University of Music and Dramatic Arts Graz

Institute of Electronic Music and Acoustics

Evaluation of Robust Features for Singing Voice Detection

Project Thesis

Author: Patrick Gampp

Graz, December 13, 2010

Supervisor : Dr. Alois Sontacchi

Abstract

The detection of singing voice segments within music signals is an important object of research in the field of music information retrieval, since it serves as an essential pre-stage for applications like singer identification, lyrics recognition, singing melody extraction and many more.

The objective of this thesis is the implementation and evaluation of an pattern recognition system with the capability of detecting singing voice in music signals. For this purpose, a support vector machine classifier is utilized in conjunction with MFCC features, including also long-term features of MFCCs and their delta features. Furthermore, an energy-based feature is proposed. Feature subset selection has been carried out by means of linear discriminant analysis together with sequential forward and backward elimination subset space search strategies. The resulting subsets were evaluated in combination with the classifier using 10-fold cross-validation, resulting in a mean accuracy of 75.6 % with a standard deviation of 2.5 % for the best subset.

Finally, the system was tested with a database, which was provided by Mathieu Ramona and obtained a mean accuracy of 69.7 %.

Kurzfassung

Die Detektion von Singstimmen innerhalb eines Musiksignals ist ein wichtiger Forschungsgegenstand im Bereich des Music Information Retrievals, da sie eine entscheidende Vorstufe für Anwendungen wie Sänger-Identifikation, Liedtext-Erkennung, Gesangs-Extraktion und viele mehr darstellt. Das Ziel dieser Arbeit ist die Implementierung und Beurteilung eines Mustererkennungs-Systems mit der Fähigkeit, Singstimmen in Musiksignalen detektieren zu können. Hierzu wird ein Support Vector Machine Klassifizierer in Verbindung mit MFCC-Merkmalen, sowie Langzeit-Merkmalen von MFCCs und ihren Delta-MFCCs, verwendet. Darüber hinaus wird ein weiteres Merkmal aufgestellt, welches auf Energie basiert. Die Auswahl von Merkmals-Teilmengen wurde mit Hilfe der Diskriminanzanalyse und den Suchstrategien der sequentiellen Forwärtsselektion sowie der Rückwärtseliminierung durchgeführt. Die ermittelten Teilmengen wurden mit Hilfe des Klassifizierers und einer 10-fachen Kreuzvalidierung evaluiert. Die beste Merkmalsmenge erreichte eine mittlere Genauigkeit von 75.6 % bei einer Standardabweichung von 2.5 %.

Zum Abschluss wurde das System mit einer Datenbank, die von Mathieu Ramona bereitgestellt wurde getestet, und erreichte hierbei eine mittlere Genauigkeit von 69.7 %.

Contents

Chapter 1

Introduction

The vast quantity of music data that can be found in online music stores or other music collections requires methods to handle and access this data via perceptive criteria. This field of research about the automatic extraction of meta data from music is called music information retrieval (MIR).

Especially the singing voice is an essential element of western popular music. Therefore, diverse applications based on the detection of singing voice are objects of research: singer identification [1–5], singing melody extraction [6–8], singing performance/quality evaluation [9, 10], singing voice transcription [11, 12], query-by-singing/humming [13], lyrics recognition [14], automatic identification of singing language [15], etc. The detection of singing voice segments functions as pre-processing for the applications referred to above.

In the field of speech recognition, there has been a lot of research effort since the nineteen-fifties. However, the application of these developed systems to singing voice has not achieved satisfactory results even though the vocal apparatus is the same. Speech and singing differ significantly with regard to production and perception. Singing voice for instance contains approximately 90 % voiced sounds, whereas speech embodies only 60 % voiced and 40 % unvoiced sounds for the English language [16]. Another unique feature primarily of opera singing is the so-called *singer's formant*, which manifests itself in a frequency emphasis around 2-4 kHz. It helps the voice to be clearly audible in the presence of the orchestra, without amplification of the singer's voice [17, 18].

The definition of features, which make singing voice content distinguishable is complicated by the fact, that singing voice exhibits primarily harmonic partials, which overlay with the harmonic portions of accompanying music instruments.

The objective of this thesis is the implementation of a pattern recognition system for singing voice detection, i.e. the classification of segments into singing voice and non singing voice segments.

The thesis is organized as follows: chapter 2 will briefly outline the basic elements of a pattern recognition system with particular emphasis on content, which plays an important role for singing voice detection and was utilized within the scope of this thesis. In chapter 3, the implemented singing voice detection system will be discussed in its design. This chapter also addresses the tuning and setting of classifier parameters as well as the selection of feature subsets. The results of the implemented system on a test database which was assembled and provided by Mathieu Ramona [19] will be presented in chapter 4. Chapter 5 will discuss the design of the implemented system and its results. Finally, chapter 6 will draw conclusions for further improvements.

Chapter 2

Pattern Recognition

Pattern recognition is a part of machine learning and the term stands for the assignment of a label to a incoming object, which is represented by features which allow to characterize the signal. Well-known applications are for instance speech recognition, face recognition and finger print recognition among others. This chapter shows the basic components needed for a pattern recognition system (see figure 2.1) and describes them with respect to the problem of singing voice detection:

- The incoming digital signal is preprocessed for instance by noise reduction or the normalization of the range of values.
- The patterns are represented by features, which quantify characteristics of an object. As a consequence, the data size is reduced. Features are specifically adapted to the application in which they are used. They should have low calculation costs and should be robust against irrelevant transformations of the input signal.
- A classifier assigns the objects, represented by features to a class. Numerous pattern recognition applications are of high complexity. There-

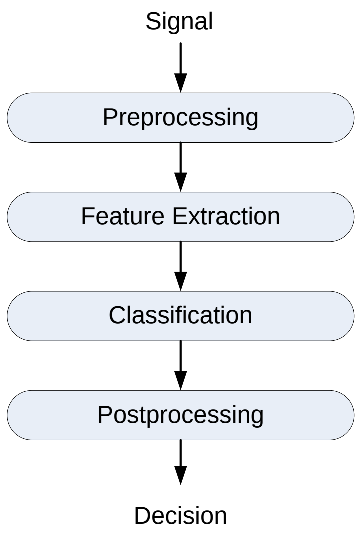

Figure 2.1: Basic elements of a pattern recognition system

fore, the classifier has to learn, which features belong to a certain class. This can be done with several learning strategies. Here, a supervised learning strategy is used: the model parameters of the classifier can be adjusted with the help of training data, where the class membership of each feature vector is known.

- The decisions of the classifier can be post-processed by analysis of adjacent segments. For example decisions can be filtered in time-domain by means of median filtering, or sequences of decisions can be analyzed by a Hidden Markov Model as used in [19].

2.1 Features

This section describes a selection of features used in connection with singing voice detection in literature and is divided into acoustic and statistical features. All discussed features are based on the frequency spectrum of the signal. This is calculated via Short-time Fourier transform (STFT) [20]. The window-size of the STFT is chosen such, that the signal in this time interval can be regarded as quasi-stationary. Typically, values for window size are around 25 ms with a hopsize around 10 ms [21].

2.1.1 Acoustic features

Mel frequency cepstral coefficients

Mel frequency cepstral coefficients (MFCC) [22] were originally used in the context of speech communication but are also widespread in music signal description. Rocamora and Herrera [21] compared numerous audio features for singing voice detection and the MFCC were the most successful feature.

First, the signal is divided into frames where the fast Fourier transform (FFT) is applied to. That means, one MFCC vector is calculated for each frame. Then, the logarithm of the magnitude spectrum is taken. This is perceptually motivated since loudness perception was found out to be of logarithmic nature and the phase information of the spectrum is nonessential compared to the magnitude [23]. The next step is to group and average the frequency bins according to the mel frequency scale which is approximately linear up to 1000 Hz and afterwards logarithmic. Again, this is a perceptually motivated measure since also the bandwidth of the auditory filters is increasing with frequency. The grouping step is realized via triangular filters and results in typically 40 bands. This signal is reduced to typically 13 MFCC via discrete cosine transform (DCT).

Log Frequency Power Coefficients (LFPC)

Nwe et al. [24] use the LFPC, where the signal frames are filtered by a 12-band filterbank with their filter bandwidth increasing logarithmically. The feature is defined as:

$$\mathrm{LFPC}_{k'}[n] = 10\log_{10}\left|\frac{S_{k'}[n]}{N_{k'}}\right| \tag{2.1}$$

where $S_{k'}[n]$ is the energy of frequency band k' at frame n. Therefore all squared values of bins k within a frequency band k' are summed.

Harmonic coefficients (HC)

The HC is a feature used by Chou and Gu [25] and is defined as $\mathrm{HC} = \max_\tau(\mathrm{TA}[\tau] + \mathrm{SA}[k_\tau])$ with TA and SA denoting the temporal- and spectral autocorrelation functions respectively, where τ and k_τ are the corresponding time and frequency bin lags.

$$\mathrm{TA}[\tau] = \frac{\sum_{n=0}^{N-\tau-1}\bar{x}[n]\bar{x}[n+\tau]}{\sqrt{\sum_{n=0}^{N-\tau-1}\bar{x}^2[n]\sum_{n=0}^{N-\tau-1}\bar{x}^2[n+\tau]}} \tag{2.2}$$

$$\mathrm{SA}[\tau] = \frac{\sum_{k=0}^{M/2-k_\tau-1}\bar{X}[k]\bar{X}[k+k_\tau]}{\sqrt{\sum_{k=0}^{M/2-k_\tau-1}\bar{X}^2[k]\sum_{k=0}^{M/2-k_\tau-1}\bar{X}^2[k+k_\tau]}} \tag{2.3}$$

Signal frame $\bar{x}[n]$ with length N and magnitude spectrum $\bar{X}[k]$ generated by an M-point FFT are the zero-mean versions of $x[n]$ and $X[k]$ respectively.

