UNIVERSITY OF MUSIC AND PERFORMING ARTS GRAZ
INSTITUTE OF ELECTRONIC MUSIC AND ACOUSTICS

Abstract feature sets in analysis and classification of phonation types in singing

A comparative study of different feature sets

Master's thesis

Authors: Paul Armin Bereuter, BSc 01431696

Supervisors: Univ.Prof. Dipl.-Ing. Dr.techn. Alois Sontacchi
Dipl.-Ing. Manuel Brandner, BSc

Date: Graz, January 19, 2022

Abstract

The human voice apparatus is capable of producing phonation types with different timbral characteristics. These are perceived as distinct voice-qualities such as *normal*, *breathy* or *pressed*. In professional singing these different phonation types are intentionally used to transport emotions. Although the strenuous usage of unhealthy voice qualities such as involuntarily *pressed* should be minimized in order to reduce the risk of voice disorders. Therefore, professional singers in training still strongly rely on the feedback given to them by vocal coaches or experts. However, the advances in the field of speech signal processing, with regard to classification algorithms building on supervised or unsupervised machine learning (ML), provide important tools to deepen and facilitate the feedback on sung phonation types. Typically, the foundation of this machine learning based classification task is an abstract feature set, designed to provide a meaningful description of the voice qualities.

The aim of this thesis is the comparison of abstract feature sets that, on the one hand, are already well established in speech signal processing and, on the other hand, the proposal and analysis of a novel feature set based on a signal representation built on joint temporal and spectral modulations.

The most prominent features in speech signal processing are the mel frequency cepstral coefficients (MFCCs). For them different feature set variations are created. This is done by a variation of filterbanks, the modification of the filterbank's center frequencies using vocal tract length normalization and pertubation, as well as cepstral liftering of the coefficients. The classification performance of these MFCC variants are compared and it is shown that the MFCC variant created with an inverted mel-filterbank performs best with regards to voice quality classification.

The novel feature set proposed in this thesis is derived from the so-called modulation power spectrum (MPS), which is calculated with a 2D-Fourier transform of the log spectrogram of a sung vocal sample.

A subsequent feature analysis using a Plus-L Minus-R feature selection (L-R selection) algorithm is carried out. Using the L-R selection the classification performance of the MFCC feature set created with the inverted mel-filterbank, the MPS-based feature set and a combined version are compared. Overfitting behaviour within the different feature sets are discussed. The analysis shows that the MPS-based feature set outperforms the MFCC feature set variant and therefore can be deemed as a notable alternative with regards to the classification of phonation types.

All classification tasks carried out in this thesis use support vector machines (SVMs) and a novel database created at the Institute of Electronic Music and Acoustics (IEM) in Graz. The database comprises 1140 samples recorded with 10 professional singers for three instructed voice qualities (*normal*, *breathy* and *pressed*). Furthermore, the recorded samples have been ranked in a listening experiment with regards to the perceived voice quality. This allows the usage of two sets of labels, one based on the instructions (*instruction* labels) given to the singers in the recording process and the other one based on the ranking results apprehended from the listening experiment (*experiment* labels). The comparison of the different labels allows a reduction of the full dataset to obtain more conclusive data, regarding the phonation types. Additionally, the interchange of the two label variants allows a comparison of the ranking results from the listening experiment with the classification achieved by the ML-based approach. It is shown that the ML-based classification works better if the *instruction* labels are used and also that the ML-based classification yields more correctly classified samples in comparison to the results achieved with the previously conducted listening experiment.

Kurzfassung

Der menschliche Stimmapparat ist in der Lage, klangfarblich unterschiedliche Phonationstypen zu erzeugen. Diese werden als distinkte Stimmqualitäten wie z.B. *normal*, *behaucht* oder *gepresst* wahrgenommen. Im Bereich des professionellen Gesangs werden diese Phonationstypen verwendet, um Emotionen zu transportieren. Eine belastende Verwendung ungesunder Stimmqualitäten sollte dabei vermieden werden, um das Risiko von Stimmstörungen zu verringern. Aus diesem Grund sind professionelle Sänger*innen in der Ausbildung nach wie vor auf die Rückmeldung von Expert*innen oder Gesangslehrer*innen angewiesen. Die Fortschritte in der Sprachsignalverarbeitung mit Fokus auf Klassifikationsalgorithmen, basierend auf maschinellem Lernen (ML), stellen Werkzeuge zur Verfügung, welche die Rückmeldung über die gesungenen Phonationstypen erleichtern und vertiefen können. Die Grundlage für die beschriebene Klassifikation mittels ML ist ein abstraktes Set an Beschreibungsgrößen (engl. feature set), welche die Stimmqualität ausreichend charakterisieren. Das Ziel dieser Arbeit ist der Vergleich dieser abstrakten Beschreibungsgrößen, wobei zum einen bereits in der Sprachsignalverarbeitung etablierte Größen und zum anderen neuartige Größen, abgeleitet aus einer modulationsbasierten Signalrepräsentation verarbeitet werden.

Die prominentesten Merkmale in der Sprachsignalverarbeitung sind die Mel-Frequenz-Cepstrum-Koeffizienten (MFCCs). Für sie werden unterschiedliche Varianten mittels Variation der Filterbänke, Modifikation der Filterbank-Mittenfrequenzen durch Vokaltraktlängennormalisierung und -pertubation sowie durch cepstrales Liftering der Koeffizienten erstellt. Die Ergebnisse der Stimmqualitätsklassifikation dieser MFCC-Varianten werden verglichen, und es zeigt sich, dass die mit einer invertierten Mel-Filterbank erstellte MFCC-Variante die besten Ergebnisse erzielt.

Die vorgeschlagenen neuartigen Beschreibungsgrößen werden aus dem sogenannten Modulationsleistungsspektrum (MPS) abgeleitet, das mit einer 2D-Fourier-Transformation des logarithmierten Spektrogramms der Gesangssignale berechnet wird.

Die unterschiedlichen Beschreibungsgrößen werden unter Verwendung eines Plus-L Minus-R Algorithmus (L-R Auswahl) weiter analysiert. Mit Hilfe der L-R Auswahl wird die Stimmqualitätsklassifikation der MFCC-Variante, der MPS-basierten Größen sowie eines kombinierten Satzes beider Größen verglichen. Die Analyse zeigt, dass das MPS-basierte Feature-Set die MFCCs übertrifft und daher durchaus als Alternative in Bezug auf die Stimmqualitätsklassifikation angesehen werden kann.

Alle in dieser Arbeit durchgeführten Klassifikationsaufgaben verwenden Support Vector Machines (SVMs) und eine neue Datenbank, die am Institut für Elektronische Musik und Akustik (IEM) in Graz erstellt wurde. Die Datenbank umfasst 1140 Aufnahmen, die mit 10 professionellen Sänger*innen für drei instruierte Stimmqualitäten (*normal*, *behaucht* und *gepresst*) aufgenommen wurden. Die Aufnahmen wurden in einem Hörversuch hinsichtlich der wahrgenommenen Stimmqualität bewertet. Dadurch können zwei Sets an Stimmqualitätslabels, eines basierend auf den Anweisungen, die die Sänger*innen während der Aufnahme erhielten (Instruktionslabels), und das andere basierend auf den Ergebnissen des Hörversuchs (Hörversuchslabels) verwendet werden. Durch einen Labelvergleich kann der gesamte Datensatz reduziert werden, um in Bezug auf die Phonationstypen aussagekräftigere Daten zu erhalten und es kann die Stimmqualitätsbewertung aus dem Hörversuch mit der ML-basierten Klassifikation verglichen werden. Es zeigt sich, dass mittels ML-basierten Klassifikation bessere Ergebnisse mit den Instruktionslabels erzielt werden und dass diese im Vergleich zu den Ergebnissen des zuvor durchgeführten Hörversuchs einen höheren Prozentsatz an korrekten Klassifizierungen liefert.

Statuatory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

date	Paul Armin Bereuter

Contents

List of Figures

List of Tables

Mathematical Notation

The mathematical notation used in this thesis is summarized in Table 1.

Table 1 *Mathematical symbols and notation*

a, b, c	scalars
$\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$	vectors
$\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$	matrices or a twodimensional set of values
$x[n]$	a discrete-time signal
$(x * h)[n]$	discrete (circular) convolution of $x[n]$ and $h[n]$
$\tilde{x}[k] = \mathcal{F}_{n \mapsto k}\{x[n]\}[k]$	discrete N-point Fourier transform $\tilde{x}[k]$ of the discrete-time signal $x[n]$
$\tilde{\boldsymbol{X}}[\boldsymbol{\omega}] = \mathcal{F}^{\text{2D}}_{\boldsymbol{\theta} \mapsto \boldsymbol{\omega}}\{\boldsymbol{X}[\boldsymbol{\theta}]\}[\boldsymbol{\omega}]$	discrete twodimensional $[N \times M]$-point Fourier transform $\tilde{\boldsymbol{X}}[\boldsymbol{\omega}]$ of the discrete time-frequency signal representation $\boldsymbol{X}[\boldsymbol{\theta}]$
$\lvert z \rvert$	absolute value of a complex variable $z \in \mathbb{C}$
$\mathcal{N}(\mu, \sigma^2)$	a Gaussian random variable with mean μ and standard deviation σ
$\mathcal{U}(a, b)$	A uniformly distributed random variable in the interval $[a, b]$
$\mathbb{E}\{x[n]\}$	the expected value of $x[n]$ over time
$\#\,(\cdot)$	number of elements
$\hat{a}$	*estimation* of a quantity a
f_0	fundamental frequency
f_s	sampling frequency
$\mathscr{X}$	a feature set
:	subject to
$\in$	element of
$\forall$	for all

1 Introduction

This thesis comprises 4 parts and an appendix. The first part, chapter 1, deals with the motivation behind the classification of phonation types in singing and also deals with the definition of various terms that are used throughout this thesis. Chapter 2 comprises the theory behind the analyzed feature sets as well as the used classification model. The next part is given with chapter 3, which deals with the implementation of the carried out classification analysis, in terms of the used data, the implementation of the classification model as well as the quantification of the different feature sets' classification performance. The thesis is concluded by the summary and discussion of all made observations and the outlook on possible future research areas in chapter 4. The appendix holds additional tables, plots and results, which are referenced throughout this thesis.

1.1 Motivation and Background

A crucial part in the vocal training of professional singers is to gain the ability of being able to fully control the sung phonation type as well as the predominant usage of healthy phonation types. Phonation types are also often referred to as voice-qualities which according to [58, p.172] describe a "[...]personal voice timbre[...]". In accordance to fundamental research gathered in [57, p.152-157], these voice qualities are strongly associated with the singer's emotion. Whereas sweet, seductive, soft but also sad and depressed emotions are associated with a *breathy* voice, hard feelings and anger are transported using a *pressed* voice quality in singing. By being able to control the voice quality, singers are able to add an emotional layer to their performances leading to a transportation of feelings from singer to listener. Control in this sense refers to maintaining sung phonation types by varying pitches and loudness. Nonsingers often exhibit fluctuating voice qualities with changing pitches and loudness [59, p.74]. Another important benefit of having full control over the sung voice quality is the prevention of voice disorders, for which an extensive usage of unhealthy voice qualities, such as unintentionally *pressed*, can be a source [60]. Within professional vocal training a vital aspect in gaining control over the voice quality is feedback on the current sung voice quality, which is usually given by a professional vocal coach. In order to intensify and extend this feedback, current research such as [53], [19], [18] and [20] use the advances of digital signal processing with respect to machine learning (ML) in order to set up supervised learning problems, which allow a computational classification of sung vocal signals with regards to different phonation types. One drawback within this current research are the limited datasets. The processed datasets often only comprise sung vocal samples of one or two singers and descriptive measures that are very common within the research fields of signal processing and automatic speech recognition, but nevertheless do not accurately reflect the physiological processes that lead to the distinction of different voice qualities in singing. This is where the focal point of this thesis lies. The results of this thesis provide novel descriptive measures that allow an ML-based classification of phontation types in singing as well as a extended performance analysis of established measures and modifications of such. A newly created dataset which was recorded at the Institute of Electronic Music and Acoustics (IEM) at the University of Music and Performing Arts Graz, as well as evaluated and labelled with a listening experiment, allows to draw a differentiated conclusion on the generalization capabilities of the proposed descriptive measures and provides a performance analysis of the established measures. The following sections, 1.2-1.4, lay out a general introduction into the physiology of speech production and the different phonation types, a

comparison between speech and singing voice as well as the usage of machine learning (ML) within the analysis of speech and singing voice.

