ECONSTOR
Make Your Publications Visible.

A Service of

ZBW Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Garczarek, Ursula; Weihs, Claus; Ligges, Uwe

Working Paper

Prediction of notes from vocal time series produced by singing voice

Technical Report, No. 2003,01

Provided in Cooperation with:
Collaborative Research Center 'Reduction of Complexity in Multivariate Data Structures' (SFB 475), University of Dortmund

Suggested Citation: Garczarek, Ursula; Weihs, Claus; Ligges, Uwe (2003) : Prediction of notes from vocal time series produced by singing voice, Technical Report, No. 2003,01, Universität Dortmund, Sonderforschungsbereich 475 - Komplexitätsreduktion in Multivariaten Datenstrukturen, Dortmund

This Version is available at:
https://hdl.handle.net/10419/49328

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

WWW.ECONSTOR.EU

Mitglied der Leibniz-Gemeinschaft

Prediction of Notes from Vocal Time Series Produced by Singing Voice

Ursula Garczarek[1], Claus Weihs, Uwe Ligges

Fachbereich Statistik, Universität Dortmund
44221 Dortmund, Germany

Abstract

Aiming at optimal prediction of the correct note corresponding to a vocal time series we trained a classification algorithm on the basis of parts of interpretations of Tochter Zion (Händel) and tested the algorithm on the remaining parts.

As classification algorithm we use a radial basis function support vector machine together with a "Hidden Markov" method as a dynamisation mechanism and some smoothing for categorical data. With this we were able to obtain a minimum of 5% average classification error and a maximum of 26% on data from an experiment with 16 singers.

Keywords: Radial Basis Functions, Support Vector Machines, Classification, Time Series, Prediction, Singing Voice

[1] garczarek@statistik.uni-dortmund.de

1 Introduction

Analogously to speech recognition systems on computers, our aim is to train a classification algorithm in order to be able to predict the correct note corresponding to a vocal time series.

As classification algorithm we use a radial basis function support vector machine together with a "Hidden Markov" method as a dynamisation mechanism. With this we were able to obtain a very reasonable predictive power on data from an experiment with 16 singers singing "Tochter Zion" (Händel).

The paper is structured as follows: In Section 2 data preparation is described. In Section 3 we describe the statistical methods used for prediction. In Section 4 the results are reported and discussed.

2 Data Preparation

The time series data from an experiment with 16 singers singing "Tochter Zion" (Händel) (cp. Weihs et al., 2001) is cut into overlapping sections of 256 observations, overlaps starting in the middle of the preceding section. For all of these sections the periodogram (cp. Brockwell and Davis, 1991) has been calculated (data sampled with 11025 Hz in a 16 bit resolution). Hence, we get roughly 86 ($= 2 \times (11025/256)$) periodograms per one second of sound, whereas the duration of the whole song is roughly 60 seconds.

In order to reduce complexity, we restricted the frequencies of the periodograms for further analyses to those frequencies that can be performed by the human voice, including fundamental frequencies and a reasonable amount of overtones. In particular, we chose the 40 Fourier frequencies in [258.4 Hz, 1938.0 Hz] for women, and 40 Fourier frequencies in [129.2 Hz, 1808.8 Hz] for men. Each of these frequencies becomes a variable in the following sections.

"Tochter Zion" has the form `ABA`. The first parts `A` and `B` are used as the learning set **L**. For each section in these two parts the algorithm is given the correct note (corresponding to a fundamental frequency) ideally sung in the corresponding time period.

The correct fundamental frequencies are derived from the piano accompaniment as well as starting and ending points of the notes. A suitable segmentation procedure has been described by Ligges et al. (2002). For our purposes, the segmentation results were manually approved or corrected, respectively.

The last part `A` is used for assessing the goodness of the note classification (test set $\mathbf{T}$), i.e. the correct note is compared to the note predicted by the trained classifier.

3 Statistical Methods

3.1 Learning of Prediction Rules

After data preparation, different prediction rules are built on the learning data that represent different stages of a multi-step learning procedure:

1. **Basic quantization of evidence**
   The evidence on each note $n \in \mathbf{N} := \{1, ..., N\}$ in the observed periodogram $\vec{x}_t$ at time point t, $t = 1, ..., T$, is quantized by the membership functions $m : \mathbf{N} \times \vec{x} \in \mathbf{X}$ of support vector machines with radial basis functions (see Section 3.2). The membership functions are scaled such that for any given periodogram they define a probability distribution over the notes: $m(n, \vec{x}) \geq 0$ and $\sum_{n=1}^{N} m(n, \vec{x})) = 1$ for all $n \in \mathbf{N}$ and all $\vec{x} \in \mathbf{X} \subseteq \mathbb{R}^K$.