Spectral flux

The spectral flux (SF) [26] is often used in the field of music instrument classification. The SF embodies the extent of spectral change between two consecutive frames at time index m and $m-1$ with energy normalized magnitude spectrum $\hat{X}$. It is computed as follows:

$$\mathrm{SF}_m = \sum_{k=0}^{M/2-1} \left(\hat{X}_m[k] - \hat{X}_{m-1}[k]\right)^2 \tag{2.4}$$

Spectral roll-off

The spectral roll-off R is a descriptor which indicates where the majority of energy is located in the spectrum. R is the greatest frequency value which fulfills the following inequation:

$$\sum_{k=0}^{R-1} X_m[k]^2 \leq \gamma \sum_{k=0}^{M/2-1} X_m[k]^2 \tag{2.5}$$

Rocamora and Herrera [21] used $\gamma = 0.85$.

Vibrato

The vibrato is a recurrent change of the pitch, a frequency modulation. It can be described by the two parameters rate and extent. The vibrato rate stands for the number of oscillations per second, whereas the extent includes the range of pitch change from the center frequency. For the vibrato description of signals , Khine et al. as well as Nwe and Li [5,27] use a cascaded two layer vibrato filterbank like in Fig. 2.2, where every center frequency is located at musical notes, i.e. filter bandwidth b is not constant, it is proportional to the center frequency. The first layer consists of 96 overlapping trapezoidal bandpass filters whose filter function is tapered between $\pm 0{,}5$ and $\pm 1{,}5$ semitones and is showed in (2.6). A filter bandwidth of 1,5 semitones is sufficient since the vibrato extent is normally less than one semitone [17]. All the filterbanks embrace 7 octaves from 65 to 16 kHz because singing has high frequency harmonics [5]. Each bandpass filter is followed by 5 non-overlapping rectangular filters with the same frequency extent and position as the first layer's filter.

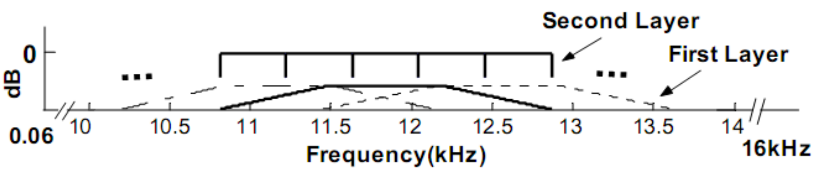

Figure 2.2: Vibrato filter [27]

$$W_{TRPZ}(f) = \begin{cases} 1,5 + 3\frac{f}{b} & \text{for } -\frac{b}{2} \leq f < -\frac{b}{6} \\ 1 & \text{for } -\frac{b}{6} \leq f \leq \frac{b}{6} \\ 1,5 - 3\frac{f}{b} & \text{for } \frac{b}{6} < f \leq \frac{b}{2} \end{cases} \tag{2.6}$$

After the octave scale filtering the logarithmic energy of each band is computed and after a DCT a octave frequency cepstral coefficient (OFCC) vector with 9 elements is obtained for each audio frame of 20 ms with 13 ms overlap. In order to capture the vibrato rate, the delta parameters of two consecutive frames are also considered.

2.1.2 Statistical features

Statistical features give a description of the spectrum's shape by interpreting it as a probability distribution. The individual frequencies and the normalized spectral amplitude correspond to the random variable and distribution respectively.

Spectral centroid

The spectral centroid (SC) [28] represents the center of gravity of a distribution and apparently, it correlates with the perceived *brightness* of a sound [2].

$$\text{SC} = \frac{\sum_{k=1}^{M/2} kX_m[k]}{\sum_{k=0}^{M/2-1} X_m[k]} \tag{2.7}$$

Skewness

The skewness is an indicator for the asymmetry of a distribution around the mean. If the values of the distribution are located more at upper frequencies than at lower frequencies, the skewness is negative and vice versa. The definition is:

$$\text{Skewness} = \frac{\sqrt{\frac{M}{2}} \sum_{k=0}^{M/2-1} \left(X_m[k] - \tilde{X}_m \right)^3}{\left(\sum_{k=0}^{M/2-1} \left(X_m[k] - \tilde{X}_m \right)^2 \right)^{3/2}} \tag{2.8}$$

where $\tilde{X}_m$ is the mean value of the magnitude spectrum $X_m[k]$ at frame m.

Kurtosis

The kurtosis shows how spired a distribution is. A pointed distribution has a high kurtosis value while a flatter one has a low value.

$$\text{Kurtosis} = \frac{\frac{M}{2} \sum_{k=0}^{M/2-1} \left(X_m[k] - \tilde{X}_m \right)^4}{\left(\sum_{k=0}^{M/2-1} \left(X_m[k] - \tilde{X}_m \right)^2 \right)^{2}} - 3 \tag{2.9}$$

Spectral Flatness

The spectral flatness (SF) [29] is a descriptor for the similarity of a distribution with a flat spectrum. It is the geometric mean divided by the arithmetic mean of the spectrum. A SF value close to 1 means high similarity to the ideal spectrum of white noise, and a low value stands for a more tonal signal. Rocamora and Herrera [21] compute SF values for four frequency bands in the area of 200 Hz and 5000 Hz.

$$\text{SF} = \frac{\sqrt[\frac{M}{2}]{\prod_{k=0}^{M/2-1} X_m[k]}}{\frac{1}{M/2} \sum_{k=0}^{M/2-1} X_m[k]} \tag{2.10}$$

2.2 Classification Techniques

In this section, a selection of wide-spread classification techniques is outlined.

2.2.1 Gaussian mixture model

A Gaussian mixture model (GMM) is the representation of data in feature space with D- dimensions by a linear combination of M Gaussian probability density functions (PDF), where each class is modeled by one GMM. It is a widespread approach used for diverse classification tasks: [30] use it for detection of singing voice segments, [31] for onset detection, [4,32,33] for singer identification, [34] for pitch estimation, [35] for singing voice separation, [25] for singing voice separation and [36] for melody transcription. According to Reynolds and Rose [37], the definition of a GMM is:

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{M} w_i g_i(\mathbf{x}) \tag{2.11}$$

where $\mathbf{x}$ is a D-dimensional feature vector, w_i with $i = 1, \ldots, M$ are the weights of individual D-variate Gaussian PDFs $g_i(\mathbf{x})$:

$$g(\mathbf{x}) = \frac{1}{(2\pi)^{D/2} |\mathbf{\Sigma}|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x} - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mu) \right\} \tag{2.12}$$

with mean vector μ and determinant of D-by-D covariance matrix $|\mathbf{\Sigma}|$. T is the matrix transpose operator. Consequently, a GMM k is entirely represented by it's parameter set $\lambda_k = \{w_i, \mu_i, \mathbf{\Sigma}_i\}$.

2.2.2 Maximum likelihood parameter estimation

For each class c, the model is trained by a sequence of N feature vectors $\mathbf{X} = \{\mathbf{x}[1], \mathbf{x}[2], \ldots, \mathbf{x}[N]\}$. The aim is to find parameters which best explain the set of feature vectors. This process is called maximum likelihood parameter estimation. A popular technique is the expectation-maximization (EM) algorithm [38] which, after starting with a initial model λ tries to reestimate the model parameters $\overline{\lambda}$ iteratively, such that $p(\mathbf{X}|\overline{\lambda}) \geq p(\mathbf{X}|\lambda)$. Reynolds [37] uses the following reestimation formulas:

$$\overline{w_i} = \frac{1}{N}\sum_{n=1}^{N} p(i|\,\mathbf{x}[n], \lambda) \tag{2.13}$$

$$\overline{\mu_i} = \frac{\sum_{n=1}^{N} p(i|\,\mathbf{x}[n], \lambda)\mathbf{x}[n]}{\sum_{n=1}^{N} p(i|\,\mathbf{x}[n], \lambda)} \tag{2.14}$$

$$\overline{\sigma}_i^2 = \frac{\sum_{n=1}^{N} p(i|\,\mathbf{x}[n], \lambda)\mathbf{x}[n]^2}{\sum_{n=1}^{N} p(i|\,\mathbf{x}[n], \lambda)} - \overline{\mu}_i^2 \tag{2.15}$$

Equation (2.13) computes the mixture weights, (2.14) the means and (2.15) the variances. σ_i^2, $\mathbf{x}[n]$ and μ_i are the elements of vectors σ_i^2, $\mathbf{x}[n]$ and μ_i. The a posteriori probability for component density i is given by

$$p(i|\,\mathbf{x}[n], \lambda) = \frac{w_i g_i(\mathbf{x}[n])}{\sum_{k=1}^{M} w_k g_k(\mathbf{x}[n])} \tag{2.16}$$

Problems of the training process are the choice of model order M and the initial model parameters, since the EM algorithm converges to the local maximum.