1.2 Voice Physiology and Phonation Types in Singing

1.2.1 The Human Speech Production Apparatus

The physiology of singing and speech are based on the same physical procceses occuring inside the human voice apparatus, which generally can be divided into three parts. The first part, marked in blue in Figure 1.1, are the respiratory organs (lungs) which generate the air-flow passing through the trachea into the second part, the so-called voice source [59] or glottal region [10]. In Figure 1.1, the voice source is outlined in orange and a close up is depicted in Figure 1.2. The tracheal airflow travelling from the lungs towards the glottal region is periodically interrupted by vibrating vocal folds, resulting in the so-called glottal or transglottal flow. The vibrating vocal chords are the defining factor of voiced speech, which can be presumed to be the general case for singing [57]. In an acoustical and speech signal processing context the glottal flow (GF) or its derivative (dGF) are often referred to as the excitation or source signal [15]. The glottal flow passing through the vocal folds resembles a pulse train. An exemplary illustration of the glottal flow's waveform is shown in part one of Figure 1.1. The excitation signal's frequency spectrum is visible in part 2 of Figure 1.1 (marked in orange) and shows a frequency spectrum with whole-number multiples of the fundamental frequency, whose intensity decend with increasing frequency. The close up depiction of the voice source region in Figure 1.2 shows that the vocal folds form a bottleneck seperating the trachea and the adjacent vocal tract. After the excitation signal is produced through the tracheal airflow passing through the vibrating vocal folds, the pulse train like signal enters the vocal tract which forms the third part of the human speech production apparatus. The vocal tract acts as a resonator, which filters the excitation signal and applies the so-called formants. The formants are the defining aspects to what the listener perceives as vowels e.g. */a/*, */e/*, */i/*, */o/* or */u/*. As the main focus of this thesis lies on the voice quality, which is defined in the voice source region, a more thorough discussion of the voice source region, marked in Figure 1.1, and its influence on the creation of different phonation types is carried out in subsection 1.2.2.

1.2.2 The Physiology of Different Phonation Types

Different phonation types come about with different vibration modes of the vocal folds. In this thesis three voice qualities are described and are subject of the executed analysis and classification. The voice qualities of interest are *modal/normal*, *breathy* and *pressed* voice quality. Originating in speech signal processing, Gobl *et al.* proposed the sources of different voice qualities based on the physiological processes in the glottal region in [15], and Sundberg defines phonation types for singing based on physiological processes in [57]. The definitions on the production of the three relevant voice qualities in this thesis can therefore be considered as a unified definition of both Gobl and Sundberg's take on the production of different phonation types due to different glottal behaviour. Helpful illustrations to deepen the understanding on how different phonation types are formed within the human body are given by Figure 1.2 and 1.3. Whereas the glottal region illustrated in Figure 1.2 is displayed in a frontal plane dissection, Figure 1.3 depicts the glottal region in a horizontal dissection.

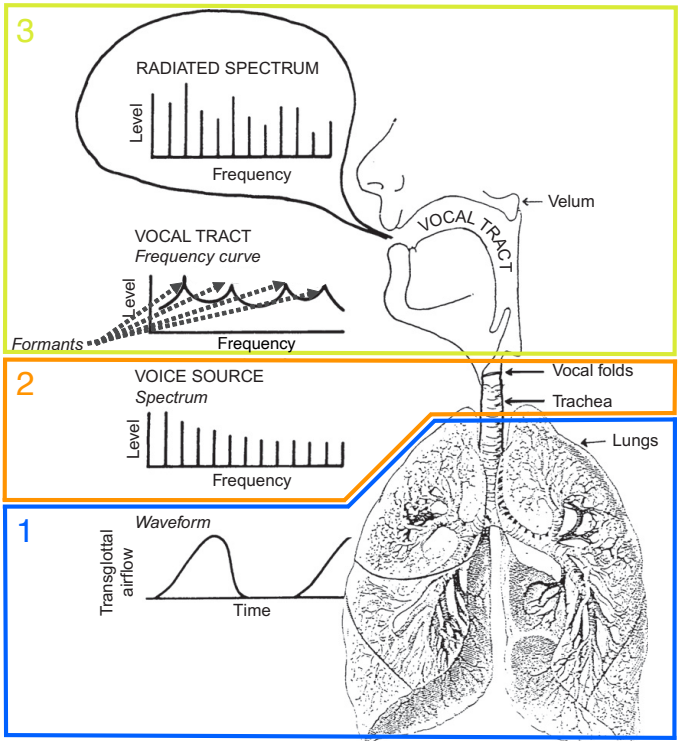


Figure 1.1 *The human speech production apparatus divided into three parts. Source: [59, Fig. 1]*

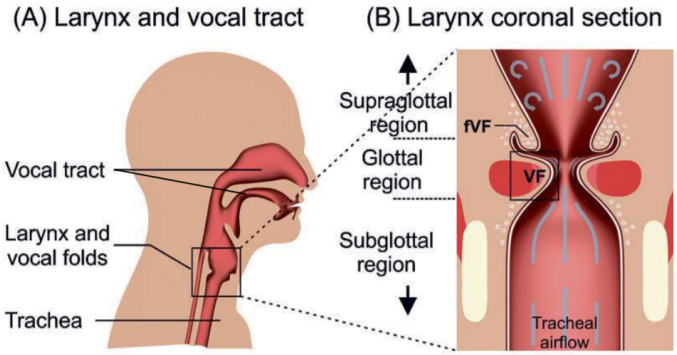


Figure 1.2 *An anatomic depiction of the glottal region and the vocal folds. Source: [10, Fig. 1]*

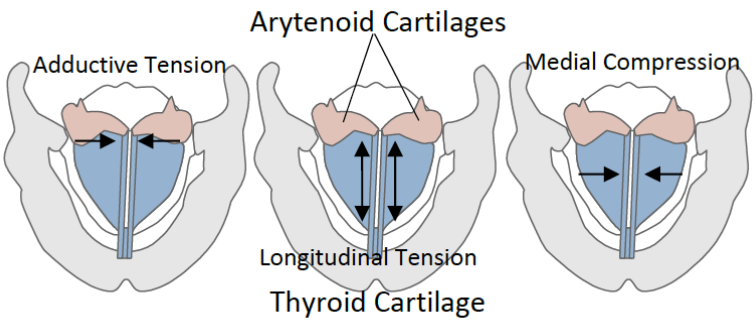


Figure 1.3 *Horizontal dissection of the glottal region. Source: [65, Fig. 2]*

Using Figure 1.2 and 1.3, the production of the three phonation types can be explained as follows:

Breathy voice quality

In [15], Gobl states that breathiness in speech occurs when the adductive tension, present to the top end of the vocal folds, is reduced to a minimum, and the weak medial compression is applied. The adductive tension and medial compression are visualized with black arrows in Figure 1.3. The minimal tension and weak compression cause an incomplete closure of the vocal folds while vibrating. Due to the minmal adductive tension, the vocal folds reside in a Y-shaped state, leaving an opening at the top of the vocal folds, even during a closure phase of one glottal cycle [57]. The definition of a glottal cycle is discussed in section 1.3. The remaining opening during the glottal closure phase for *breathy* voice is depicted in the middle of Figure 1.4. The constant opening lets the turbulent tracheal airflow enter the vocal tract at any time causing a *breathy* voice perception. The constant airflow entering the vocal tract is often referred to as aspiration noise [24].

Normal voice quality

In a speech context, *normal* voice quality is also referred to as *modal* voice. In [15], Gobl cites moderate adductive tension and medial compression as the reason for a full-length vibration of the vocal folds, as it is the case for *modal* voice quality. This implicitly leads to a full glottal closure of the vocal folds during vibration. When taking a look at the literature concerning the singing domain, Sundberg further differentiates between *normal* and *flow* voice quality [57]. The explanation of *normal* phonation from [57] coincides with the one from [15], whereas *flow* phonation can be described as the phonation type that occurs when "[...] glottal adduction is reduced to a minimum" resulting in a voice quality that is still not *breathy* [59, p.74]. So, according to [59], *flow* phonation is the phonation type in the *normal* voice quality range that is closest to *breathy* phonation but isn't perceived as *breathy*. *Flow* phonation can also be viewed as a singing technique, which allows to reach " [...] higher levels of loudness with minimum effort." [53, p.] ". In the context of this thesis, the differentiation between *normal* and *flow* phonation is neglected and both voice qualities are referred to as *normal* or *modal* voice quality.

Pressed voice quality

When looking at Figure 1.4, it becomes visible that no *pressed* voice quality is mentioned. However, the voice quality whose glottal closure instant (GCI) is illustrated on the right hand side of the *modal* voice, is called *creaky*, which is also defined in [15]. Creak, according to [15], is defined through high adductive tension and medial compression. Therefore, it can be viewed as the opposite of *breathy*. The high adductive tension is visualized in Figure 1.4, where contraction of the top of the vocal folds is visible. This fits the definition of what is called *pressed* phonation in [57], where it is defined in a singing context. An extreme, strenuous usage of *pressed* phonation is also often a cause of voice disorders [60].

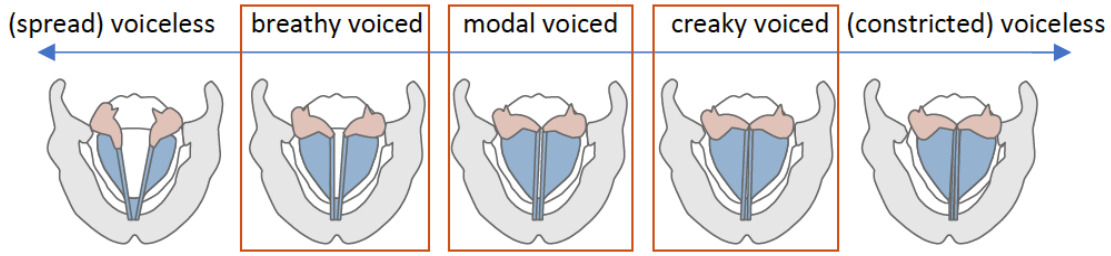
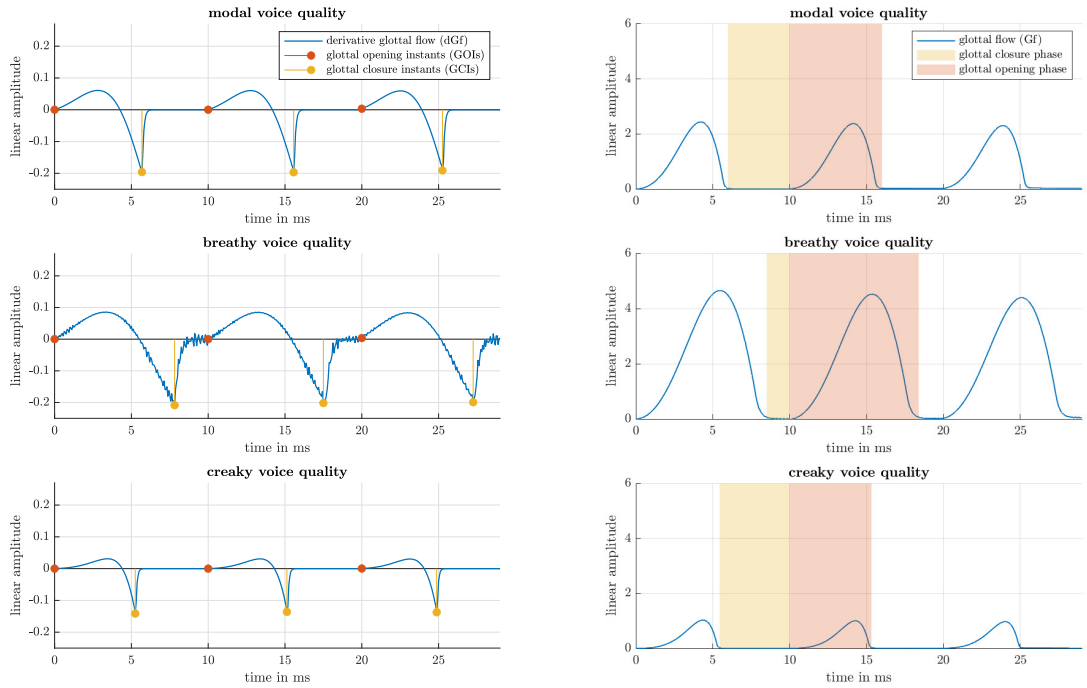


Figure 1.4 *Vocal folds at glottal closure instants for different voice qualities. Source: [65, Fig. 1]*

As a result of the usage of mentioned phonation types, the tracheal flow is also interrupted in different ways, leading to different waveforms concerning the glottal flow. The modelling of these waveforms is discussed in the following section.