2. **Static Prediction**
   In the static fashion the note $\hat{n}_t^s$ with the highest current evidence $m(n, \vec{x}_t)$ from the periodogram $\vec{x}_t$ is predicted:

$$\hat{n}_t^s \quad := \quad \arg\max_{n \in \mathbf{N}} (m(n, \vec{x}_t)) \tag{1}$$

3. **Estimation of transition probabilities between notes**
   The transition probabilities between true notes are estimated by the observed frequencies on the learning set.

4. **Dynamized Prediction**
   A Hidden Markov Model (see Section 3.3) is instantiated with these transition

probabilities and the scaled membership values as emission probabilities. With this model, we estimate the probability $p(n|\vec{x}_1, ..., \vec{x}_t)$ that the true note is n given the current observed periodogram $\vec{x}_t$ and all observed periodograms before $\vec{x}_1, ..., \vec{x}_{t-1}$. Based on these estimates, the second rule predicts the note with highest estimated probability:

$$\hat{n}_t^d := \arg\max_{n \in \mathbf{N}} (\hat{p}(n|\vec{x}_1, ..., \vec{x}_t)) \tag{2}$$

5. **New quantization of evidence from predictors**
   How often a note n is confused with another by some predictor $\hat{n} \neq n$ on the learning set is counted in the so-called confusion matrix:

$$\mathbf{C}(\mathbf{L}) = \begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,K} \\ c_{2,1} & c_{2,2} & \dots & c_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ c_{K,1} & c_{K,2} & \dots & c_{K,K} \end{bmatrix} \tag{3}$$

   with $c_{i,j} := \sum_{t=1}^{T_\mathbf{L}} \mathbb{I}_i(\hat{n}_t)\mathbb{I}_j(n_t)$. Divided by its sum $c_{i,.} := \sum_{t=1}^{T_\mathbf{L}} \mathbb{I}_i(\hat{n}_t)$ each row of $\mathbf{C}(\mathbf{L})$ gives the relative frequencies on the learning set that j was the true note, given that i was predicted. The standardized rows are thus estimators of the conditional probabilities for each note $p(j|\hat{n} = i)$:

$$\hat{p}(j|\hat{n} = i) := \frac{c_{i,j}}{c_{i,.}}.$$

   That way, $\hat{p}(n|\hat{n}^s)$ and $\hat{p}(n|\hat{n}^d)$, $n, \hat{n}^s, \hat{n}^d \in \mathbf{N}$, quantize the evidence one gets from the predictions $\hat{n}^s$ or $\hat{n}^d$ for the note n.

6. **Smoothed Static and Dynamic Prediction**
   Especially for professional singers vibrato is observed in singing. In order not to mix vibrato with tone changes, a smoothing algorithm with a window size adapted to the individual singer is used. To do smoothing at time point t, one uses the evidence for the notes one gets from $\hat{p}(n|\hat{n}_s^s)$ or $\hat{p}(n|\hat{n}_s^d)$ in some window around t: $s \in W_t(w) := [max(t - w, 1), min(t + w, T)]$. To predict, equal weight is given to the evidence at any time point s in the interval:

$$\hat{n}_t^{ss} := \arg\max_{n=1,...,N} \left(\sum_{s \in W_t} \hat{p}(n|\hat{n}_s^s) \right) \tag{4}$$

$$\hat{n}_t^{sd} := \arg\max_{n=1,...,N} \left(\sum_{s \in W_t} \hat{p}(n|\hat{n}_s^d) \right) \tag{5}$$

The optimal size of $w \in \{1, ..., L\}$ is determined in terms of learning error rate. L is some definition of the minimum length of a tone. Here, $L = 20$ which is the length of time of a quaver in the given experiment.

3.2 Support Vector Machines with Radial Basis Functions (RBFSVM)

For the quantization of the strength of membership we use support vector machines using radial basis function kernels (cp. Vapnik, 1995, and Schölkopf, 1998) as implemented in the Support Vector Machine (SVM) toolbox 2.51 for Matlab by Schwaighofer (2002).