Decision function

After modeling each class c with an individual GMM, the class $\hat{c}$ has to be found, which produces the highest likelihood for a given observation sequence $\mathbf{X}$, i.e.

$$\hat{c} = \arg\max_k p(\lambda_k|\,\mathbf{X}) \tag{2.17}$$

In the case of singing voice detection, only two classes have to be distinguished. Therefore, a decision function can cope with the demand of equation (2.17). Tsai and Wang [39] use the following equation:

$$f_1(\mathbf{x}) = \log(p(\lambda_v|\,\mathbf{x})) - \log(p(\lambda_{nv}|\,\mathbf{x})) \tag{2.18}$$

where λ_v stands for the vocal and λ_{nv} for non-vocal model. If $f_1(\mathbf{x}) > 0$, the feature vector $\mathbf{x}$ belongs to the vocal class, otherwise it belongs to the non-vocal class. Lukashevich et al. [30] propose a function which can produce better decision results because of a more appropriate data spread, enabling a better class separation:

$$f_2(\mathbf{x}) = \frac{p(\lambda_v|\,\mathbf{x})}{p(\lambda_v|\,\mathbf{x}) + p(\lambda_{nv}|\,\mathbf{x})} - 0.5 \tag{2.19}$$

Additional post processing on $f_2(\mathbf{x})$ with autoregressive moving average filtering and smoothing with a Hamming window results in an average increase of correct classification from 72,7% to 81,3%.

2.2.3 Support vector machine

The support vector machine (SVM) is used amongst others in [19, 40, 41] for the task of singing voice detection. In [21], the SVM was compared against other classifiers like k-nearest neighbor and neural networks. SVM in conjunction with MFCC features exhibited the best performance.

Linear separable data

Given is a set of training data $X = \{\mathbf{x}[n], c[n]\}$ consisting of D-dimensional feature vectors $\mathbf{x}[n]$ and classes $c[n] \in \{+1, -1\}$ a decision function can be found which assigns incoming data to a certain class, i.e. $f_X : \Re^D \mapsto \{+1, -1\}$. It can be proven that, the probability of misclassification is higher for decision functions with high complexity [42]. The support vector machine (SVM) is based on the low-complex hyperplane separator:

$$\mathcal{H} = \{\mathbf{x}|\ \langle \mathbf{n}, \mathbf{x} \rangle + b = 0\} \tag{2.20}$$

where $\mathbf{n}, \mathbf{x} \in \Re^D, b \in \Re$. The decision function is $f_X = \operatorname{sgn}(\langle \mathbf{n}, \mathbf{x} \rangle + b)$. The optimal hyperplane to separate two classes has the maximum possible

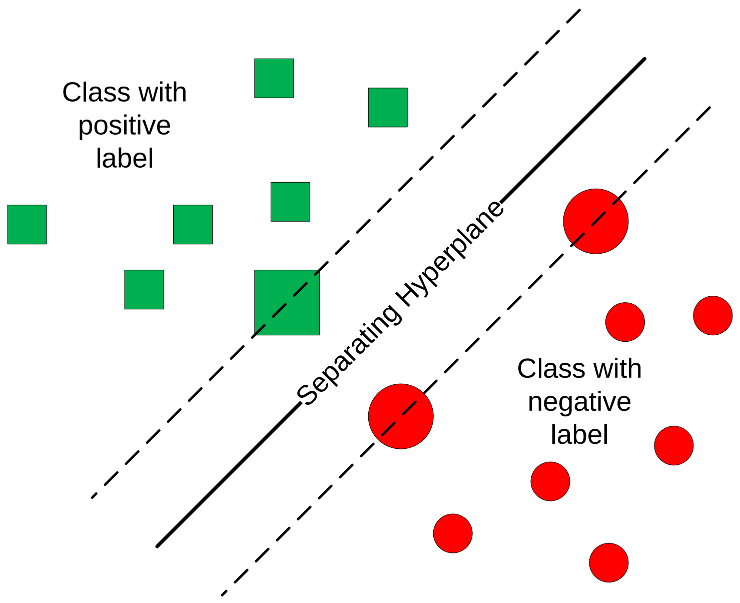

Figure 2.3: Hyperplane with maximal margin, separating two different classes. The members which adjoin the hyperplane margin are called support vectors (indicated by bold symbols).

distance between members of each class (see Fig. 2.3). The distance of a point $\mathbf{x}[n]$ belonging to class $c[n]$ from a hyperplane $\mathcal{H}$ is

$$d(\mathcal{H}, \mathbf{x}[n]) = c[n]\left(\left\langle \frac{\mathbf{n}}{||\mathbf{n}||}, \mathbf{x}[n] \right\rangle + \frac{b}{||\mathbf{n}||}\right) \tag{2.21}$$

The margin of hyperplane $\mathcal{H}$ is the distance from points $(\mathbf{x}[1], +1)$ and $(\mathbf{x}[2], -1)$ to $\mathcal{H}$. Combining equation (2.21) for these points yields

$$\left\langle \frac{\mathbf{n}}{||\mathbf{n}||}, (\mathbf{x}[1] - \mathbf{x}[2]) \right\rangle = \frac{2}{||\mathbf{n}||} \tag{2.22}$$

Optimization and Lagrangian function

Equation (2.22) is the key to position a maximum margin hyperplane $\mathcal{H}$ between points of two classes. The margin can be maximized, when $||\mathbf{n}||$ is minimized, because distance d is inversely proportional to $||\mathbf{n}||$. This is synonymous with $\min \langle \mathbf{n}, \mathbf{n} \rangle$ and to make sure that hyperplane $\mathcal{H}$ correctly separates the data it must be guaranteed that $c[n](\langle \mathbf{n}, \mathbf{x}[n] \rangle + b) \geq 1$. This optimization problem can be solved by the help of the Lagrangian function:

$$L(\mathbf{n}, b, \alpha) = \frac{1}{2} \langle \mathbf{n}, \mathbf{n} \rangle - \sum_n \alpha[n] \left[c[n](\langle \mathbf{n}, \mathbf{x}[n] \rangle + b) - 1 \right] \tag{2.23}$$

where α is the Lagrange multiplier and has to fulfill $\alpha[n] \geq 0$. The optimal conditions are set to

$$\frac{\partial L}{\partial \mathbf{n}} = 0 \quad \Rightarrow \quad \mathbf{n} = \sum_n c[n]\alpha[n]\mathbf{x}[n] \tag{2.24}$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad \sum_n c[n]\alpha[n] = 0 \tag{2.25}$$

One can get the dual Lagrangian function W by inserting the solutions of Eq. (2.24) in Eq. (2.23).

$$W(\alpha) = \sum_n \alpha[n] - \frac{1}{2} \sum_{n,m} \alpha[n]\alpha[m]c[n]c[m] \langle \mathbf{x}[n], \mathbf{x}[m] \rangle \tag{2.26}$$

The dual problem can be solved finding the maximum of W subject to the constraints $\alpha[n] \geq 0$ and $\sum_n \alpha[n]c[n] = 0$. The solution are values for the Lagrangian multiplier $\alpha[n]$ which are predominantly zero. The only nonzero values represent points which lie very close to the hyperplane $\mathcal{H}$. These points are called support vectors. That means the positioning of the maximal margin hyperplane only depends on the support vectors. The normal vector can now be computed with $\mathbf{n} = \sum_n c[n]\alpha[n]\mathbf{x}[n]$.

Decision function

The final decision function which assigns new unclassified feature vectors $\mathbf{x}[n]$ to a class $c[n]$ can be written as:

$$f_X = \operatorname{sgn}(\langle \mathbf{n}, \mathbf{x}[n] \rangle + b) = \operatorname{sgn}(\sum_n \alpha[n]c[n] \langle \mathbf{x}[n], \mathbf{x}[n] \rangle + b) \tag{2.27}$$

Mapping with kernel functions

The basic idea is to map data which is not easily separable by a hyperplane into a space representation, where the problem is easy to solve. A nonlinear map Φ transforms data into a higher dimensional dot product space (it is also called feature space, which is misleading here): $\Phi : \Re^D \mapsto \mathcal{F}$. Eq. (2.28) shows an example for a nonlinear map augmenting the dimension:

$$\Phi \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2} x_1 x_2 \end{pmatrix} \tag{2.28}$$

In $\mathcal{F}$ the data can be easily separated using the low complex hyperplane maximal margin classifier as described earlier. It can be seen in Eq. (2.26) and (2.27) that it's crucial property is that $\mathbf{x}_i$ only occurs in dot products. Computing dot products in high dimensional feature space $\mathcal{F}$ can be very expensive. Therefore suitable kernel functions $\mathcal{K}$ with inherent dot product properties are utilized. $\mathcal{K}$ is a function which exists in $\Re^D$ but behaves like a dot product in higher dimensional space $\mathcal{F}$. Mapping the data with function Φ and computing the dot product in $\mathcal{F}$ is all represented by the kernel functions. Eq. (2.29) elucidates this duality.

$$\langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle = \left\langle \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2} x_1 x_2 \end{pmatrix}, \begin{pmatrix} z_1^2 \\ z_2^2 \\ \sqrt{2} z_1 z_2 \end{pmatrix} \right\rangle \tag{2.29a}$$

$$= x_1^2 z_1^2 + x_2^2 z_2^2 + 2 x_1 z_1 x_2 z_2 = (x_1 z_1 + x_2 z_2)^2 \tag{2.29b}$$

$$= \langle \mathbf{x}, \mathbf{z} \rangle^2 =: \mathcal{K}(\mathbf{x}, \mathbf{z}) \tag{2.29c}$$

If a function is a kernel function can be ascertained by Mercer's theorem. The function $\mathcal{K}$ has to be symmetric: $\mathcal{K}(x_1, x_2) = \mathcal{K}(x_2, x_1)$, and kernel matrix

$K_{ij} = \mathcal{K}(x_i, x_j)$ has to be positive definite. Examples for kernels are:

$$\mathcal{K}(x_1, x_2) = \langle x_1, x_2 \rangle \qquad \text{(Linear)} \tag{2.30a}$$

$$\mathcal{K}(x_1, x_2) = (\gamma \langle x_1, x_2 \rangle + \beta)^d \qquad \text{(Polynomial)} \tag{2.30b}$$

$$\mathcal{K}(x_1, x_2) = \exp\{-\gamma ||x_1 - x_2||^2\} \qquad \text{(Gaussian)} \tag{2.30c}$$

2.2.4 Multilayer perceptron

The multilayer perceptron (MLP) is a subclass of artificial neural networks (ANN), which are inspired by the central nervous system in the brain. The classifier is applied amongst others in [43] for singing voice detection.