1.3 The Voice Source in a Signal Processing Context

An exemplary sketch of a glottal flow waveform is added of Figure 1.1. This waveform visualizes air passing through the vocal folds. The most prominent model delivering a mathematical description of the glottal flow and its derivative is the LF-model established by Fant & Liljencrants in [12]. For the analysis carried out in this thesis, the mathematical formulation is not relevant. Nevertheless, the waveforms, that can be generated by using the LF-model, contain important insights on the distinction of the phontation types and they also reflect the physiological processes mentioned in subsection 1.2.2. Exemplary waveforms for each of the three phonation types created with the synthesizer presented in [4], are depicted in Figure 1.5. The synthesizer employs the LF-model of [12] in order to create the voice source signal named derivative glottal flow (flow change), which is depictd in Figure 1.5 (a). The integrated version of the derivative glottal flow, called glottal flow is shown in Figure 1.5 (b). The glottal flow can be understood as the airflow through the glottis and its derivative holds information on the flow change. It has to be pointed out, that in the synthesizer presented in [4] and the created figure Figure 1.5 use the term *creaky*, but as the definitions of *creaky* voice in [15] and *pressed* from [57] coincide in terms of the physiological processes, the term *creaky* in Figure 1.5 is equivalent to *pressed*, which is used in this thesis.



(a) exemplary derivative glottal flow (flow change)

(b) exemplary glottal flow

Figure 1.5 *Glottal flow and flow change for different voice qualities, the depicted curves are synthesized signals and their amplitudes are not to be interpreted as physical measures.*

As shown in Figure 1.5, the voice source signals differ for different phonation types. One aspect which distinguishes the source signal of the different voice qualities is the course of the derivative glottal flow (dGF) depicted in Figure 1.5 (a). For *breathy* voice quality the flow change (dGF) is the least sparse signal in Figure 1.5 (a), because the signal part with zero amplitude is limited to a very short time span. The aspiration noise given due to the triangular shaped opening during the closed phase of a glottal cycle, is also visible. In comparison to *breathy* phonation, *normal/modal* voice quality exhibit a larger time where the amplitude is zero, and the dGF for *pressed* phonation shows the longest time span with an amplitude of 0. Another aspect that the dGF of *pressed* voice shows, is the lower positive amplitude and a sharper negative impulse at the GCI, which is the glottal closure instant, denoting the instant at which the glottis is closed. The glottal opening instant (GOI) is also marked in the dGF plots of Figure 1.5. The integrated version of the dGF, the glottal flow, holds information on the airflow passing the glottis. The highest airflow is given with *breathy* phonation, followed by *normal/modal* and *pressed* voice quality. Additionally, the glottal closure phase, lasting from the GCI to the GOI, and the opening phase, lasting from the GOI to the next GCI, also indicate the distinction between the voice quality classes. The shortest closure phase is given for *breathy* voice quality as the vocal folds do not really close during the glottal cycle. The longest closure phase is given with *pressed* voice quality, due to the high tension on the vocal folds, which leads to a higher glottal resistance and vocal fold closing [57]. This undermines the source filter theory, which also includes the view that the phonation type is an aspect of the singing voice, predominantly formed by the voice source at the glottis and the vocal tract is responsible for applying the formants, perceived as vowels [57] .

1.4 Overview on Speech and Singing Analysis

As shown in the previous section, the voice quality is an aspect formed at the voice source, which inevitably brings the idea with it that the phonation type is detectable when the source signal is estimated. This theoretical consideration lead to a lot of research that has been carried out and deals with the estimation of the source signal through a procedure denoted glottal inverse filtering (GIF) proposed by Alku in [1]. The main idea behind the inverse filtering method is the estimation of the vocal tract filter based on adaptive filter theory such as the linear predictive coding (LPC) [22], and the subsequent inverse usage of the estimated vocal tract filter onto the sung vocal signal [1]. There have been certain improvements towards the LPC-based filter estimation with approaches such as cepstral liftering of the excitation signal, before vocal tract estimation [51] or the usage of weighting functions with the aim to deemphasize certain parts in a vocal signal, which worsen the estimation performance of the vocal tract filter [7]. Based on Alku's glottal inverse filtering procedure from [1], analysis environments and repositories have been created such as [7] and [2], which hold a multitude of scripts that allow the glottal source signal analysis using the software application `Matlab`. However, the vocal tract filter esimtation using LPC comes with limits, that arise with increasing fundamental frequencies as summarized in [4], leading to a faulty distinguishment of the voice qualites with regards to higher pitches. Thus, more recent approaches including machine learning (ML) based classification tasks deal with direct processing of the vocal signal. The work of Kathania in [20] as well as Kadiri with [19] and [18] present phonation type classification approaches in which features descriptive of the voice quality are calculated and used in classic ML-based classification tasks. The advantage is that the underlying phonation type descriptive features are directly calculated from a time frequency representation of the vocal signals. Different types of voice quality descriptive features as well as the potential and the methods behind the ML based classification tasks carried out in this thesis are dealt with in chapter 2.

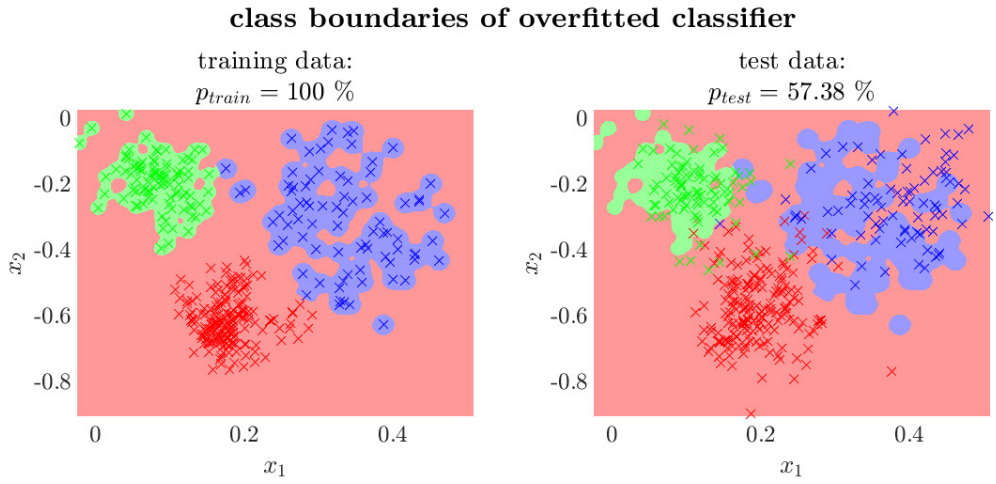
2 Methods and Theoretical Background

The main focus of this thesis lies on the classification of phonation types used in singing. Therefore, a machine learning (ML) based classification task using supervised learning is set up. The structure of such a task is mentioned in section 2.1 and subsequently, the theoretical background of each step in the classification task is discussed.

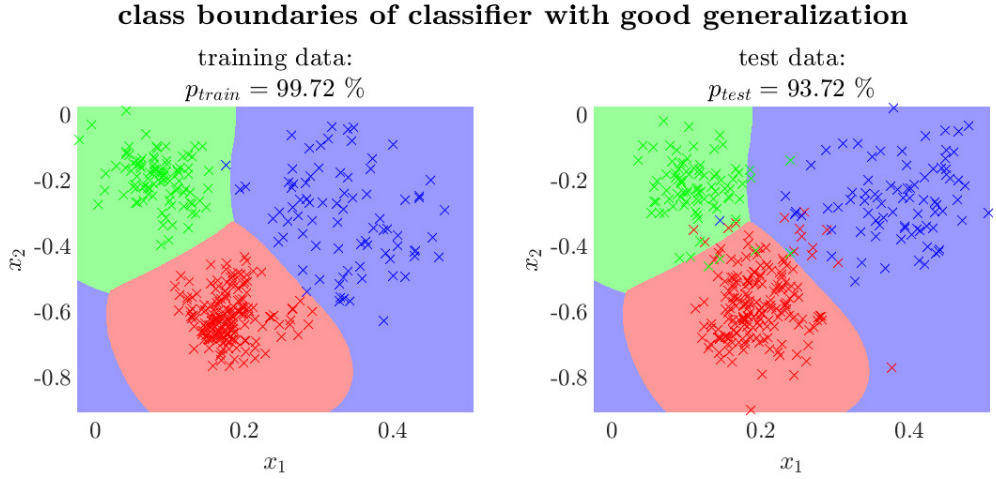
2.1 Classification using Supervised Learning

In [5] a supervised learning problem concerning classification is defined by an underlying feature set whose classes are known. The class affiliations are referred to as *labels* or *targets* for each data sample. This means that a class label exists for each sample contained in a feature set. The feature sets compared in this thesis are generally denoted as $\mathscr{X}$ and are representative of the voice quality contained in sung vocal samples. The feature calculation process is called feature extraction [5, p.2]. The features used in this thesis are different variants of the mel frequency cepstral coefficients (MFCCs) (see section 2.2) and the second features are derived from the joint temporal and spectral modulation signal representation called modulation power spectrum (MPS) (see section 2.3). Due to the fact that the MFCCs are derived from a time frequency signal representation and the other calculated features are based on the MPS, they are referred to as so-called abstract feature sets, as they are not directly measureable within the time signal.

In a supervised classification task the feature set is split into a *training and test set*, where the training set is used to fit a classifier towards the data. Fitting a classification model/classifier means that the model learns the description of the classes provided by the abstract feature set and subsequently enables the classification of data the classifier has not seen yet, that's where the test data comes into play. The test data is held out and presented to the trained classifier, which allows an assessment of the classifier's performance. Thus, it is often also called *hold-out set*. The classifier's ability of distinguishing new data is called *generalization* [5]. The score, on the test data gives information on how well a classifier generalizes and is calculated as the percentage of correctly classified samples. But it is advisable to also view *training* and *test score* in relation, because they hold important information on the selected classification model (classifier), more precisely on the complexity of the model. A high *training score* and a lower *test score* indicate what in [5, p.32] is referred to as *overfitting*. Overfitting occurs when, the complexity of a model is too high. The model then fixates on the training data, as shown in the academic example depicted in Figure 2.1. It visualizes the class boundaries within the space spanned by the feature vectors. This is called a *feature space*. In the case of Figure 2.1 the *feature space* is twodimensional and spanned by the features x_1 and x_2. The class boundaries for each class are indicated by varying colors. The marks indicate the samples of a test and training set. It is clearly visible that Figure 2.1 (b) performs way better on the test set. The better performance is also displayed by the test and *training scores* denoted as p_{test} and p_{train}, which give the percentage of a correctly classified sample of the respective subset (see Equation 3.3 and 3.4). The subfigure Figure 2.1 (b), shows an academic example of good generalization. Both *training* and *test score* lie in the same vicinity and the classification of the test set, which the model has not seen during its fitting procedure, works reasonably well.