SVMs are identifying so called "support vectors" most important for the distinction between objects of two classes given measurements $\vec{x} \in \mathbf{X}$ on the objects. We train support vector machines for each note n ($y(n) := 1$) against all others $\{1, ..., N\} \setminus n$ ($y(n) := -1$) on the basis of the observed periodogram $\vec{x} \in \mathbf{X}$. The support vectors are defining points of the decision surface between these two classes.

In the linearly separable case, any finite number of observations from the two classes in some finite data set $\mathbf{D}$ can be separated without errors by linear hyperplanes. The support vectors of the basic linear SVM span those two parallel hyperplanes that separate the observations in $\mathbf{D}$ with the largest margin between them. That is, the hyperplane is defined by the equation:

$$\vec{w}_n(\mathbf{D})'\vec{x} + \theta_n(\mathbf{D}) \quad := \quad 0,$$

where the normal $\vec{w}_n(\mathbf{D})$ of the hyperplane is given by

$$\vec{w}_n := \sum_{t=1}^{T_\mathbf{D}} \alpha_t(n) y_t(n) \vec{x}_t \tag{6}$$

with non-negative parameters $\alpha_t(n)$, $t = 1, ..., T_\mathbf{D}$ that are non-zero for the so-called support vectors. The optimal parameters $\vec{w}_n(\mathbf{D})$ and $\theta_n(\mathbf{D})$ are chosen such that all objects of the data set with $y_t(n) = 1$ lie "above" and all objects with $y_t(n) = -1$ lie "below" the hyperplane, $t = 1, ..., T_\mathbf{D}$, and such that the size of the margin $\frac{2}{\|\vec{w}_n(\mathbf{D})\|}$ is maximal.

In the linearly non-separable case, one first of all does the kernel trick: one maps the data via some function Φ into some higher dimensional feature space, and constructs a separating hyperplane with maximum margin there. By the use of a kernel function, it is possible to compute the separating hyperplane without explicitly carrying out the map into the feature space (Schölkopf and Smola, 2002, p.15). As in some arbitrarily high dimensional feature space one can separate any two finite dimensional groups of distinct objects without error, in case of overlapping groups one has to fight overfit. In case of support vector machines, one reaches a higher fit the smaller the margin is. To allow for errors, one does no longer only maximize the margin but its sum with some error penalty term in dependence of some parameter C. The larger C is, the higher the penalty is for errors.

We use radial basis function kernels which are local gaussian densities around data points:

$$\mathbf{K}\left(\vec{x},\vec{y}|\sigma^2\right) = \exp\left(\frac{-\sum_{k=1}^{K}(x_k-y_k)^2}{2\sigma^2}\right), \tag{7}$$

where σ^2 defines the width of the RBF-kernel.

The decision surface between note n and all others is now defined by the equation

$$\left(K(\vec{w}_n(C,\sigma^2,\mathbf{D}),\vec{x}) + \theta_n(C,\sigma^2,\mathbf{D})\right) \;:=\; 0 \tag{8}$$

where the optimal $\vec{w}_n(C,\sigma^2,\mathbf{D})$ is also some weighted sum of support vectors in the data set $\mathbf{D}$ as in (6). The size of the margin in the feature space is

$$\frac{2}{K(\vec{w}_n,\vec{w}_n|\sigma^2)}.$$

To quantize the membership of a time point t with observed periodogram $\vec{x}_t \in \mathbf{X}$ to the notes $n \in \mathbf{N}$ we use the signed euclidean distance to the corresponding hyperplanes:

$$m^*(n,\vec{x}|C,\sigma^2,\mathbf{D}) \;=\; \frac{K(\vec{w}_n(C,\sigma^2,\mathbf{D}),\vec{x}|\sigma^2) + \theta_n(C,\sigma^2,\mathbf{D})}{\sqrt{K(\vec{w}_n(C,\sigma^2,\mathbf{D}),\vec{w}_n(C,\sigma^2,\mathbf{D}))}}. \tag{9}$$

To let the membership values for some observed periodogram $\vec{x} \in \mathbf{X}$ define some probability distribution over the set of notes $\mathbf{N}$, we standardize them with the so-called softmax transformation (Bridle, 1990):

$$m(n,\vec{x}|C,\sigma^2,\mathbf{D}) \;=\; \frac{\exp\left(m^*(n,\vec{x}|C,\sigma^2,\mathbf{D})\right)}{\sum_{n\in\mathbf{N}}\exp\left(m^*(n,\vec{x}|C,\sigma^2,\mathbf{D})\right)}.$$

Two parameters have to be chosen and are not part of the automatic optimization procedure of RBFSVMs: The parameter C that controls the trade off between margin maximization and error minimization and the kernel width σ^2. To adjust (C, σ^2) otherwise, one can use the Bernoulli loss experiment, and any optimization method that finds extremal points of multidimensional functions to minimize some estimate of the error probability in dependence of (C, σ^2). We estimate the error probability by cross-validation: we partitioned the learning set by means of sampling stratified on notes such that 75% of the objects from the learning set form the training set $\mathbf{T}$, and 25% the validation set $\mathbf{V}$.