Besides the input (stimulus) and the output (response) of the network it consists of one or more hidden layers of neurons. The connections of neurons in the network are implemented by synapses. The neuron output only fires when the sum of weighted input signals exceed a certain threshold. This behavior is modeled by integrator and a nonlinear differentiable activation function, mainly a logistic function:

$$\varphi_i(v_j[n]) = \frac{1}{1 + e^{-v_j}} \tag{2.31}$$

where v_j is the weighted sum of all inputs of a neuron and the bias. The classification of an object represented by its features can be done by training the MLP. That means all the inner weights $\mathbf{W}$ of the network have to be adjusted with the help of hand-labeled references, indicating the class membership of every feature vector.

The induced local field is:

$$v_j[n] = \sum_{i=0}^{m} w_{ji}[n] y_i[n] \tag{2.32}$$

where $w_{j0}[n]$ is the bias $b_j[n]$ applied to neuron j. The output $y_j[n]$ of neuron

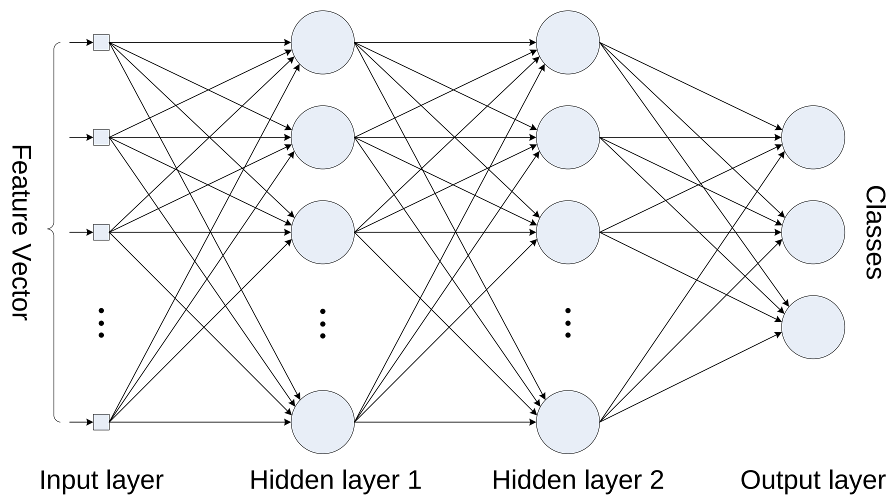

Figure 2.4: Layout of Multi layer perceptron. Adapted from [44]

j is:

$$y_j[n] = \varphi_j(v_j[n]) \tag{2.33}$$

The delta rule

The error $e_j[n]$ of neuron j at iteration n is the difference of the desired output, i.e. the class $c_j[n]$ and the neuron's actual output $y_j[n]$:

$$e_j[n] = c_j[n] - y_j[n] \tag{2.34}$$

The instantaneous error energy for neuron j is defined as $\frac{1}{2}e_j^2[n]$. Consequently, the total error energy E over all neurons in the output layer can be written as

$$E[n] = \frac{1}{2}\sum_{j\in C} e_j^2[n] \tag{2.35}$$

C is the set of neurons in the output layer. The average squared error energy is then

$$E_{av}[n] = \frac{1}{N}\sum_{n=1}^{N} E[n] \tag{2.36}$$

with the n-th training pattern and training pattern set size N. E_{av} acts as a representation for the learning performance and can be affected by the change of the weighting parameters. E_{av} is called cost function and in the following back-propagation algorithm it is the key element for gradient descent, i.e. trying to minimize the cost function in weight space by using the steepest path. For this, a correction $\Delta w_{ji}[n]$ has to be applied to the weighting parameters which can be written according to Haykin [44] as:

$$\Delta w_{ji}[n] = -\eta \frac{\partial E[n]}{\partial w_{ji}[n]} = \eta \delta_j[n] y_i[n] \tag{2.37}$$

where η is the learning-rate parameter, the step size of the descent. Eq. (2.37) is called delta rule where $\delta_j[n]$ is the local gradient of neuron j:

$$\delta_i[n] = e_j[n] \varphi_j'(v_j[n]) \tag{2.38}$$

The dash on the right side of φ denotes a differentiation with respect to the argument.

Back-propagation formula

The back-propagation algorithm contains two main parts, the forward and the backward pass. In the forward pass the response of the output layer to a certain input pattern is calculated. In the backward pass, error signals are propagated backwards through the network starting from the output layer. Local gradients $\delta_j[n]$ for each neuron j are recursively computed and therefore corrections $\Delta w_{ji}[n]$ can applied to the weights. This is the reason why a relation between adjacent layers is needed. There are two cases. For the first case, neuron j is a output node. $\delta_j[n]$ can be computed with Eq. (2.38). For the second case, the neuron is in one of the hidden layers. The back-propagation formula which determines the propagation of local gradients from one layer to another towards the input of the MLP is:

$$\delta_j[n] = \varphi_j'(v_j[n]) \sum_k \delta_k[n] w_{kj}[n] \tag{2.39}$$

where k stands for neurons in a layer which is behind the layer of neurons j regarding the forward pass orientation.

The back-propagation algorithm

The back-propagation algorithm has to modes: in the batch mode, coefficient updates are executed after all training vector of an epoch are presented. The second mode is the online mode where the coefficients are adjusted after each training vector. One epoch of a training vectors is defined as $\{\mathbf{x}[n], c[n]\}_{n=1}^{N}$ where $\mathbf{x}[n]$ are the features from a signal, $c[n]$ are the corresponding classes like singing and non-singing voice and N is the size of the epoch. The following back-propagation algorithm according to Haykin [44] works in online mode:

1. Initialization. Initialize weights and thresholds with values from a uniform distribution with zero mean.

2. Presentation of training examples. Present the next feature vector $\mathbf{x}[n]$ at time instance n which is a part of feature set $\mathbf{X}$ to the input of the network.

3. Forward computation. Compute the response to training vector by preceding layer by layer towards the output, i.e. compute the local induced fields $v_j^{(l)}$ for the neurons j of each layer l and their corresponding outputs y_i^l. Determine the error signals $e_j[n]$.

4. Backward computation. Compute the local gradients $\delta_j^{(l)}$:

$$\delta_j^{(l)}[n] = \begin{cases} e_j^{(L)}[n]\varphi_j'(v_j^{(L)}[n]) & \text{for output layer L,} \\ \varphi_j'(v_j^{(l)}[n]) \sum_k \delta_k^{(l+1)}[n] w_{kj}^{(l+1)}[n] & \text{for hidden layer } l. \end{cases} \tag{2.40}$$

   In Eq. (2.40) it can be seen that the error $e_j^{(L)}[n]$ of a neuron j is only computed for the output layer with Eq. (2.34). For all other layers, the error propagates backwards via the local gradients. Update the weights $w_{ij}^{(l)}$ according to:

$$w_{ji}^{(l)}[n+1] = w_{ji}^{(l)}[n] + \alpha(w_{ji}^{(l)}[n-1]) + \eta\delta_j^{(l)}[n]y_i^{(l-1)}[n] \tag{2.41}$$

   Eq. (2.41) is a generalized delta rule version of Eq. (2.37). The second term with momentum constant α, which controls the influence of preceding weights is added.

5. Iteration. Go back to step 2 until the stopping criterion is met.

2.3 Feature Selection

A selection of features is fundamental in order to reduce the classification-error. As a consequence of a smaller set of features after the selection process, memory requirements and computational costs decrease and a smaller training data set is required [45].

Regarding the evaluation of feature subsets, the literature discriminates basically between two approaches:

- Filter method: The evaluation of the discriminational power of a single or a set of features happens independently of the classifier. An evaluation criterion is applied to feature data and occasionally also to class data.

- Wrapper method: The classifier is used for the evaluation feature subsets. A widespread method for the evaluation of the predictive power of a feature set is to perform cross-validation [46] with an error measure.

In sections 2.3.1 and 2.3.2, two evaluation criteria for filter-based feature selection of single features or subsets with multiple features are discussed. The Fisher criterion and LDA were used for experiments within the scope of this thesis.

An important problem of feature selection is how to search relevant combinations of features for forming a feature subset, because the evaluation of all possible feature combinations can be very costly and time-consuming for a large number of features. Two established search strategies are for instance forward selection and backward elimination. The Forward selection strategy begins with one feature and successively adds the feature, which generates the highest gain compared to the feature subset before. The backward elimination strategy starts with all available features and successively removes the feature which produces the highest accuracy gain. Both algorithms stop the search, when the modification of the featureset does not induce an improvement of accuracy anymore [45].