(a) overfitted class boundaries



(b) good generalization

Figure 2.1 *Academic example on the effects an overfitted classifier has on predicted class boundaries and the test and training score.*

In this thesis the chosen classifiers are support vector machines (SVMs). The theoretical background and their adjustable parameters are discussed in section 2.5. The academic example depcited in Figure 2.1 (a) illustrates what happens if classifier parameters are chosen poorly. The complexity of a classifier is not only determined by certain classifier parameters but also with the dimensionality of the present feature space, which is determined by the number of features. If the dimensionality increases the training data within the higher dimensional feature space becomes sparser. This makes it easier for the classifier to find a possibility of separating the classes within the feature space, leading again to the effect that the classifier is overfitted towards the training data. This is what generally is referred to as *the curse of dimensionality* [5, p.33-38]. Thus, it makes sense to montior the classifier's performance over an increasing number of features. This is possible with a feature selection algorithm, which allows a reduction of the feature space and selects the features into a certain order. If the number of features the algorithm has to select is increased, reduced feature sets with increasing number of features are obtained and their performance can be evaluated and monitored. This is exactly what is done in the analysis carried out in section 3.5. A short introduction to the used features is given in section 2.2 and 2.3, the employed SVM classifier is mentioned in section 2.5 and the used

Plus-L Minus-R feature selection algorithm in order to monitor the influence of the feature space dimensionality is theoretically introduced in section 2.6.

2.2 Mel Frequency Cepstral Coefficients (MFCCs)

The first features used are mel frequency cepstral coefficients (MFCCs). The MFCCs are often used features in the field of automatic speech recognition; for instance, in HTK a toolkit to create hidden markov models in [66]. Basically, the MFCCs are cepstral representations of a logarithmized and in auditory frequency bands summarized frequency spectrum of a signal. The cepstral domain is entered by applying an inverse Fourier transform or a discrete cosine transform (DCT) onto a frequency spectrum of a signal. It can be viewed as a "spectrum of a spectrum". Because two transforms and the logarithm are involved in order to enter the cepstral domain, a filter operation, which in the time domain is given by a convolution and a multiplication in the frequency domain, is represented in the cepstral domain as a sum. A common model encountered within speech signal processing is the source filter model, which is a mathematical formulation of the physiology described in subsection 1.2.2, which states that a speech signal is given through the convolution of a voice source signal $E[n]$ with a vocal tract filter response $h_{\mathrm{VT}}[n]$ as denoted in Equation 2.1. In the cepstral domain this convolution becomes a sum, which is a reason why cepstral analysis is often applied in speech analysis [47, p.210-215].

$$s[n] = (E * h_{\mathrm{VT}})[n]. \tag{2.1}$$

1. In order to calculate the MFCCs the first step is the Fourier transform of the signal into the frequency domain as formulated in Equation 2.2.

$$S[k] = \mathcal{F}_{n\mapsto k}\{s[n]\}[k] \tag{2.2}$$

2. The absolute values frequency spectrum is then filtered with a filterbank, which sums the frequency bins k into N_{MEL} frequency bands in order to reduce the number of frequency bins, which is also referred to as *binning* [66, p.95]. The filterbank is created according to the mel frequency scale, which originates from perceptual properties of the human auditory system. The used filterbanks ares discussed in subsection 2.2.1. The binning is carried out in the frequency domain, thus the filtering process can be carried out with a vector matrix multiplication as formulated in Equation 2.3.

$$\boldsymbol{m} = \boldsymbol{H} \cdot \boldsymbol{x}$$

$$\begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_{N_{\mathrm{MEL}}} \end{pmatrix} = \begin{bmatrix} h_0[0] & h_0[1] & \dots & h_0[N-1] \\ h_1[0] & h_1[1] & \dots & h_1[N-1] \\ \vdots & & & \\ h_{N_{\mathrm{MEL}}-1}[0] & h_{N_{\mathrm{MEL}}-1}[1] & \dots & h_{N_{\mathrm{MEL}}-1}[N-1] \end{bmatrix} \cdot \begin{pmatrix} |S[0]| \\ |S[1]| \\ \vdots \\ |S[N-1]| \end{pmatrix} \tag{2.3}$$

Where $\boldsymbol{m}$ is the vector that holds the binned frequency spectrum in N_{MEL} frequency bands. $\boldsymbol{H}$ is the filterbank used to sum the frequency bins. The rows of $\boldsymbol{H}$ hold the N-point frequency responses of the filters for each frequency band and $\boldsymbol{x}$ holds the N-point frequency transformed absolute values of the signal in vector notation.

3. The binned frequency spectrum is then logarithmized and transformed into the cepstral domain with either an inverse Fourier transform or a discrete cosine transformation (DCT). In this thesis the same calculation approach as denoted in [66] is used, where a Type II DCT is used. The calculation of the i^{th} mel frequency cepstral coefficient for a filterbank with N_{MEL} frequency bands is given with:

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=0}^{N_{\text{MEL}}-1} \log(m_j) \cos\left(\frac{\pi i}{N}(j-0.5)\right) \tag{2.4}$$

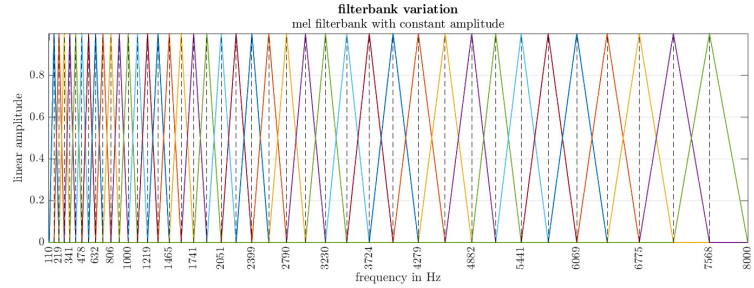
These three steps outline the calculation process of the mel frequency cepstral coefficients. In this thesis several modifications are applied in order to create different MFCC variants. The used filterbanks are discussed in subsection 2.2.1 and 2.2.2 and the scaling procedure applied in the cepstral domain, which is mentioned in subsection 2.2.3.

2.2.1 Mel Filterbank Variation

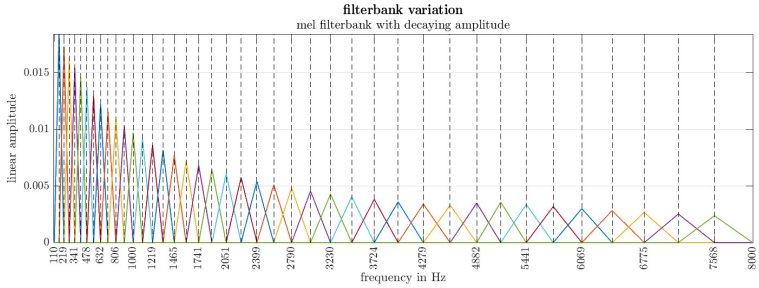
The filterbank used in the calculation of MFCCs is classically a triangular shape filterbank in which the center frequencies are equally spaced along the mel frequency scale. The mel scale has been defined with the help of psychoacoustical listening experiments, where the listeners were asked to quantify the perceived pitch [50]. It has been shown that perceived pitch in relation to the frequencies is not linear. As a result, the mel scale was proposed. The term mel is derived from the word "melody" and also specifies the unit of the the perceived pitch [50]. Mathematically the scale is defined as:

$$Mel(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.5}$$

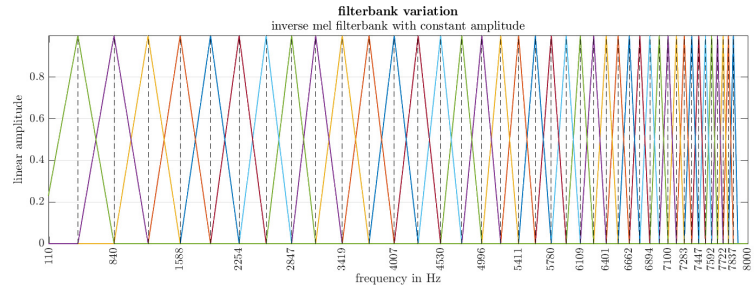
The filterbanks used, span a frequency range of $110\,\text{Hz}$ to $8000\,\text{Hz}$, but vary in the spacing of their center frequencies. Figure 2.2 holds a depiction of all filterbanks used in this thesis. The center frequencies of the the classic triangular shaped mel filterbanks depicted in Figure 2.2 (a) and (b) are spaced according to Equation 2.5. The difference between the filterbanks depicted in Figure 2.2 (a) and (b) is the amplitude. Figure 2.2 (a) depicts a filterbank with a constant amplitude of 1 and Figure 2.2 (b) depicts a filterbank in which the area under the filter curves is normalized to 1 which, in terms of filtering means that each filter-band contains the same amount of energy in case of a white noise input signal. The filterbanks depicted in Figure 2.2 (c) and (d) are inverted versions of the classic mel filterbanks. The reason for their usage is described in subsection 3.4.1. The last filterbank variant used in this thesis is a linearily space filterbank Figure 2.2 (e), where the mel scale is not applied for the center frequency spacing. For this variant, the term mel frequency cepstral coefficient is misleading, as no mel frequency spacing is applied. The coefficients calculated using this filterbank are often also referred to as linear frequency cepstral coefficients or LFCCs as for instance in [20]. Nevertheless, as in this thesis the variant is mentioned under the umbrella term mel frequency cepstral coefficients as the only difference between MFCCs and LFCCs is the center frequency spacing. All filterbanks were created using D. Ellis' rastamat library [11].



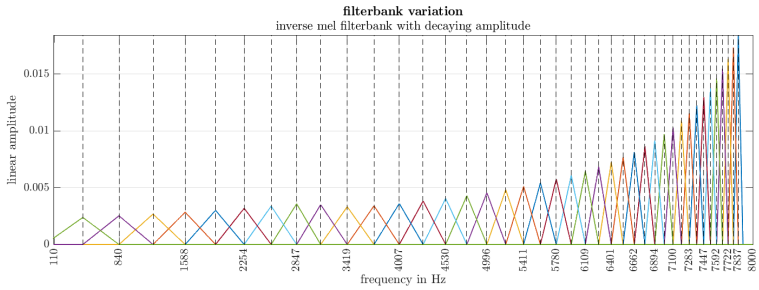
(a) classic Mel-filterbank with constant amplitude



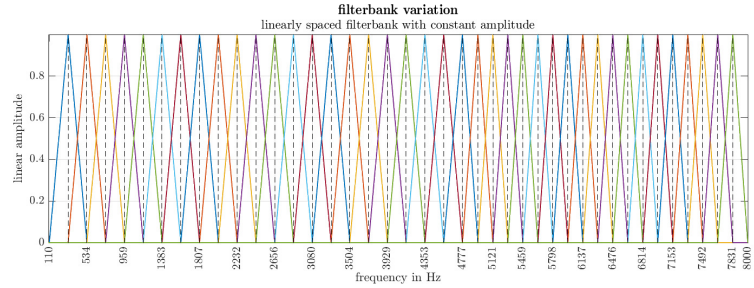
(b) classic Mel-filterbank with decaying amplitude



(c) inverse Mel-filterbank with constant amplitude (inverted filterbank (a))



(d) inverse Mel-filterbank with increasing amplitude (inverted filterbank (b))



(e) linearly spaced filterbank with constant amplitude

Figure 2.2 *Used filterbank types to create MFCC variations, filterbanks are created with* `fft2melmx()` *from [11].*