One often applied optimization method is the gradient decent algorithm used by Chapelle et al. (2001) for (heavier) SVM model selection. As many of these methods, the gradient decent algorithm might get stuck in local minima, yet, it does not if the error surface is convex. Nevertheless, given certain restrictions on the functions' surface, these algorithms are *optimal* optimization strategies. Otherwise, they are *heuristic* optimization strategies (Luenberger, 1973).

Based on model assumptions on the error surface, one can alternatively apply methods from statistical experimental design to optimize parameters. We assumed here that the error probability can be approximated by some quadratic function of $(\log_{10}(C),\ -\log_e(\sigma^2)/K)$ restricted to the cube $[-1,\ 2]^2$.

That way we restricted the search for the best parameters $(C,\ \sigma^2)$ on the rectangle $[\frac{1}{10},\ 100] \times [\frac{K}{e^2},\ Ke]$. Defining 5 optimal experimental points according to a central-composit plan for two variables (see e.g. Weihs and Jessenberger, 1999) the quadratic function was fitted and optimal parameters were determined by the minimum of the fitted quadratic function on the rectangle.

3.3 Hidden Markov Model

Hidden Markov Models are used to model time series. In a Hidden Markov Model, one assumes that the distribution $P_{\vec{X}_t}$ of the random vector $\vec{X}_t$ of observables of some system only depends on the state $S_t = s$ the system is in: $P_{\vec{X}_t} \equiv P_{\vec{X}|s}$ for all $t = 1, ..., T$ with $S_t = s \in \mathbf{S}$. The time dependency results from the dependency among states. This is modelled by some markov chain: that is the distribution of the

state of the system at time point t depends on the past only through the last state before: $P_{S_t} = P_{S|s^-}$ for all $t = 2, ..., T$ with $S_{t-1} = s^- \in \mathbf{S}$. Therefore parameters of the distributions in an HMM are the states' *transition probabilities* $p(s|s^-), s, s^- \in \mathbf{S}$ and the the so-called *emission probabilities* $p(\vec{x}|s)$, $\vec{x} \in \mathbf{X}$, $s \in \mathbf{S}$.

In our case a time series represents a sung interpretation of "Tochter Zion". The system is the singer, the states are the notes $n_t \in \{1, ..., N\}$ and the observables the periodograms $\vec{X}_t$, $t = 1, ..., T$. Clearly, since the 256 observation sections of the time series are shorter than a note, the note sung in one section depends on the note sung in the preceding section.

For each sequence of notes $n_t \in \mathbf{N}$, $t = t_1, ..., T$ and known prior distribution for N_{t_1}, the transition and emission probabilities determine the probability to have observed $\vec{x}_t$, $t = t_1, ..., T$. Therefore, the probability $p(n|\vec{x}_{t_1}, ..., \vec{x}_T)$ of some note n at the end of some sequence of time points for which we do not know the true note but only the periodograms $\vec{x}_t$, $t = t_1, ..., T$, is the sum of the probabilities of any path leading to n. It can be calculated using the forward step in the forward-backward procedure for finding the next state in HMMs (cp. Rabiner and Juang, 1993).

4 Results

Note prediction was done on the basis of two different algorithms: RBFSVM with Hidden Markov model (RBFSVM-HMM) and without such model (RBFSVM-static). Moreover, on these two algorithms smoothing (labelled *smoothed*) was applied or not (*pure*). Table 1 shows the prediction error rates on the last part `A` of "Tochter Zion" for the various combinations of algorithms and each singer.

From the last column of the table it is apparent that 4 singers can be predicted with less than 7.5% error, whereas other 4 singers only with more than 15% error.