2.3.1 Fisher's Linear Discriminant Analysis

The aim of Fisher's linear discriminant analysis (LDA) for a two class scenario is to find a hyperplane $\mathcal{H}$, which separates the feature data $\mathbf{x}$ belonging to each of these classes according to the Fisher criterion [47]. The hyperplane with normal vector $\mathbf{n}$ is given as:

$$\mathcal{H} = \{\mathbf{x} | \ \langle \mathbf{n}, \mathbf{x} \rangle + b = 0\} \tag{2.42}$$

The basic idea of Fisher's criterion [48] is, that feature data is assumed to be well-separable, if the distance between the mean vectors of each class is large, while the scattering of each class is small. The Fisher criterion for multidimensional feature space is defined as:

$$J(\mathbf{n}) = \frac{\mathbf{n}^T \mathbf{S}_B \mathbf{n}}{\mathbf{n}^T \mathbf{S}_W \mathbf{n}} \tag{2.43}$$

$\mathbf{S}_B$ represents the between-class scatter matrix. This matrix contains values, which stand for the distance between the mean vectors for all feature combinations.

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T \tag{2.44}$$

$\mathbf{S}_W$ stands for the within-class scatter matrix and contains the sum of variances of each class for all feature combinations.

$$\mathbf{S}_W = \sum_{n \in c_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in c_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T \tag{2.45}$$

The mean vector of each class c (for a two-class problem k is 1 or 2) over all time instances n is given as:

$$\mathbf{m}_k = \frac{1}{N_k} \sum_{n \in c_k} \mathbf{x}_n \tag{2.46}$$

and the projection onto hyperplane $\mathcal{H}$ results in:

$$m_k = \mathbf{n}^T \mathbf{m}_k \tag{2.47}$$

Equation 2.43 is differentiated with respect to $\mathbf{n}$ in order to determine the maximum possible value of $J(\mathbf{n})$ for a evaluated set of features. The maximum can be reached by variation of the direction of the hyperplane's normal vector $\mathbf{n}$, when the following equation is valid:

$$(\mathbf{n}^T \mathbf{S}_B \mathbf{n}) \mathbf{S}_W \mathbf{n} = (\mathbf{n}^T \mathbf{S}_W \mathbf{n}) \mathbf{S}_B \mathbf{n} \tag{2.48}$$

The scalars $\mathbf{n}^T\mathbf{S}_B\mathbf{n}$ and $\mathbf{n}^T\mathbf{S}_W\mathbf{n}$ can be disregarded, because only the direction of the normal vector $\mathbf{n}$ is relevant and not it's length. Multiplication with $\mathbf{S}_W^{-1}$ yields:

$$\mathbf{n} = \mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{n} \tag{2.49}$$

Equation 2.44 shows that $\mathbf{S}_B\mathbf{n}$ points to the direction of $\mathbf{m}_2 - \mathbf{m}_1$. The normal vector $\mathbf{n}$ of hyperplane $\mathcal{H}$ for a two-class problem is therefore given as:

$$\mathbf{n} = \mathbf{S}_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1) \tag{2.50}$$

When Eq. (2.50) is applied to Eq. (2.43), the maximal value of the Fisher criterion for a selected feature or feature subset can be calculated.

2.3.2 Correlation-based Feature Selection

The Pearson coefficient [49] can be used as criterion for correlation-based feature selection (CFS) [50] and is defined as:

$$\varrho(X, C) = \frac{\text{cov}\{X, C\}}{\sqrt{\text{var}\{X\}\text{var}\{C\}}} \tag{2.51}$$

where cov and var denote the covariance and variance respectively. X and C are random variables representing feature and class.

An estimation for Eq. (2.51) is:

$$\hat{\varrho}(X, C) = \frac{\sum_m(\mathbf{x}_i[m] - \bar{\mathbf{x}}_i)(c[m] - \bar{c})}{\sqrt{\sum_m(\mathbf{x}_i[m] - \bar{\mathbf{x}}_i)^2 \sum_m(c[m] - \bar{c})^2}} \tag{2.52}$$

where m is the frame-index of the STFT. The bar notation stands for the mean value over time index m.

This correlation measure can only evaluate a single feature for feature ranking of single features. In [50], a correlation-based criterion was designed, which evaluates correlation between the selected feature subset containing

κ features and the class, denoted by $\bar{\varrho}(X_\kappa, C)$. $\bar{\varrho}(X_\kappa, X_\kappa)$ stands for the averaged correlation coefficient between the feature subsets. The criterion used for CFS as defined by Hall [50] is:

$$J(X_\kappa, C) = \frac{\kappa\, \bar{\varrho}(X_\kappa, C)}{\sqrt{\kappa + \kappa(\kappa - 1)\, \bar{\varrho}(X_\kappa, X_\kappa)}} \tag{2.53}$$

2.4 Evaluation of Classification-Performance

The following sections comprise approaches which where applied in the scope of this thesis in order to evaluate the performance of the pattern recognition system.

2.4.1 K-fold Cross-Validation

K-fold cross-validation [46] is an acknowledged approach for the evaluation of the classifier's model to predict the class-membership of unknown frames. For this purpose, the dataset is divided into k subsets of equal size. In every run, another subset is used for testing, while the remaining $k - 1$ subsets are used for the training of the classifier's model. Finally, the mean of k test results can be calculated. The evaluation of the k test subsets is accomplished by an error measure like for instance frame accuracy. This measure is the ratio of correct classified frames to the number of all frames.

2.4.2 Confusion Matrix

The confusion matrix (see table 2.1) delivers insight into the errors related to each class. The row index of the matrix stands for the actual classes P (Positive class, i.e. singing voice is present) and N (Negative class, i.e. singing voice is not present). The column index represents the predicted

		Predicted classes	
		$\hat{P}$	$\hat{N}$
Actual classes	P	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

Table 2.1: Confusion matrix

classes $\hat{P}$ and $\hat{N}$. Consequently, it can be evaluated whether a classifier tends to deliver results, which are biased towards one class.

Chapter 3

The Implemented System

This chapter describes the implemented pattern recognition system for singing voice detection. The implementation was completely programmed in MATLAB except the classifier. A support vector machine classifier called LIBSVM by Chang and Lin [51] was used. An system overview can bee seen in figure 3.1. Basically, the system comprises three stages:

- **Validation stage:** the aim of this stage is to find a predictive model by selecting adequate features (see chapter 3.4) and finding optimal SVM-parameters (see chapter 3.3).

- **Training stage:** the calculated parameters and features from the validation stage are used to train the classifier with files from the training data base. In both, the validation and the training stage, hand-labeled class data for the training of the model is used. This used learning strategy is called supervised learning (see chapter 2). In both stages, not all available frames are used for building the predictive model. The frames are randomly selected from the database. Ramona states in [19], that an increase of training data frames does not have an noticeable

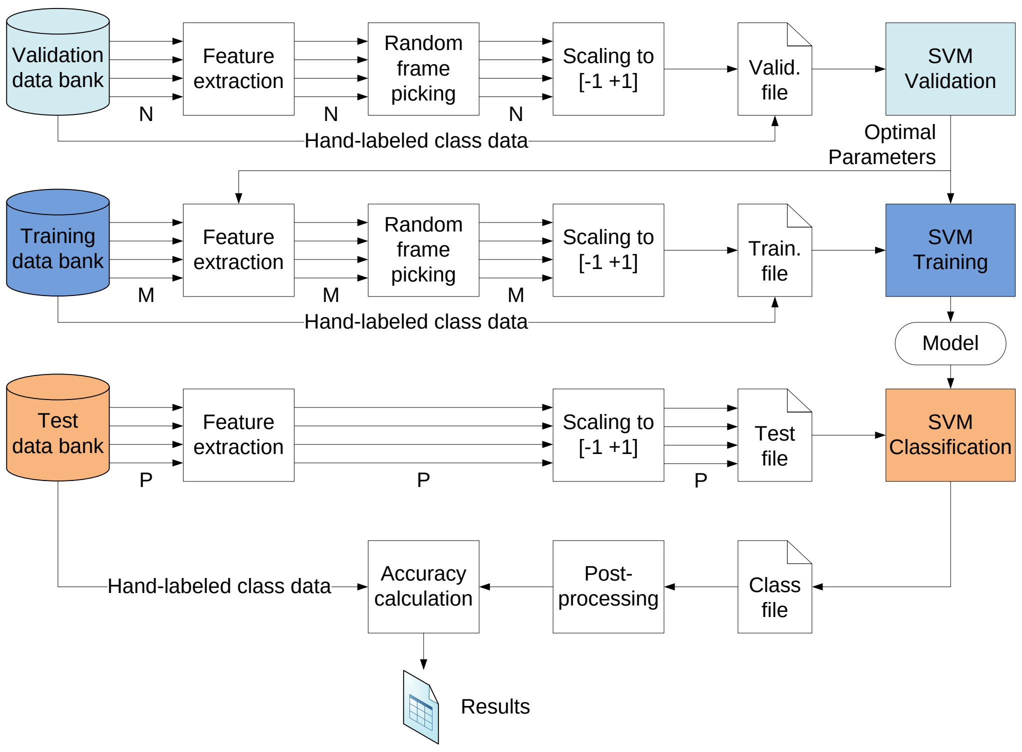

Figure 3.1: Main flow

impact on the classification result. This statement was verified by own empirical experiments. For computational reasons, like in [19], 20000 frames were randomly extracted from the database for training.

- **Test stage:** The predictive model, calculated in the training stage is used for the prediction of singing voice segments of unknown test items.

In all stages, the range of the feature values is linearly scaled to the interval $[-1, +1]$ [52].