2.2.2 Vocal Tract Length Normalization and Perturbation using Frequency Warping

An approach of modifying the filterbank center frequencies is presented in [23], which is also employed in [66] and in [16]. The first augmentation of the filterbank center frequencies, used in this thesis is proposed in [23] and is called vocal tract length normalization (VTLN) and has the aim to diminish the influence of the vocal tract length by shifting the filterbank center frequencies according to a warping factor α. The center frequencies f_c are shifted according to Equation 2.6 from [16].

$$\tilde{f}_c = \begin{cases} \alpha \cdot f_c, & f_c \leq f_{\text{max}} \cdot \frac{\min(\alpha,1)}{\alpha} \\ \frac{f_s}{2} - \frac{fs/2 - f_{\text{max}} \cdot \min(\alpha,1)}{fs/2 - f_{\text{max}} \cdot \frac{\min(\alpha,1)}{\alpha}} \cdot (f_s/2 - f_c), & \text{otherwise} \end{cases} \tag{2.6}$$

In order to estimate the frequency warping factor for the VTLN a minimum mean square error (MMSE) calculation, with a reference MFCCs $\boldsymbol{c}_{\text{ref}}$, is set up. A discretely spaced warping factor vector is created and each warping factor is used to shift the filterbank center frequencies according to Equation 2.6 and the MFCCs $\boldsymbol{c}$ are calculated with Equation 2.4. The estimated frequency warping factor $\hat{\alpha}_{\text{VTLN}}$ is then given as the warping factor for which the minimum error occurs, as noted in Equation 2.7.

$$\hat{\alpha}_{\text{VTLN}} = \underset{\alpha}{\operatorname{argmin}} \left(\mathbb{E}\left\{ (\boldsymbol{c} - \boldsymbol{c}_{\text{ref}})^2 \right\} \right) \;:\; \hat{\alpha}_{\text{VTLN}} \in [0.88, 1.12] \tag{2.7}$$

The second filterbank center frequency augmentation is the method called vocal tract length pertubation (VLTP) proposed in [16], which also follows the idea that shifting the filterbank center frequencies can diminish the influence of the vocal tract length on the calculation of the MFCCs. In contrast to VTLN, where the warping factor responsible for shifting the center frequencies is estimated in order to align the influence of varying vocal tract lengths of different singers, VTLP is designed to randomize the frequency warping factor, which can be understood as a vocal tract length whitening process. Thus, the frequency warping factor for VTLP is chosen as a uniformly distributed random variable within the interval $0.88 \leq \hat{\alpha}_{\text{VTLP}} \leq 1.12$, as mentioned in Equation 2.8

$$\hat{\alpha}_{\text{VTLP}} = \mathcal{U}(a, b) = \mathcal{U}(0.88, 1.12) \tag{2.8}$$

The interval for $\hat{\alpha}_{\text{VTLP}}$ is given in [23] and covers the $25\,\%$ variation in vocal tract length, that is observable for adults.

In order to shortly assess the estimation performance of the VTLN warping factor estimation mentioned in Equation 2.7, the influence of vowels and voice quality on the estimation results is investigated. With the synthesizer presented in [4], signals were synthesized where the vocal tract filter holding the formant structure of the signal, was shifted with a fixed factor α_{true}. A reference signal, where no shift was applied, is used to estimate the shifting factor $\hat{\alpha}$. However, the voice quality and vowel of the synthetic reference signal and shifted signal do not always coincide. Thus, the influence of different voice qualities, or vowels in synthesized signals on the warping factor estimation process is visualized in Figure 2.3 and Figure 2.4. A perfect estimation would result in a diagonal line in the subplots (see Figure 2.3 and Figure 2.4). It is shown in Figure 2.3 that if samples with different voice qualities are used, no drastic influence on the estimation results have to be anticipated. Regarding the influence of the vowels on the estimation process, it is visible in Figure 2.4 that the estimation works best, if the reference signal and the signal for which the warping factor is to be estimated contain the same vowel.

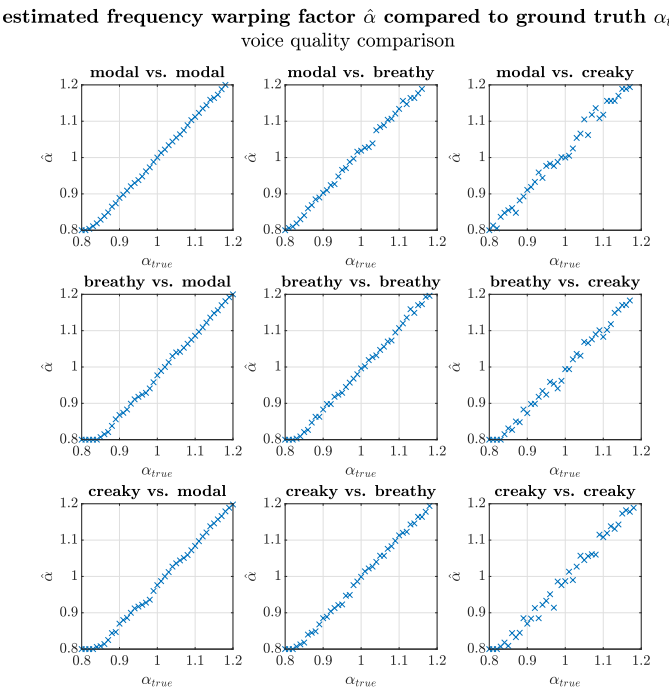


Figure 2.3 *Frequency warping factor estimation, effects of different voice qualities.*

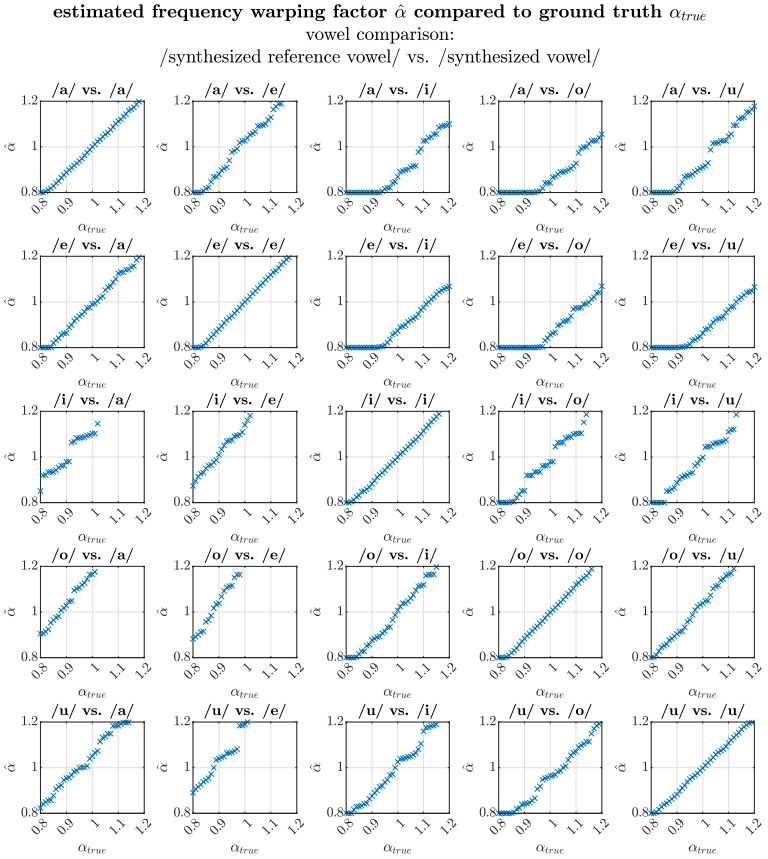


Figure 2.4 *Frequency warping factor estimation, effects of different vowels.*

2.2.3 Cepstral Liftering

The last augmentation method of the MFCC comprises a scaling procedure using a cepstral lifter. A lifter, which is an innuendo to the term “filter“, is a signal manipulation in the cepstral domain by multiplication of a cepstral lifter signal with the cepstral coefficients. As proposed in [47, p.415] liftering emphasizes the mid-range MFCCs in order to equalize the amplitudes of the mid-range MFCCs towards the amplitudes of the lower MFCCs, which are linked to channel conditions, such as noise. In this thesis the cepstral lifter is chosen in accordance to [66], where a sinusiodal lifter is proposed. The liftering of the i^{th} MFCC is formulated in Equation 2.9.

$$\tilde{c}_i = 1 + \frac{L}{2} \cdot \sin\left(\frac{\pi n}{L}\right) c_i \tag{2.9}$$

The lifter parameter is chosen with $L = 22$ [66]. Figure 2.5 depicts the scaling with the used cepstral lifter. The behaviour of the depicted lifter resembles the frequency response of a pre-emphasis filter, as it amplifies mid and high-range cepstral coefficients.

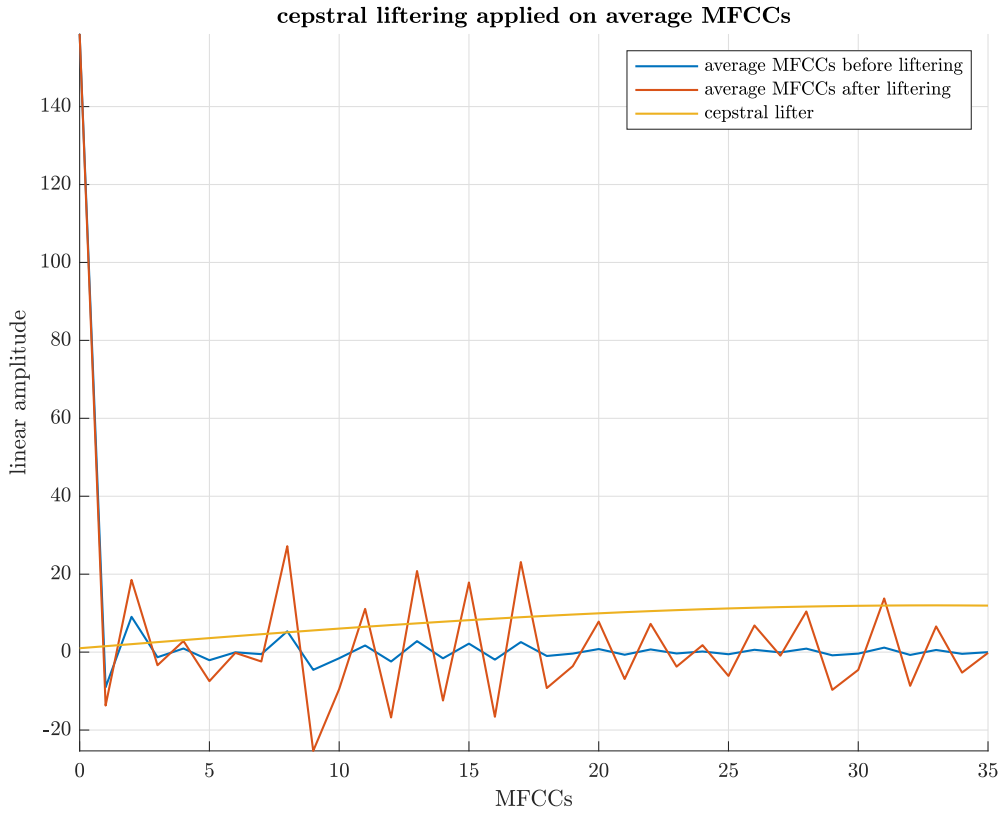


Figure 2.5 *Effects of cepstral liftering on exemplary MFCCs.*

2.3 Modulation Power Spectrum (MPS)

Similar to the underlying approach of the MFCCs, where a frequency spectrum is binned into frequency bands that are spaced in the same way that humans perceive pitch, is the idea of modulation based signal representations. Through extensive research carried out in the field of biology, it has been proven with the help of several psychophysical studies that the human auditory system is very sensitive towards temporal and spectral modulations [56]. Thus, a joint temporal and spectral modulation based signal representation of acoustic signals is apperent. In [56] the modulation power spectrum (MPS) is introduced and used to analyze natural sounds. The usage of the MPS in regards to sound manipulation has been thoroughly researched in [44]. The calculation of the MPS is based on a time frequency signal representation. In [56], the autocorrelation matrix is used as the underlying time frequency signal representation, whereas in [44] the MPS is directly derived from a spectrogram. In this thesis the calculation is kept in line with [44] and the spectrogram is used to calculate the modulation power spectrum.