Figures 1 (a, b) show the predicted notes vs. the ideal notes in the various 256 observations sections generated by the optimal algorithm for the best and the worst predictable singer, i.e. for T1 and B4. The figures should be interpreted as follows: Ideal notes are indicated in grey, sung notes in black. Ideal notes are overplotted by sung notes. Thus a horizontal black line on white background indicates an error.

singer	pure		smoothed	
	static	HMM	static	HMM
S1	20.5	14.8	18.5	14.6
S2	15.4	10.9	11.5	10.1
S3	20.2	13.2	19.5	12.1
S4	14.8	6.7	9.7	5.6
A1	23.2	18.9	20.1	17.8
A2	28.6	20.3	26.1	19.4
A3	12.8	5.6	10.0	5.4
A4	17.2	13.9	12.3	13.6
A5	23.9	10.6	22.3	10.2
T1	10.6	5.5	8.8	5.4
T2	18.3	9.5	14.5	6.3
T3	18.4	11.5	14.0	10.3
B1	17.4	10.9	12.3	8.3
B2	28.0	21.2	23.5	19.6
B3	20.0	14.1	19.2	14.3
B4	29.0	27.4	24.5	27.7
minimum	10.6	5.5	8.8	5.4
median	19.2	12.4	16.5	11.2
maximum	29.0	27.4	26.1	27.7

Table 1: Prediction error rates on the last part `A` of "Tochter Zion".

Very low black lines also indicate misclassification, that can often be identified as caused by breathing periods or silence, respectively. Obviously, in "B4" more errors occur than in "T1".

The figure help interpreting the possible causes of error rates. Singers with lots of singing errors cannot be well predicted, like singer "B4" who has problems with correct timing.