Database	**Items**	**Length**
Validation	15	≈ 1 h
Training	52	≈ 3.3h
Test	14	≈ 1 h

Table 3.1: Databases

3.1 Database

The audio files used in the scope of this thesis were assembled by Mathieu Ramona [19]. He provided links to 93 files from www.jamendo.com as well as manual annotations, which give information about the presence of singing voice in every file (these annotations act as a reference). He divided the files into three non-overlapping subsets for validation (16 files), training (61 files) and testing (16 files). All kinds of parameter adjustments are solely performed on the validation and training database in order to avoid possible over-fitting to the test data. Unfortunately some links were broken, so the databases used for this thesis comprise less items (see table 3.1). The database contains more than 5 hours of music.

3.2 Feature Extraction

3.2.1 MFCC-based Features

The extracted feature set consists of 79 features (see table 3.2), from which 78 features are based on MFCC-features. In [21], a feature set based on MFCCs achieved the best result of all tested feature-sets with an accuracy of 84.5 %.

In the scope of this thesis, 13 MFCCs and their delta features denoted

Feature	**Coefficients**
MFCC	13
MFCCmed	13
MFCCstd	13
dMFCC	13
dMFCCmed	13
dMFCCstd	13
EBR	1

Table 3.2: All extracted features

by *dMFCC* were extracted. Delta features are computed as the difference between the feature values of two consecutive frames and thus represent the first derivative.

Additionally, long-term features calculating the median and standard deviation over a time window are utilized. In [21] different window lengths and hopsizes were tested. A window length of 1 second achieved the best performance and is consequently used for extraction here. These long-term features applied to MFCCs and dMFCCs are denoted by *MFCCmed*, *dMFCCmed* and *MFCCstd*, *dMFCCstd* respectively.

The features were extracted with a sampling rate of 11025 Hz. The STFT was computed using a Hamming window of approximate 23 ms length. Every 10 ms, a new spectrum is available.

3.2.2 Energy Band Ratio

The Energy Band Ratio (EBR) is the energy of the frequency band between 200 and 2000 Hz related to the first MFCC coefficient. The first coefficient represents the DC-component of the envelope of a STFT-spectrum and is

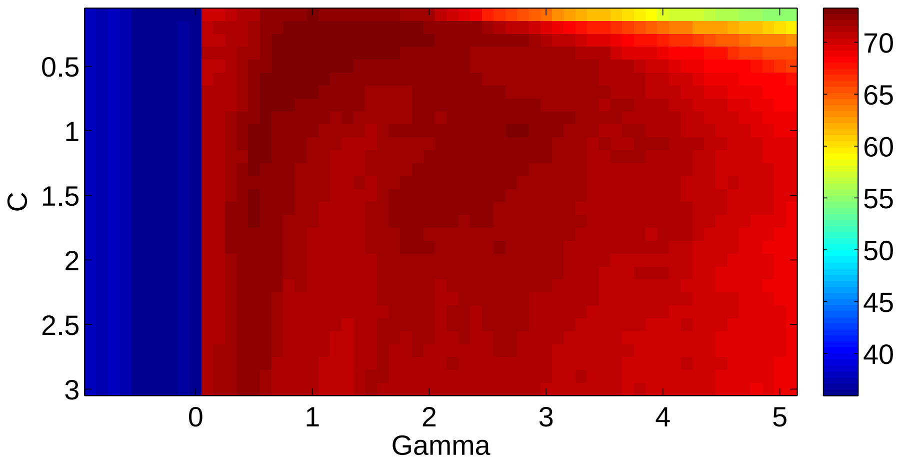

Figure 3.2: Evaluation of SVM-parameters C and gamma by mean accuracy of 10-fold cross-validation in percent.

related to the RMS-value of the spectrum. The selected frequency band is an area where the main part of the energy singing voice is located. For unvoiced speech signals on the other hand the main energy extends up to 4000 Hz [21].

3.3 Grid Search

The SVM classifier can be used with different parameters and kernels. The optimal settings depend on the application. Here, a RBF kernel was used and the optimal values for SVM-parameters C and γ were determined with the help of the 10-fold cross-validation method (see chapter 2.4) and the validation database. After searching a coarse parameter grid including values of $C = 2^{-5}, 2^{-3}, \ldots, 2^{15}$ and $\gamma = 2^{-15}, 2^{-13}, \ldots, 2^{3}$, a finer grid search was completed, containing promising parameter areas. The result of the fine grid search can be seen in figure 3.2. The best results were obtained by $C = 1$ and

$\gamma = 0.3$. It can be seen, depending on different parameter values, changes in mean accuracy ranging from about 40 % to over 70 % are possible.

3.4 Feature Selection

This chapter describes the process of feature subset selection from the 79 extracted features. First, a feature ranking was executed, to make a statement about the importance of single features, evaluated by the Fisher criterion. Subsequent steps include forward selection and backward elimination feature subset selection with Fisher criterion as evaluation measure. After the selection of potential feature subset candidates, the subsets were evaluated, comprising the SVM classifier accompanied by 10-fold cross-validation as evaluation measure.

3.4.1 Feature Ranking

The 79 extracted features were ranked by applying the Fisher criterion as described in chapter 2.3.1 on every single feature. Table 3.3 shows the results of the best ten features. For the calculation of the relative score, the Fisher criterion values of all features were normalized such, that the best feature exhibits a value of 1. The number in front of the feature name denotes the coefficient number.

According to the Fisher criterion ranking, EBR is the best feature. It is remarkable, that eight out of the ten best features include long-term features. The remaining two features correspond to the RMS-value of a STFT-spectrum.

Rank	Feature name	Rel. Score
1	EBR	1.00
2	4. MFCCstd	0.98
3	1. MFCCmed	0.73
4	1. MFCC	0.62
5	8. MFCCstd	0.46
6	7. MFCCstd	0.30
7	6. MFCCstd	0.28
8	9. MFCCstd	0.26
9	5. MFCCstd	0.24
10	3. MFCCstd	0.21

Table 3.3: Variable ranking results

3.4.2 Feature Subset Selection

Filter Method

The first step of the feature subset selection utilizes a filtering method, in order to identify potential feature subsets by the use of the Fisher criterion as performance criterion. Forward selection and backward elimination algorithms (see chapter 2.3) for the generation of the subsets were chosen.

Figure 3.3 shows the Fisher criterion results of both subset selection algorithms. It can be seen, that the Fisher values of both subset algorithms differ very slightly. For a feature subset size of above 40 features, no remarkable improvement of the Fisher criterion can be observed, whereas there is a steep ascent for small subset sizes up to 10 features.

The lower subfigure shows the number of different features which were chosen by forward and backward selection algorithms. It is noticeable, that the subsets of both algorithms differ in 60 % of the cases by only a single

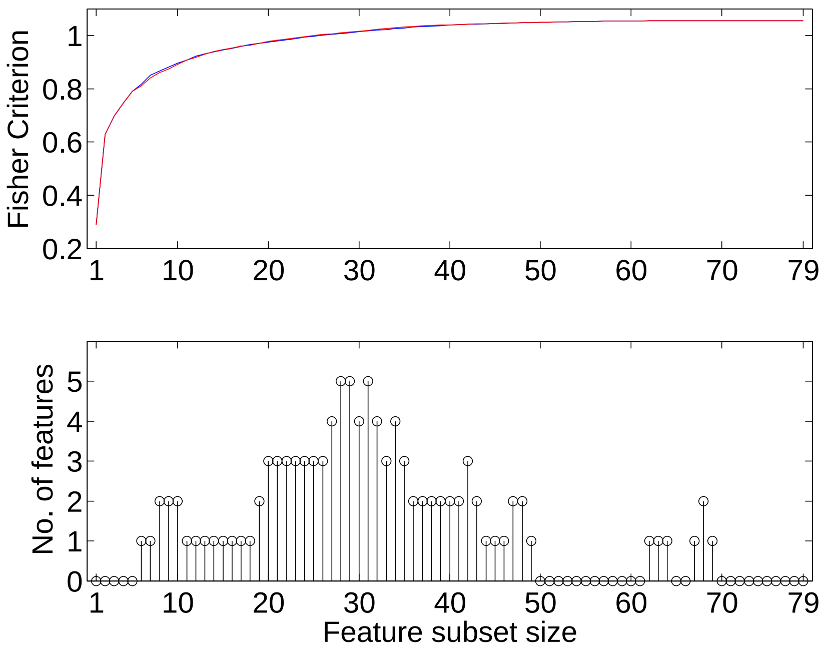

Figure 3.3: Upper subfigure: feature subset selection by forward selection (red graph) and backward elimination (blue graph). Lower subfigure: Number of features in which the subsets of forward selection and backward elimination differ.

feature or include exact the same features.

Wrapping

The feature subsets which were determined by the previous filtering method are evaluated with the help of the SVM classifier. This is achieved by the usage of a 10-fold cross-validation and the mean and standard deviation of accuracy over all cross-validation runs. The features subsets, both of forward selection and backward elimination, were tested up to a subset size of 40 features. It became apparent, that the evaluation of higher feature set sizes will not evoke improvement in accuracy.