The discrete time frequency representation, spectrogram $\hat{\boldsymbol{X}}[m,k]$ of a signal $s[n]$, assuming an infinitely long signal, is mathematically denoted as:

$$\hat{\boldsymbol{X}}[m,k] = \sum_{n=-\infty}^{\infty} s[n]w[n-m]e^{-jkn} = \mathcal{F}_{n\mapsto k}\left\{s[n]w[n-m]\right\}[m,k] \tag{2.10}$$

The calculation of a spectrogram denoted in Equation 2.10, can be seen as the Fourier transform of signal blocks that are windowed with the time window function $w[n-m]$, whereas m is the time variable denoting the current signal block [44, p.14]. The squared spectrogram in decibels is written as:

$$\tilde{\boldsymbol{X}}[m,k] = 10 \cdot \log_{10}\left(\left|\hat{\boldsymbol{X}}[m,k]\right|^2\right) = 20 \cdot \log_{10}\left(\left|\hat{\boldsymbol{X}}[m,k]\right|\right) \tag{2.11}$$

The MPS $\hat{\boldsymbol{S}}(f_{t_{\text{mod}}},\tau)$ is then calculated as the 2D-Fourier transform of the squared spectrogram in decibels $\tilde{\boldsymbol{X}}[m,k]$ [44]. The twodimensional Fourier transform dissects the spectrogram image into gratings that correspond to so-called *ripple sounds* [56]. These *ripple sounds* build the basis of the analyzed image and can be viewed as building blocks that, when put together, result in the underlying spectrogram image. Similar to sinusoidal components obtained from an onedimensional Fourier transform applied onto a audio sample, the *ripple sounds* show sinusoidal amplitude modulations in time and frequency [56]. Equation 2.12 shows the calculation of an MPS based on the 2D-Fourier transform analytically formulated as well as a compact notation. For the compact notation the time and frequency variables m and k of a spectrogram with the dimensions $[M \times N]$ are summarized using the vector $\boldsymbol{\theta} = [m,k]^T$ and the modulation variables are summarized using $\boldsymbol{\omega} = [f_{t_{\text{mod}}},\tau]^T$. In this thesis only the magnitudes of the MPS are further analyzed.

$$\begin{aligned}
\hat{\boldsymbol{S}}(f_{t_{\text{mod}}},\tau) &= \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1}\sum_{k=0}^{N-1} \tilde{\boldsymbol{X}}[m,k] e^{-j2\pi\left(\frac{f_{t_{\text{mod}}}\cdot m}{M} + \frac{\tau\cdot k}{N}\right)} \\
\hat{\boldsymbol{S}}(f_{t_{\text{mod}}},\tau) &= \hat{\boldsymbol{S}}(\boldsymbol{\omega}) = \mathcal{F}^{\text{2D}}_{\boldsymbol{\theta}\mapsto\boldsymbol{\omega}}\Big\{\tilde{\boldsymbol{X}}[m,k]\Big\}[\boldsymbol{\omega}]
\end{aligned} \tag{2.12}$$

An easier interpretation of the 2D-Fourier transform is to imagine it as two classic Fourier transforms one along the time axis of the spectrogram and the other along the frequency axis [44]. Similar to the 1D-Fourier transform the 2D-Fourier transform results in a symmetric depiction, if a real valued image is processed. Thus the resulting MPS is mirrored along the spectral modulation axis at $\tau = 0$. Negative spectral modulations can be discarded. An exemplary MPS created from a sung vocal sample and the modulation range, used in the analysis is illustrated in Figure 2.6 on the right.

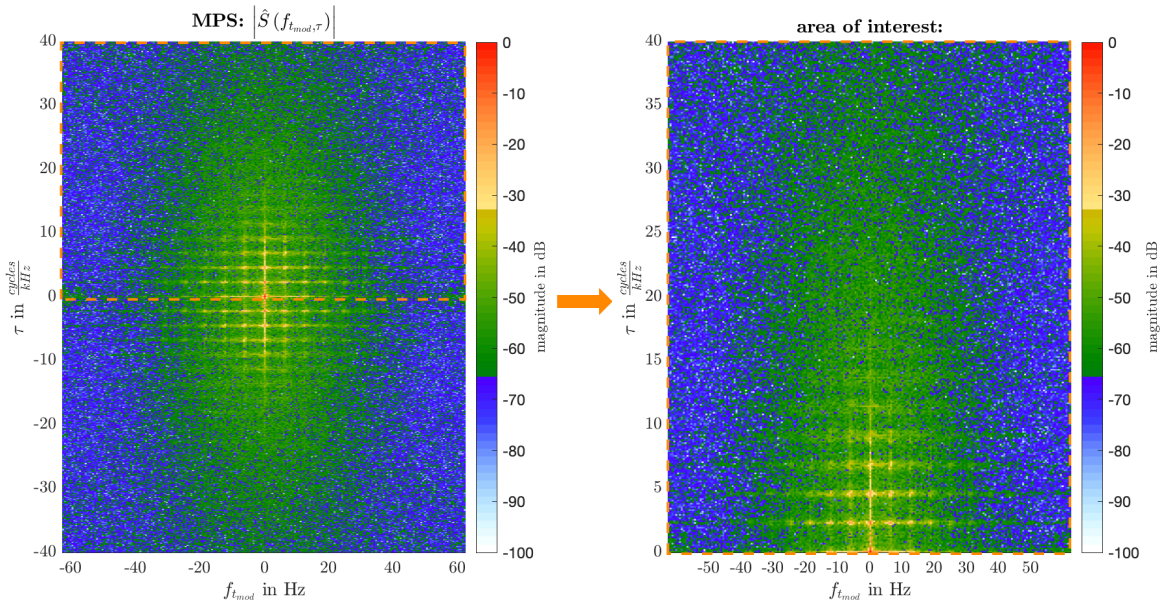


Figure 2.6 *Full modulation power spectrum and used modulation range.*

Temporal modulations

The x-axis of a MPS depicts the temporal modulations denoted with $f_{t_{\text{mod}}}$ present in a signal. The temporal modulations of a signal comprise the amplitude modulation (AM) and frequency modulation (FM). If either of them are present in a signal, vertical lines located at the respective modulation frequency $f_{t_{\text{mod}}}$ in the MPS become visible. The signal behind the MPS depicted in Figure 2.6 is a real sung vocal sample, with a distinctly perceivable vibrato. As shown in the work of Sciri in [54], the vibrato can be theoretically separated into shimmer, which in technical terms is an amplitude modulation and the jitter which corresponds to a frequency modulation. For instance, the first vertical line in Figure 2.6, indicating the vibrato frequency comprising AM and FM is located at ca. 6 Hz.

Spectral modulations

On the y-axis of a MPS the spectral modulations τ are projected. The spectral modulations are not to be mistaken with the frequency modulation. When imagining the 2D-Fourier transform as a Fourier transform along the time and frequency axis, the temporal modulations are calculated with the Fourier transform that is executed along the time axis. The spectral modulations, on the other hand, are derived from the Fourier transform along the frequency axis. This means that the spectral modulations describe the composition of the harmonics/overtones in a signal. In [44] it is stated that the spectral modulations can be interpreted as the number of harmonics contained in a kilohertz. If the integer multiple harmonics that are present in a sung vocal signal are viewed as a new signal which is then subject to a Fourier transform, it is evident that the whole-number multiples also describe a periodic structure and, thus, peaks at the spectral modulations which correspond to the period of the respective harmonics are formed [44, p.7-8]. As the spectral modulations are given as a Fourier transform of the frequency spectrum for each time slice of the spectrogram, the spectral modulation domain corresponds to the cepstral domain and the unit for the spectral modulations coincides with the *quefrency* unit in the cepstral domain. Reoccuring peaks along the spectral modulation axis can therefore also be viewed as *rhamonics*, the cepstral equivalent to harmonics [44].

Interpretation of a MPS

Figure 2.7, taken from [44, p.41] displays how the modulation power spectrum derived from a spectrogram is to be interpreted. At the top of Figure 2.7 a exemplary spectrogram of a sung vowel by a bariton is depicted. The parts of the spectrogram where no vibrato and therefore, no temporal modulation is present in the signal, which is given for the time range of $0\,\mathrm{s} \leq t \leq 0.6\,\mathrm{s}$ in Figure 2.7, is located at at the temporal modulations of $f_{t_{\mathrm{mod}}} = 0\,\mathrm{Hz}$. Upsweeps indicated by increasing frequencies are located in the left half ($f_{t_{\mathrm{mod}}} < 0$) of the MPS. Stronger upsweeps, where the frequency increases more rapidly, are located further on the left. Downsweeps are located in the positive temporal modulation half ($f_{t_{\mathrm{mod}}} > 0$). A special circumstance that is present in the MPS for vocal signals (speech or singing) is the triangular-shaped form located at lower modulations, marked at the bottom of the MPS visualized in Figure 2.7. This triangular shape is accredited to the vocal tract, which introduces low modulations into a signal [44], but the explicit relation between the vocal tract and the form or distinction of the triangular area, still remains unclarified to the knowledge of this thesis' author [44].

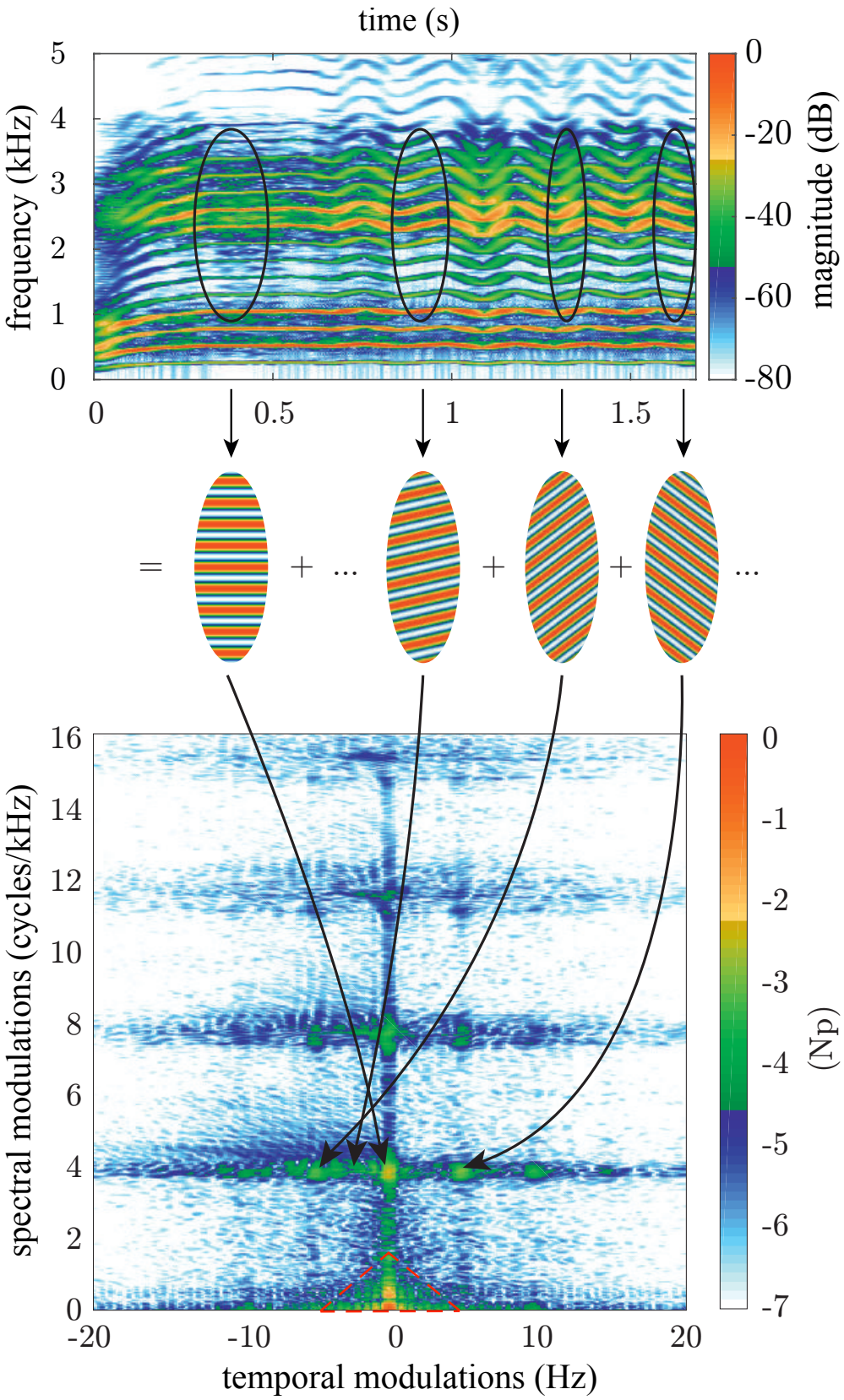


Figure 2.7 *Exemplary spectrogram and corresponding modulation power spectrum with locations of up and down sweeps within the MPS. Source: [44, p.41]*

