td>
<td>0.45</td>
<td>0.51</td>
<td>0.59</td>
</tr>
<tr>
<td>β = 30 θ = 0.025</td>
<td>β = 30 θ = 0.070</td>
<td>β = 30 θ = 0.130</td>
<td>β = 10 θ = 0.200</td>
<td>β = 10 θ = 0.000</td>
<td>β = 10 θ = 0.050</td>
<td>β = 10 θ = 0.120</td>
<td>β = 10 θ = 0.200</td>
<td></td>
</tr>
<tr>
<td>Err</td>
<td>6.65</td>
<td>8.38</td>
<td>9.78</td>
<td>12.83</td>
<td>8.20</td>
<td>8.13</td>
<td>8.23</td>
<td>9.01</td>
</tr>
<tr>
<td>Std</td>
<td>0.30</td>
<td>0.71</td>
<td>0.69</td>
<td>0.49</td>
<td>0.59</td>
<td>0.66</td>
<td>0.49</td>
<td>0.77</td>
</tr>
</tbody>
</table>

For the RBF classifiers the best classification result based on a single feature was achieved with the pulse distances (PD) (error rate 34.65%). Whereas the classifier performances based on the single features PF or PL were really bad, the combination of all three features through average fusion performs significantly better (error rate 6.59%) than the best single feature (PD). This is a very good result for such a multi-class pattern recognition problem with 35 different categories. The results in Table 1 show also, that voting with β = 500 and θ = 0.02 leads to the best classification results for this dataset (error rate 6.22%). In Fig. 3a it can be observed that with this function only the first winner (class with the highest classifier output) is used for the temporal fusion. This fusion technique called majority voting for this case. The results for voting the classifier decisions are strongly dependent on the parameter θ which determines the location of the inflection point of the voting function (see Table 1 and Fig. 3). The second best result was achieved with voting with β = 100 and θ = 0.025 (see Fig. 3b). Average fusion leads to the third best result (error 6.59%). For percentiles and voting with worse parameter settings the results were in the range of 6.8 to 13.6%. k-nearest neighbour classifier techniques which have been studied in Dietrich et al. [3] seem to be not applicable for this data and lead to error rates > 10% for all fusion methods.

The best classification performances are achieved with the fusion mappings majority voting and average fusion which have been used in other applications [13,26]. The study also shows that cricket songs can be classified reliably with RBF networks together with temporal fusion techniques on multiple feature sets. Basically, they had to solve the same task as cricket females with their “real” neuronal substrate. The results presented here might serve as a heuristic tool for new experiments to understand cricket phonotaxis and neural processing, especially for tropical species. Up to now, neurophysiological experiments have been limited to Gryllus spp., which inhabit acoustically less complex biotopes, with much lesser cricket diversity. Though it might be naive to assume that there is any such thing as a neuronal correlate of the mechanisms de-
scribed here, our results provide important hints for crucial experiments. Parzen density emerged as an important feature which has been overlooked by traditional neurobiologists [5]. Finally, and independent from these questions of fundamental research, the classifiers described here could be implemented into Rapid Assessment protocols, to detect, classify and monitor tropical cricket diversity.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful and constructive comments. DORSA forms part of the Entomological Data Information System (EDIS) and is funded by the German Ministry of Science and Education (BMBF). We are grateful to Sigfrid Ingrisch (ZFMK Bonn, Germany), Klaus Heller and Frank Nischk, for providing their sound recordings, suggestions and discussions.

References


Friedhelm Schwenker received the Ph.D. degree in mathematics from the University of Osnabrück in 1988. From 1989 to 1992 he was a scientist at the Vogt-Institute for Brain Research, University of Düsseldorf. Since 1992 he is a researcher and lecturer at the Department of Neural Information Processing, University of Ulm. His main research interests are neural networks, machine learning, data mining, pattern recognition, applied statistics, and approximation theory.

Christian Dietrich was born in Oberstdorf (Germany) in 1970. He received his diploma in Computer Engineering 1998 from the University of Applied Sciences Augsburg. Since 2000, he works as a researcher in the Department of Neural Information Processing at the University of Ulm. His research interests, which are being developed in his doctoral thesis include the classification of bioacoustic time series based on the combination of classifiers trained on different feature subsets.

Hans A. Kestler is an electrical engineer and scientist at the Department of Neural Information Processing at the University of Ulm. Since his graduation from the TU-München, he is involved in interdisciplinary research with the Departments of Internal Medicine at the University of Ulm on the topics of signal and image analysis in cardiology and pathology, and functional genomics. His main research interests are pattern recognition, applied statistics and signal processing.
Klaus Riede studied zoology and biocybernetics at the Universities of Frankfurt and Tübingen. After his graduation in zoology he worked for the Max-Planck-Institute for behavioural physiology, Tübingen, on rainforest grasshopper biology in South America. From 1991 to 1997 he worked as research assistant at the University of Freiburg. Since then, he works at the Center for Development Research, University of Bonn, in cooperation with the Zoological Museum. His present research topics are biodiversity informatics and geographical information systems, and their application for conservation.

Günther Palm studied mathematics at the Universities of Hamburg and Tübingen. After his graduation in mathematics he worked at the Max-Planck-Institute for Biological Cybernetics in Tübingen on the topics of nonlinear systems, associative memory and brain theory. In 1983/84, he was a fellow at the Wissenschaftskolleg in Berlin. From 1988 to 1991 he was professor for Theoretical Brain Research at the University of Düsseldorf. Since then he is professor for computer science and head of the Department of Neural Information Processing at the University of Ulm. His research topics in computer science include information theory and applications of artificial neural networks in speech, vision, robotics, sensor-fusion and pattern recognition.