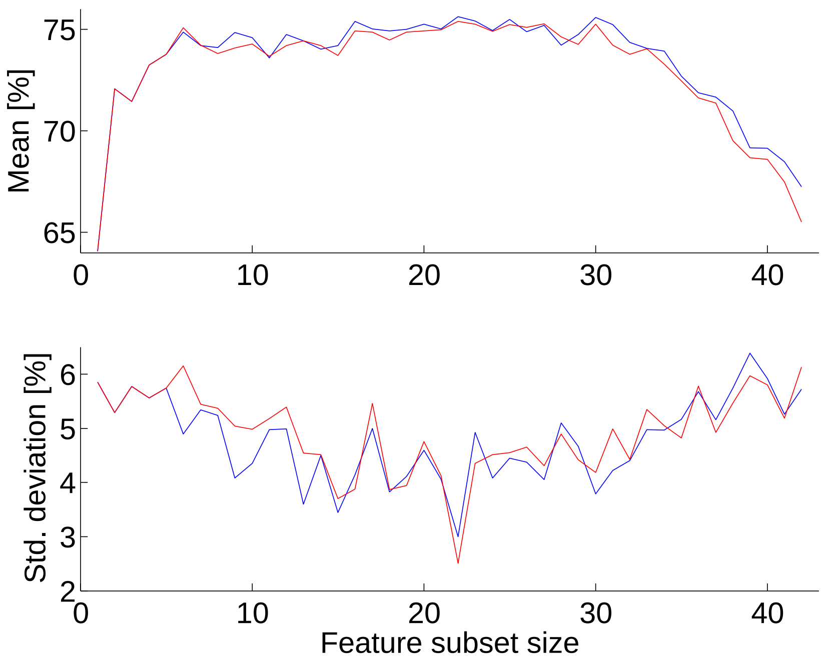

Figure 3.4: Mean and standard deviation over accuracy of 10-fold cross-validation runs. The blue and red graphs indicate the results of backward elimination and forward selection generated subsets respectively.

The results are depicted in figure 3.4. A steep ascent of mean accuracy can be observed for small subset sizes up to 6 features. Above a subset size of 6 features up to about 30 features, the mean accuracy remains in the area of 75 %. For these subset sizes, the standard deviation changes more explicitly and has its minimum at a subset size of 22. Therefore, this subset size generated by forward selection was chosen for the final test stage.

Figure 3.5 shows the confusion matrix values (see chapter 2.4) of forward selection generated subsets. Due to slight differences and clarity of depiction, only the results of forward selected subsets are shown. It is noticeable, that above a subset size of 30 features, mainly false positive values increase at the expense of a decrease of true negative values.

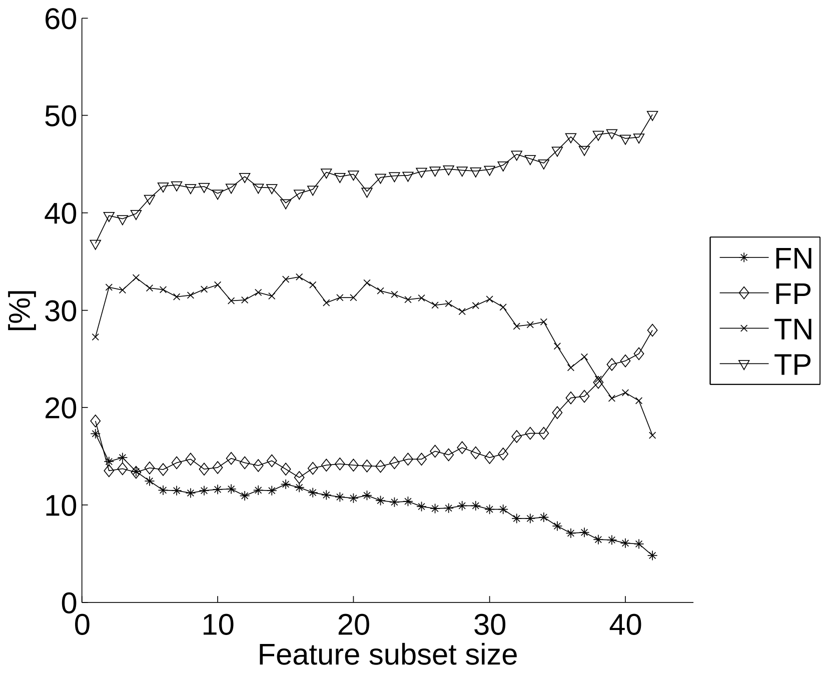

Figure 3.5: Confusion matrix results of forward selection subsets.

3.5 Post-Processing

As post-processing, simple median filtering is utilized within the system. Ramona [19] achieved a mean improvement of 8.9 % accuracy over all database items by means of median filtering with a filter size of approximately 0.5 seconds. Figure 3.6 displays the mean and standard deviation of the improvement of accuracy over all items of the validation database evoked by median filtering. The best mean improvement is about 1.7 % with a filter size of approximately 1.9 seconds. The increase of standard deviation with median filter size shows, that not all items benefit from this post-processing alike.

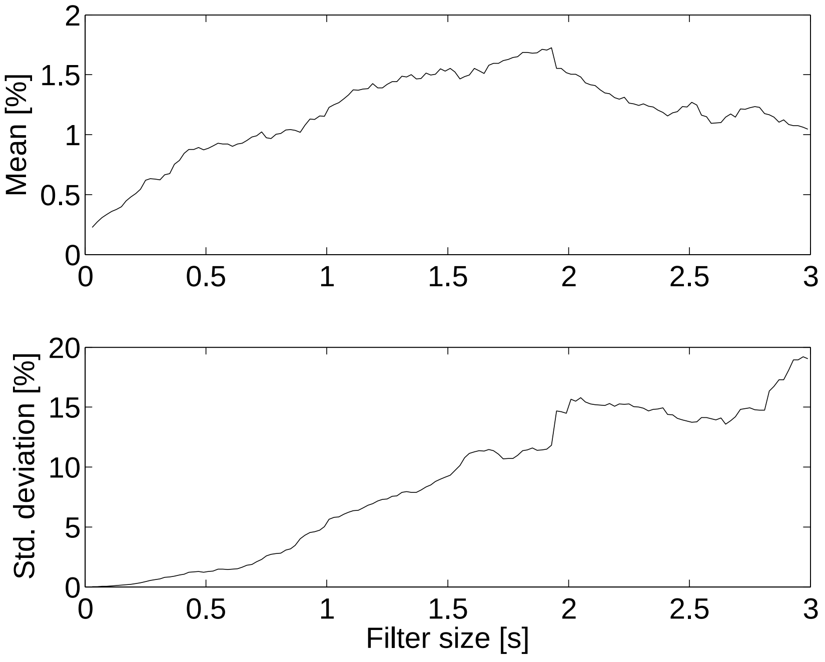

Figure 3.6: Mean and standard deviation of accuracy improvement over all validation database items caused by median filtering.

Chapter 4

Results

A SVM-model was trained, using the determined optimal kernel-parameters $C = 1$ and $\gamma = 0.3$ (see chapter 3.3) and the selected feature subset (see chapter 3.4).

The resulting system is applied to the same test database as used by Ramona [19] in order to compare the results. Unfortunately, because of broken links to audio files, especially the training database differs with respect to the number of items. Ramona used 61 items for training, whereas only 52 items were available for training here. Thus, the results are in fact not fully comparable. Also links to files of the test and validation databases used in the scope of this were broken, such that two and one items were missing respectively.

The results are presented in table 4.1. Besides the overall accuracy of each item, the table shows the accuracy with respect to each class. Furthermore, the impact of post-processing on the accuracy results is shown. The accuracy of both classes over all items is 68.1 % without, and 69.7 % with additional post-processing by median-filtering. The accuracy only for the detection of singing voice frames is 76.7 %, which is better than the accuracy of non singing voice frames with 63.7 % with post-processing. It is apparent, that the modeling of singing voice content works considerably better. By listening to the items, where non singing voice was poorly classified, it appeared that instruments had an apparent resemblance to voice-like sounds.

The performance of the implemented system is considerably inferior to the results of Ramona's singing voice detection system. Ramona's system obtains an accuracy for both classes over all items of 82.2 % with post-processing. His system performs almost equally well on both classes, but some items of the non singing voice class also reveal weak results.

Ramona states, that the results of the test set improved by about 9 % in accuracy by using simple median filtering with about 0.5 seconds filter length. Here, the post-processing improvement of the implemented system was not better than 1.6 %.

Class	**Singing Voice Accuracy [%]**		**No Singing Voice Accuracy [%]**		**Overall Accuracy [%]**	
Post-processing	–	Median	–	Median	–	Median
16 ans.wav	76.1	80.3	84.4	93.5	82.3	90.1
2003-Circons[...].wav	54.5	52.6	86.9	90.5	73.0	74.3
A Poings Fermes.wav	99.6	100	52.1	49.0	68.3	66.4
Believe.wav	74.6	80.6	85.4	87.0	78.4	82.9
Crepuscule.wav	96.2	98.0	33.7	30.3	65.6	64.9
Dance.wav	81.1	87.9	61.7	64.4	75.2	80.7
Elles disent.wav	71.0	72.5	72.9	80.2	71.6	75.1
Healing Luna.wav	93.8	96.3	43.2	39.0	64.4	63.0
Inside.wav	82.1	86.5	79.7	84.0	80.3	84.5
Say me Good Bye.wav	63.5	65.9	32.8	34.9	49.6	51.8
School.wav	43.5	39.1	96.4	97.4	67.0	65.0
Si Dieu.wav	86.6	95.4	09.1	05.4	43.6	45.4
Une charogne.wav	82.3	86.4	44.9	42.1	73.4	75.8
You are.wav	39.7	31.9	94.5	94.8	61.1	56.5
All items	**74.6**	**76.7**	**62.7**	**63.7**	**68.1**	**69.7**

Table 4.1: Accuracy results of test database.