The analysis based on the MPS, deals with the calculation of features and the aim to investigating potential correlations between calculated features and the voice quality. Thus, the dimensionality of the MPS is reduced, by summing the MPS along the temporal modulation axis which results in the summed temporal modulation power spectrum (STMPS) $\hat{\boldsymbol{S}}_{\Sigma}(f_{\mathrm{t_{mod}}})$, denoted in Equation 2.13. By summing along the spectral modulation axis, the summed spectral modulation power spectrum (SSMPS) $\hat{\boldsymbol{S}}_{\Sigma}(\tau)$, denoted in Equation 2.14 is calculated. In Figure 2.8 the calculation and resulting STMPS and SSMPS are visualized as examples. The feature extraction from the STMPS and the SSMPS is elaborated in subsection 3.4.3.

$$\hat{\boldsymbol{S}}_{\Sigma}(f_{\mathrm{t_{mod}}}) = \sum_{\tau=-\infty}^{\infty} \left|\hat{\boldsymbol{S}}(f_{\mathrm{t_{mod}}}, \tau)\right| \qquad (2.13)$$

$$\hat{\boldsymbol{S}}_{\Sigma}(\tau) = \sum_{f_{\mathrm{t_{mod}}}=-\infty}^{\infty} \left|\hat{\boldsymbol{S}}(f_{\mathrm{t_{mod}}}, \tau)\right| \qquad (2.14)$$

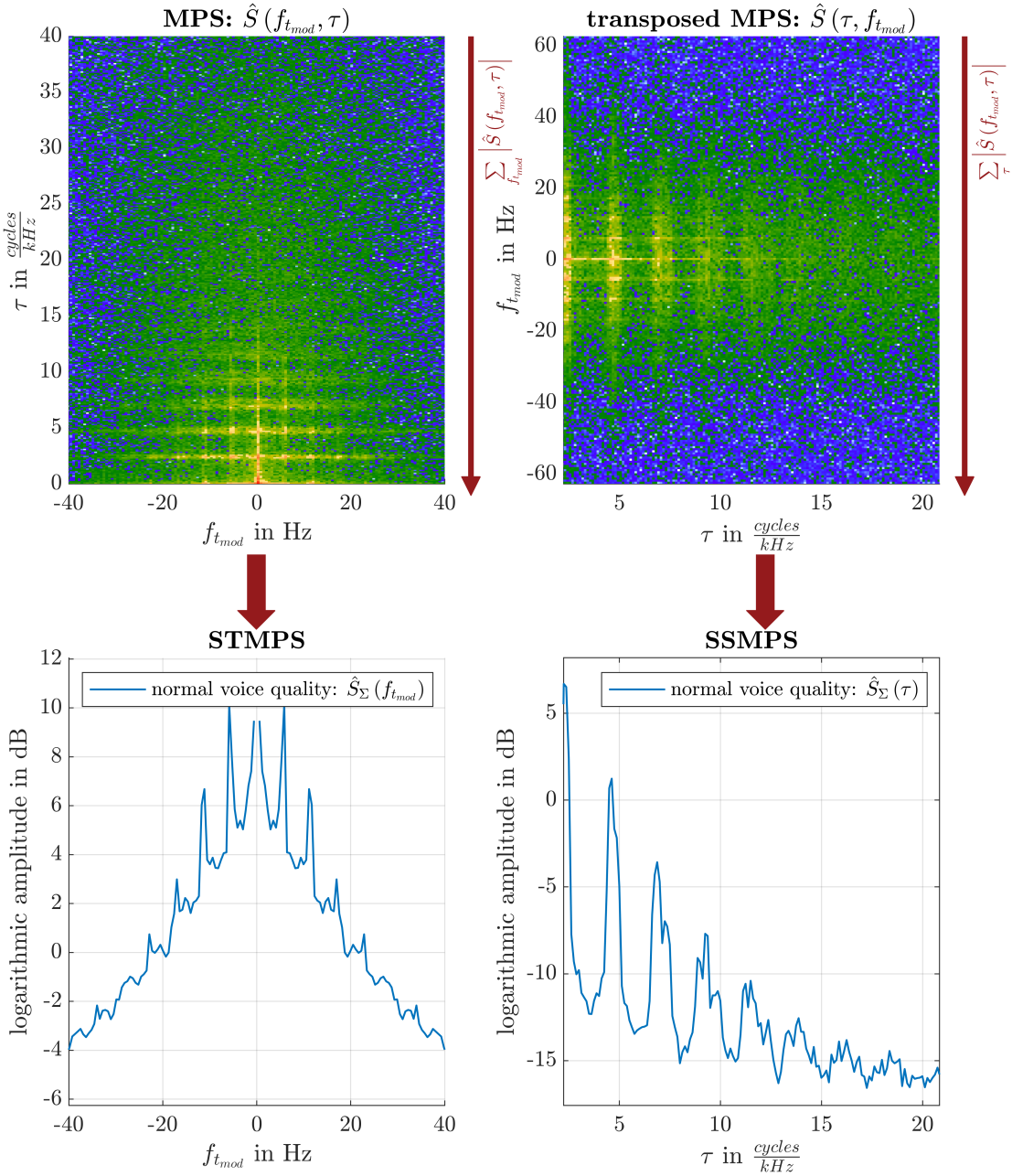


Figure 2.8 *Exemplary depiction on the calculation of the summed modulation power spectrum.*

2.4 Fundamental Frequency

The estimation of the fundamental frequency is necessary for the feature extraction mentioned in section 3.4. The estimation of the analyzed sung vocal signal's fundamental frequency is obtained using the SRH-pitch tracker proposed in [9], which is based on the summation of a residual signal's harmonics (SRH). The residual signal is calculated by glottal inverse filtering with a roughly estimated all-pole vocal tract filter. As Kraxberger *et al.* already discussed the limitations and thorough analysis of this pitch tracking algorithm in [4], a detailled description of the fundamental frequency estimation using the SRH method is not added to this thesis.

2.5 Support Vector Machine (SVM)

As discussed in section 2.1 the classifier used to process the descriptive features and execute the classification in this thesis are support vector machines (SVMs). Support vector machines are by default binary classifiers, meaning they are only able to separate two classes. Thus, a theoretical explanation of a SVMs labelled feature set $\mathscr{X}$ consisting of data samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, that comprise class descriptive features, e.g. the phonation type features used in this thesis, and binary feature labels, also called *targets* t_n, are presupposed. Hence, the labelled dataset can be written as a set of tuples:

$$\mathscr{X} = \{\langle \boldsymbol{x}_1, t_1 \rangle, \ldots, \langle \boldsymbol{x}_{N_{\text{feat}}}, t_{N_{\text{feat}}} \rangle\}, \quad \boldsymbol{x}_n \in \mathbb{R}^D, \quad t_n \in \{-1, 1\} \tag{2.15}$$

The idea behind SVMs is a two class classification problem employing a linear model. With it the data samples $\boldsymbol{x}_n$ located in the D-dimensional input space are projected onto a $(D-1)$-dimensional linear subspace, called the decision surface. This projection yields a signed measure of the perpendicular distance of the data samples $\boldsymbol{x}_n$ to the decision surface, given by the value of $y(\boldsymbol{x}_n)$ [5, p.182]. Therefore, points located on the decision surface are characterized by $y(\boldsymbol{x}_n) = 0$, which is the defining relation for the so-called decision boundary [5, p.182]. The orientation of the decision boundary is fixed by the vector $\boldsymbol{w}$, which is orthogonal to every vector lying within the decision surface and a bias term b is added, which shifts the surface [5, p.324]. It is assumed that the decision surface separates the two classes linearily and thus, the linear model $y(\boldsymbol{x}_n)$ can be written as denoted in Equation 2.18.

$$y(\boldsymbol{x}_n) = \boldsymbol{w}^T \boldsymbol{x}_n + b \tag{2.16}$$

Binary target values/labels $\hat{t}_n$ are then estimated by checking the sign of $y(\boldsymbol{x}_n)$ using the signum function as denoted in Equation 2.17. In machine learning terms the signum function, in this context, represents a so-called *activation* function.

$$\hat{t}_n = \text{sgn}(y(\boldsymbol{x}_n)) \tag{2.17}$$

Naturally, it can't be assumed that all data is linearly separable. Thus, a non-linear feature-space transformation denoted with $\phi(\cdot)$ is applied to the data. Hence, Equation 2.18 becomes:

$$y(\boldsymbol{x}_n) = \boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b \tag{2.18}$$

The idea of the non-linear feature space transformation is to transform the data into a higher dimensional space where they are linearly separable. As this feature space transformation is done with so-called *kernel functions*, this is also referred to as the *kernel-trick* [5, p.292] which is discussed in subsection 2.5.2. For the model described in Equation 2.18 this means that linear separability can again be assumed.

2.5.1 Margin Maximization

If a feature set is now linearly separable there still remains the question as to which of the possible decision boundary options is chosen to separate the data. The decision boundary is defined by

$$y(\boldsymbol{x}_n) = \boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b \stackrel{!}{=} 0 \tag{2.19}$$

as the vector $\boldsymbol{w}$, which spans the subspace on which the data is projected on, is perpendicular to the decision boundary. An academic example of problem is depicted in Figure 2.9 where three exemplary options for the decision boundary are illustrated.