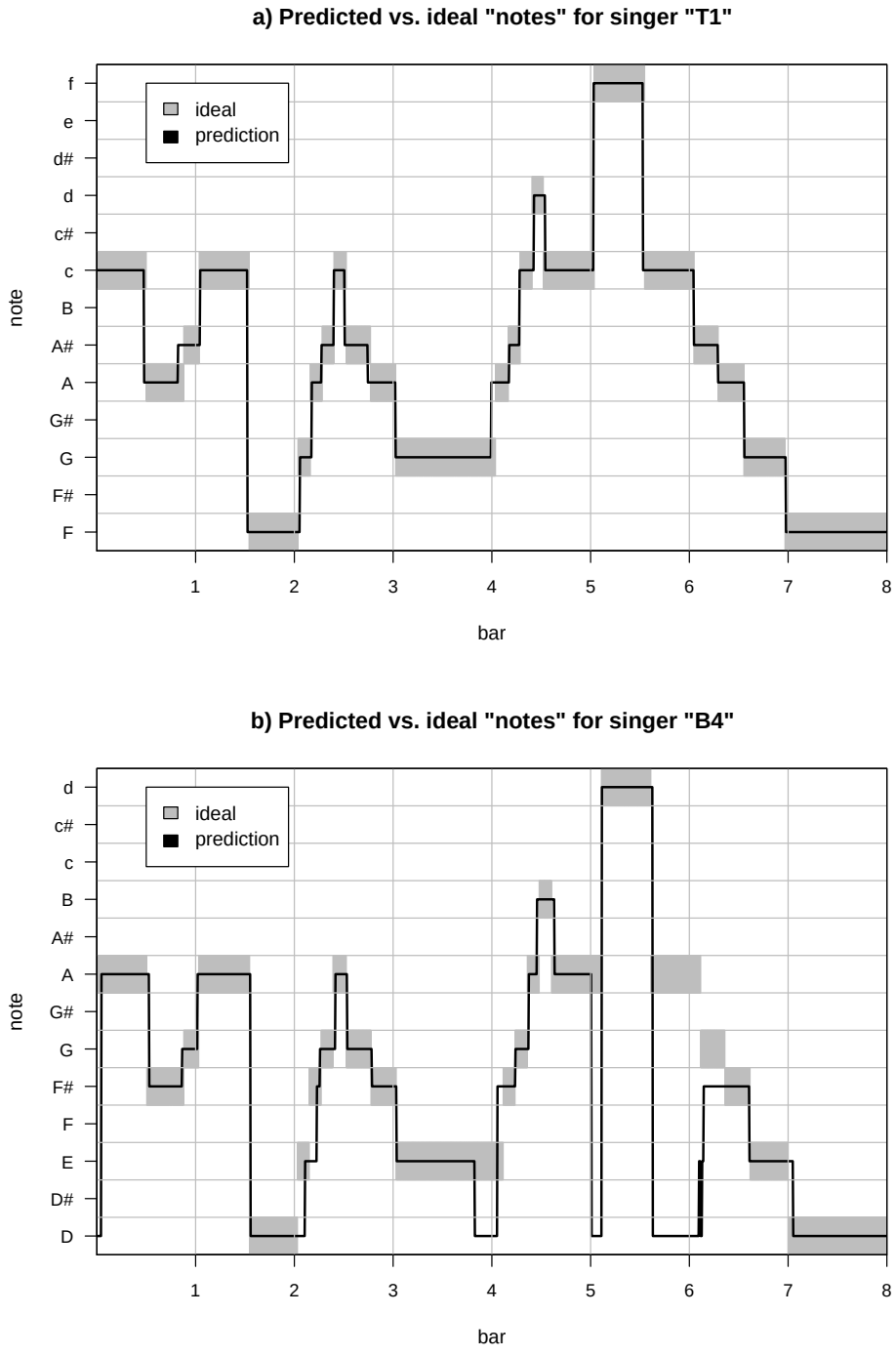

Figure 1: Predicted notes (black) vs. ideal notes (grey) for singers T1 and B4.

5 Conclusion

With our classification algorithm we were able to nearly optimally predict singers who delivered "correctly" sung notes in the training period. The corresponding goodness of classification appears to be well enough to generate nearly correct midi files corresponding to songs sung by such singers.

Unfortunately, the prerequisites are quite strong. For the learning step a singer has to sing each possible note of the song separately, otherwise a segmentation and corresponding "ideal" notes must be given by the supervisor. In our experiment, the first two parts `AB` of the whole song (`ABA`) were used for learning, while the last part `A` was used for assessing the goodness of prediction. Obviously, errors of the singers are correlated, as well as probabilities of note changes. Therefore, if arbitrary notes are presented for learning, error rates might become worse.

Acknowledgements. The financial support of the Deutsche Forschungsgemeinschaft (SFB 475, "Reduction of complexity in multivariate data structures") is gratefully acknowledged.

References

[1] BRIDLE, J. S. (1990): Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. In: F. FOGELMAN SOULIÉ and J. HÉRAULT (Eds.): *Neuro-computing: Algorithms, Architectures and Applications.* Springer, Berlin, 227–236,

[2] BROCKWELL, P. J., DAVIS, R. A. (1991): *Time Series: Theory and Methods.* Springer, New York.

[3] CHAPELLE, O., VAPNIK, V., BOUSQUET, O., MUKHERJEE, S. (2001): Choosing Multiple Parameters for Support Vector Machines. *Machine Learning, 46*, 131.

[4] LIGGES, U., WEIHS, C., HASSE-BECKER, P. (2002): Detection of Locally Stationary Segments in Time Series. In: W. HÄRDLE and B. RÖNZ (Eds.):

CompStat2002 – Proceedings in Computational Statistics – 15th Symposium held in Berlin, Germany. Physika Verlag, Heidelberg, 285–290.

[5] LUENBERGER, D. G. (1973): *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, Reading, Mass.

[6] RABINER, L., JUANG, B.-H. (1993): *Fundamentals of Speech Recognition*. Prentice Hall, New Jersey.

[7] SCHÖLKOPF, B. (1998): Support-Vektor-Lernen. In: G. HOTZ et al. (Eds.): *Ausgezeichnete Informatikdissertationen*. Teubner, Stuttgart, 135–150.

[8] SCHÖLKOPF, B., SMOLA, A. (2002): *Learning with Kernels*. MIT Press, Cambridge, MA.

[9] SCHWAIGHOFER, A. (2002): *SVM toolbox for Matlab*. See also: http://www.igi.tugraz.at/aschwaig/software.html.

[10] VAPNIK, V. (1995): *The Nature of Statistical Learning Theory*. Springer, New York.

[11] WEIHS, C., BERGHOFF, S., HASSE-BECKER, P., LIGGES, U. (2001): Assessment of Purity of Intonation in Singing Presentations by Discriminant Analysis. In: J. KUNERT and G. TRENKLER (Eds.): *Mathematical Statistics and Biometrical Applications*. Josef Eul Verlag, Lohmar, 395–410.

[12] WEIHS, C., JESSENBERGER, J. (1999): *Statistische Methoden zur Qualitätssicherung und -optimierung in der Industrie*. Wiley-VCH, Weinheim.