Chapter 5

Conclusion

A pattern recognition framework for singing voice detection has been implemented and programmed in MATLAB. A database, assembled by Mathieu Ramona [19] was used for validation, training and testing.

A feature set consisting of 79 features, mainly based on MFCCs, including long-term and delta MFCC features have been extracted. Furthermore, an energy-based feature was proposed in the scope of this thesis.

Single features have been ranked with the help of the Fisher criterion. The proposed energy feature emerged as the best feature. It was noticeable, that 8 out of the best 10 features were long-term MFCCs.

Feature subset selection has been accomplished by means of LDA and Fisher criterion in combination with sequential forward selection and backward elimination. Subsequently, the potential LDA-generated feature subsets have been evaluated in conjunction with the SVM classifier and 10-fold cross-validation on the training database. A feature set consisting of 22 features obtained a mean accuracy of 75.6 % with a standard deviation of 2.5 % over all cross-validation runs.

Finally, the system has been applied to the test database of Ramona. The proposed system obtained an overall accuracy of 69.7 %, whereas Ramona's system obtained 82,2 %. However, it has to be taken into account that the training database used here consisted of only 52 instead of 61 items, because some items were unfortunately not available anymore. The test database used here also had only 14 items instead of 16.

Chapter 6

Outlook

The test results of the implemented system show, that especially the accuracy of the non singing voice class can be improved. More features, particularly with the capability to model music content could be incorporated in the feature set.

Also the influence of features which quantify panning or coherence information between stereo channels could be studied. In contrast to music, singing voice is generally panned to the middle of the stereo panorama, whereas music exhibits a wider stereo panorama. It could be investigated, if this attributes enhance the ability of the system to discriminate between music and singing voice.

Furthermore, the impact of the missing training database items on the accuracy of the test database could be investigated.

The system at the current status utilizes only simple median filtering. More sophisticated post-processing demonstrably improves the overall performance of the system.

Bibliography

[1] M. Bartsch and G. Wakefield, “Singing voice identification using spectral envelope estimation,” *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 2, pp. 100–109, March 2004.

[2] M. A. Bartsch, “Automatic singer identification in polyphonic music,” Ph.D. dissertation, University of Michigan, MI, USA, August 2004.

[3] Y. E. Kim, “Singer identification in popular music recordings using voice coding features,” in *Proceedings of the 3rd International Conference on Music Information Retrieval*, 2002, pp. 164–169.

[4] N. C. Maddage, C. Xu, and Y. Wang, “Singer identification based on vocal and instrumental models,” *International Conference on Pattern Recognition*, vol. 2, pp. 375–378, 2004.

[5] T. L. Nwe and H. Li, “Exploring vibrato-motivated acoustic features for singer identification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 2, pp. 519–530, 2007.

[6] C. Cao, M. Li, J. Liu, and Y. Yan, “Singing melody extraction in polyphonic music by harmonic tracking,” in *ISMIR 2007, 23-30 September 2007, Vienna (Austria).*

[7] A. Chanrungutai and C. A. Ratanamahatana, “Singing voice separation for mono-channel music using non-negative matrix factorization,” *International Conference on Advanced Technologies for Communications*, pp. 243–246, Oct. 2008.

[8] Y. Zhu and S. Gao, “Extracting vocal melody from karaoke music audio,” *IEEE International Conference on Multimedia and Expo*, July 2005.

[9] C. Cao, M. Li, J. Liu, and Y. Yan, “A study on singing performance evaluation criteria for untrained singers,” *9th International Conference on Signal Processing*, pp. 1475–1478, Oct. 2008.

[10] B. Kostek and P. Zwan, “Automatic classification of singing voice quality,” *5th International Conference on Intelligent Systems Design and Applications*, pp. 444–449, Sept. 2005.

[11] L. P. Clarisse, J. Martens, M. Lesaffre, B. D. Baets, H. D. Meyer, and M. Leman, “An auditory model based transcriber of singing sequences,” in *ISMIR, 13-17 October 2002, Paris (France).*

[12] M. Ryynaenen, “Singing transcription,” in *Signal Processing Methods for Music Transcription*, A. Klapuri and M. Davy, Eds. New York: Springer, 2006, no. ISBN 0-387-30667-6, ch. 12, pp. 361–390.

[13] A. Duda, A. Nurnberger, and S. Stober, “Towards querq by singing/humming on audio databases,” in *ISMIR, 23-30 September 2007, Vienna (Austria).*

[14] T. Hosoya, M. Suzuki, A. Ito, and S. Makino, “Lyrics recognition from a singing voice based on finite state automaton for music information retrieval,” in *ISMIR, 11-15 September 2005, London (UK).*

[15] W.-H. Tsai and H.-M. Wang, "Towards automatic identification of singing language in popular music recordings," in *ISMIR, 10-15 October 2004, Barcelona (Spain).*

[16] P. R. Cook, "Identification of control parameters in an articulatory vocal tract model, with applications to the synthesis of singing," Ph.D. dissertation, Stanford, CA, USA, 1991.

[17] J. Sundberg, *The science of the singing voice.* Northern Illinois University Press, Illinois, 1987.

[18] Y. E. Kim, "Singing voice analysis, synthesis, and modeling," in *Handbook of Signal Processing in Acoustics*, D. Havelock, S. Kuwano, and M. Vorlnder, Eds. Springer New York, 2009, pp. 359–374.

[19] M. Ramona, G. Richard, and B. David, "Vocal detection in music with support vector machines," 2008, rTL (Ediradio).

[20] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time signal processing (2nd ed.).* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999.

[21] M. Rocamora and P. Herrera, "Comparing audio descriptors for singing voice detection in music audio files," 2007, music Technology Group Universitat Pompeu Fabra.

[22] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357 – 366, aug. 1980.

[23] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *In International Symposium on Music Information Retrieval*, 2000.

[24] T. L. Nwe, A. Shenoy, and Y. Wang, "Singing voice detection in popular music," in *Proceedings of the 12th annual ACM international conference on Multimedia*, 2004, pp. 324–327.

[25] W. Chou and L. Gu, "Robust singing detection in speech/music discriminator design," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, vol. 2, 2001, pp. 865–868.

[26] P. Masri, "Computer modelling of sound for transformation and synthesis of musical signals," Ph.D. dissertation, University of Bristol, 1996.

[27] S. Khine, T. L. Nwe, and H. Li, "Singing voice detection in pop songs using co-training algorithm," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1629–1632, April 2008.

[28] *MPEG-7, information technology - multimedia content description interface - part 4: Audio*, Moving Pictures Expert Group ISO/IEC JTC1/SC29/WG11 Int. Standard 15 938-4, Rev. Final Draft.

[29] J. Gray, A. and J. Markel, "A spectral-flatness measure for studying the autocorrelation method of linear prediction of speech analysis," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 22, no. 3, pp. 207 – 217, jun. 1974.

[30] H. Lukashevich, M. Gruhne, and C. Dittmar, "Effective singing voice detection in popular music using arma filtering," in *10th Int. Conference on Digital Audio Effects (DAFx-07)*, Bordeaux, France, 2007, pp. 1–4.

[31] C. C. Toh, B. Zhang, and Y. Wang, "Multiple-feature fusion based onset detection for solo singing voice," in *ISMIR 2008, Automatic Music Analysis and Transcription*, pp. 515–520.

[32] H. Fujihara and M. Goto, "A music information retrieval system based on singing voice timbre," in *ISMIR, 23-30 September 2007, Vienna (Austria).*

[33] T. Zhang, "Automatic singer identification," *International Conference on Multimedia and Expo*, vol. 1, July 2003.

[34] H. Fujihara, T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H. Okuno, "F0 estimation method for singing voice in polyphonic audio signal based on statistical vocal model and viterbi search," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 5, May 2006.

[35] Y. Li and D. Wang, "Singing voice separation from monaural recordings," in *ISMIR, October 2006, Victoria, BC (Canada)*, pp. 1–4.

[36] M. Ryynnen and A. Klapuri, "Transcription of the singing melody in polyphonic music," 2006, institute of Signal Processing, Tampere University Of Technology.

[37] D. Reynolds and R. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 72–83, Jan 1995.

[38] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[39] W.-H. Tsai and H.-M. Wang, "Automatic singer recognition of popular music recordings via estimation and modeling of solo vocal signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 330–341, 2006.

[40] T.-W. Leung, C.-W. Ngo, and R. Lau, "Ica-fx features for classification of singing voice and instrumental sound," *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 2, pp. 367–370, Aug. 2004.

[41] N. C. Maddage, "A svm-based classification approach to musical audio," in *Proceedings of the 4th International Conference on Music Information Retrieval*, 2003.

[42] N. Cristianini and B. Schölkopf, "Support vector machines and kernel methods: the new generation of learning machines," *AI Mag.*, vol. 23, no. 3, pp. 31–41, 2002.

[43] A. Berenzweig and D. P. Ellis, "Locating singing voice segments within music signals," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics 2001*, New Paltz, New York, 2001, pp. 1–4.

[44] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Prentice Hall, July 1998.

[45] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[46] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection." Morgan Kaufmann, 1995, pp. 1137–1143.

[47] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*, 2nd ed. Wiley-Interscience, November 2000.

[48] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[49] K. Pearson, "Royal society proceedings, 58, 241," 1895.

[50] M. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, University of Waikato, 1999.

[51] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[52] C. Hsu, C. Chang, and C. Lin, "A practical guide to support vector classification," *Bioinformatics*, vol. 1, pp. 1–15, 2003.