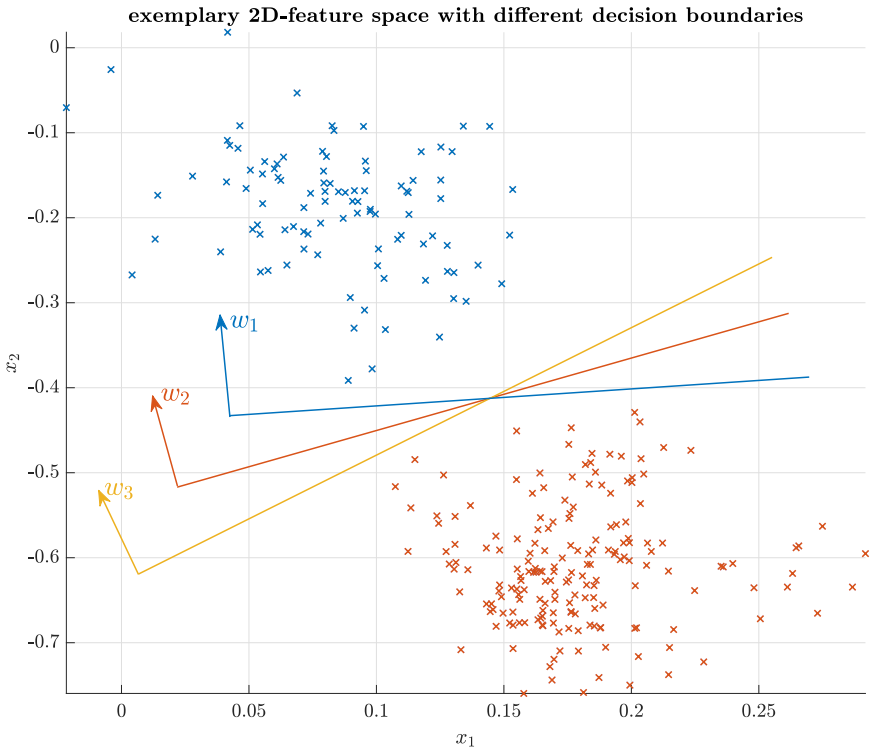


Figure 2.9 *Academic example of linear separable data with various decision boundaries.*

Each of the three boundaries in Figure 2.9 separate the data. In order to choose the optimal decision boundary an optimization strategy is necessary. Thus, the term *margin* is introduced. The margin is given "[...]as the perpendicular distance between the decision boundary and the closest of data points[...]" [5, p.327] and is visualized in Figure 2.10. The idea now is to maximize the margin in order to be able to choose the optimal decision boundary. To do so, the distance d_n of a point $\boldsymbol{x}_n$ to the decision boundary is defined [5, p.327]:

$$d_n(\boldsymbol{x}_n) = \frac{y(\boldsymbol{x}_n)}{||\boldsymbol{w}||} = \frac{\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b}{||\boldsymbol{w}||} \tag{2.20}$$

The points closest to the decision boundary are called support vectors $\boldsymbol{x}_{\text{SV}}$ and their target value is either $t_n = -1$ or $t_n = 1$. The distance of the support to the decision boundary is given with:

$$d_n(\boldsymbol{x}_{\text{SV}}) = \min_n [t_n d_n] \tag{2.21}$$

Thus, the margin can be optimized by maximizing the support vectors' distance towards the decision boundary with respect to the model parameters $\boldsymbol{w}$ and b. The maximum margin solution is obtained if Equation 2.22 is solved [5, p.327]

$$\underset{\boldsymbol{w},b}{\operatorname{argmax}} \{d_n(\boldsymbol{x}_{\mathrm{SV}})\} = \underset{\boldsymbol{w},b}{\operatorname{argmax}} \left\{ \min_n [t_n d_n] \right\} = \underset{\boldsymbol{w},b}{\operatorname{argmax}} \left\{ \min_n \left[t_n \frac{\left(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b \right)}{||\boldsymbol{w}||} \right] \right\} \tag{2.22}$$

As the term $\frac{1}{||\boldsymbol{w}||}$ is not dependent on n it can be pulled out of the min-operator and the maximum margin objective becomes [5, p.327]:

$$\underset{\boldsymbol{w},b}{\operatorname{argmax}} \left\{ \frac{1}{||\boldsymbol{w}||} \min_n \left[t_n \left(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b \right) \right] \right\} \tag{2.23}$$

It is now possible to set up the following constraints:

$$\begin{aligned} t_n \left(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b \right) &\geq \min_n \left[t_n \left(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b \right) \right] \\ \min_n \left[t_n \left(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b \right) \right] &\stackrel{!}{=} 1 \end{aligned} \tag{2.24}$$

This is possible because the term $t_n \boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b$ is always larger than the same term under the influence of the min operator and additionally it is possible to scale the term $\min_n \left(t_n \boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b \right)$ so that it equals 1 [5, p.328]. Therefore, the margin objective can again be reformulated into:

$$\underset{\boldsymbol{w},b}{\operatorname{argmax}} \left\{ \frac{1}{||\boldsymbol{w}||} \right\} \quad : t_n \left(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b \right) \geq 1 \quad \forall \quad n = 1, \ldots, N \tag{2.25}$$

Maximizing $\frac{1}{||\boldsymbol{w}||}$ is equal to minimizing $\frac{1}{2}||\boldsymbol{w}||^2$. Thus the final form of the margin objective is given with:

$$\underset{\boldsymbol{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2}||\boldsymbol{w}||^2 \right\} \quad : \quad t_n \left(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b \right) - 1 \geq 0 \quad \forall \quad n = 1, \ldots, N \tag{2.26}$$

An exemplary feature space containing linearily separable data, separated by the decision boundary which is defined by margin on which the support vectors are located is summarized in Figure 2.10.

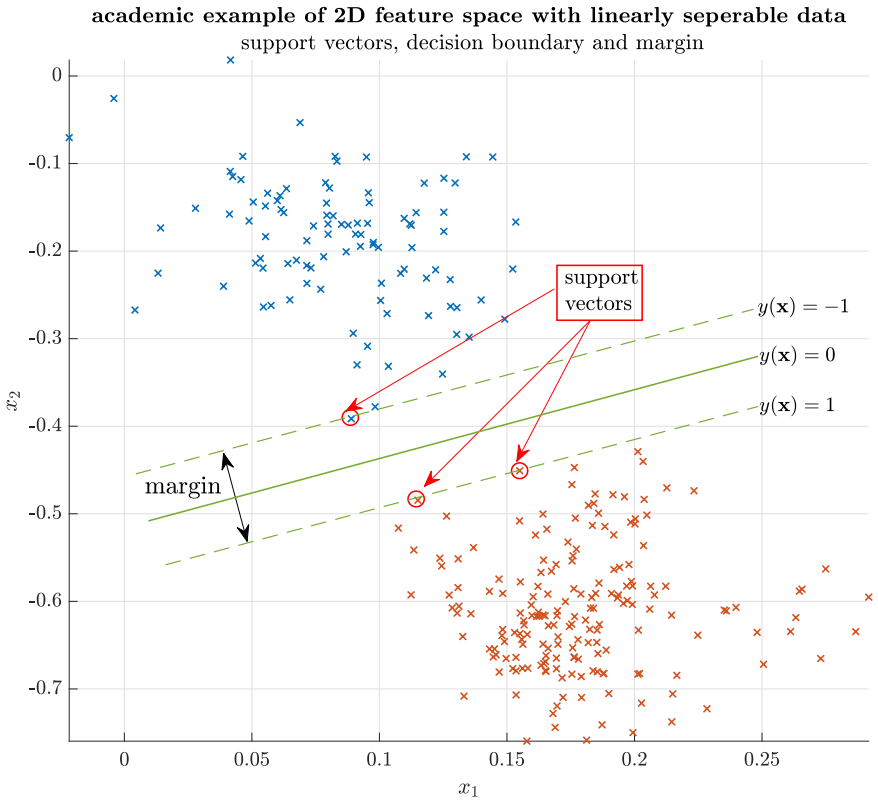


Figure 2.10 *Academic example margin, support vector and decision boundary definition. Source: [5, p.327]*

2.5.2 Non-Linear Feature Space Transformation using Kernel Functions

The margin objective mentioned in Equation 2.26 is a *convex optimization* or *quadratic programming* problem which is solvable with the so-called Lagrangian approach [5, p.328]. The Lagrangian approach utilizes Lagrangian multipliers $\boldsymbol{\alpha} = [\alpha_0, \ldots, \alpha_N]^T$ and is given as:

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{n=1}^{N} \alpha_n \left(t_n \left(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b \right) - 1 \right) \tag{2.27}$$

The optimum solution can now be calcluated by forming the derivatives of $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ with respect to $\boldsymbol{w}$ and b and setting them to zero [5, p.328]:

$$\begin{aligned} \frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial \boldsymbol{w}} &= \boldsymbol{w} - \sum_{n=1}^{N} \alpha_n t_n \phi(\boldsymbol{x}_n) \stackrel{!}{=} 0 \Rightarrow \boldsymbol{w} = \sum_{n=1}^{N} \alpha_n t_n \phi(\boldsymbol{x}_n) \\ \frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial b} &= \sum_{n=1}^{N} \alpha_n t_n \stackrel{!}{=} 0 \Rightarrow \sum_{n=1}^{N} \alpha_n t_n = 0 \end{aligned} \tag{2.28}$$

The conditions formulated in Equation 2.28 can be applied into Equation 2.27, where $\boldsymbol{w}$ and b have been eliminated due to the derivation. This yields $\tilde{L}(\alpha)$, the dual representation of the maximum margin problem, which introduces the already briefly mentioned kernel functions $k(\boldsymbol{x}_n, \boldsymbol{x}_m)$ into the

dual representation [5, p.329]. The

$$\tilde{L}(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m t_n t_m \underbrace{\phi(\boldsymbol{x}_n)^T \phi(\boldsymbol{x}_m)}_{k(\boldsymbol{x}_n, \boldsymbol{x}_m)} \tag{2.29}$$

The dual representation $\tilde{L}(\boldsymbol{\alpha})$ is then maximized in regards to $\boldsymbol{\alpha}$ under the constraints:

$$\begin{aligned} \alpha_n &\geq 0, \quad \forall\, n = 1, \ldots, N \\ \sum_{n=1}^{N} \alpha_n t_n &= 0 \end{aligned} \tag{2.30}$$

As already mentioned, the aim of the usage of kernel functions is to search for a higher dimensional space. In this higher dimensional space the data is assumed to be linearly separable. Thus, using the Lagrangian approach and the kernel function $k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \phi(\boldsymbol{x}_n)^T \phi(\boldsymbol{x}_m)$ new data is now still classifiable by observing the sign of $y(\boldsymbol{x}_n)$. However, the initial form of Equation 2.19 can be reformulated into:

$$y(\boldsymbol{x}_n) = \boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b = \sum_{m=1}^{N} \alpha_m t_m k(\boldsymbol{x}_m, \boldsymbol{x}_n) + b \tag{2.31}$$

There are several possible kernel functions to choose from, but the kernel used in this thesis is the radial basis function kernel or also called Gaussian kernel, which is given with Equation 2.32.

$$k(\boldsymbol{x}_m, \boldsymbol{x}_n) = e^{-\frac{1}{2\sigma^2} ||\boldsymbol{x}_n - \boldsymbol{x}_m||^2} \tag{2.32}$$

In this context the kernel funcion is not to be interpreted as a probability density function. Thus, the parameter σ^2 in Equation 2.32 is no statistical variance, it is simply viewed as a hyperparameter which controls the width of the kernel [5, p.297]. Equation 2.32 is often also denoted with:

$$k(\boldsymbol{x}_m, \boldsymbol{x}_n) = e^{-\gamma ||\boldsymbol{x}_n - \boldsymbol{x}_m||^2} \tag{2.33}$$

, where the term $\frac{1}{2\sigma^2}$ is replaced with γ. This notation is used in the software application `Matlab`, where the hyperparameter γ is referred to as the *kernel scale*.

2.5.3 Penalization of Non-Separable Data

In the dual representation shown in Equation 2.29, the assumption that the data is completely linearily separable is still presupposed. This is often also referred to as a hard margin constraint, meaning no points are located inside the margin and all data points are correctly classifiable. Nonetheless, this is often not the case, so in order to deal with data samples that lie within the margin or on the wrong side of the decision boundary the SVM has to be modified to the extent that some misclassifications are allowed [5]. In order to do so so-called slack variables $\xi_n \geq 0$ for each data point are introduced. The slack variables are defined as follows:

$$\xi_n = \begin{cases} 0, & \forall\, n \;:\; y(\boldsymbol{x_n}) = \{-1, 1\} \\ |t_n - y(\boldsymbol{x}_n)|, & \text{else} \end{cases} \tag{2.34}$$

This means that if the data samples are on the margin the slack variable equals $\xi_n = 0$ and otherwise the variable equals the distance towards the correct side of the margin. Thus a data point on the decision boundary has a slack variable of $\xi_n = 1$, data points with $\xi_n < 1$ lie on the correct side of

the decision boundary, but within the margin area, and data points with $\xi_n > 1$ are misclassified [5]. An exemplary depiction on how the slack variables are defined is visualized in Figure 2.11.

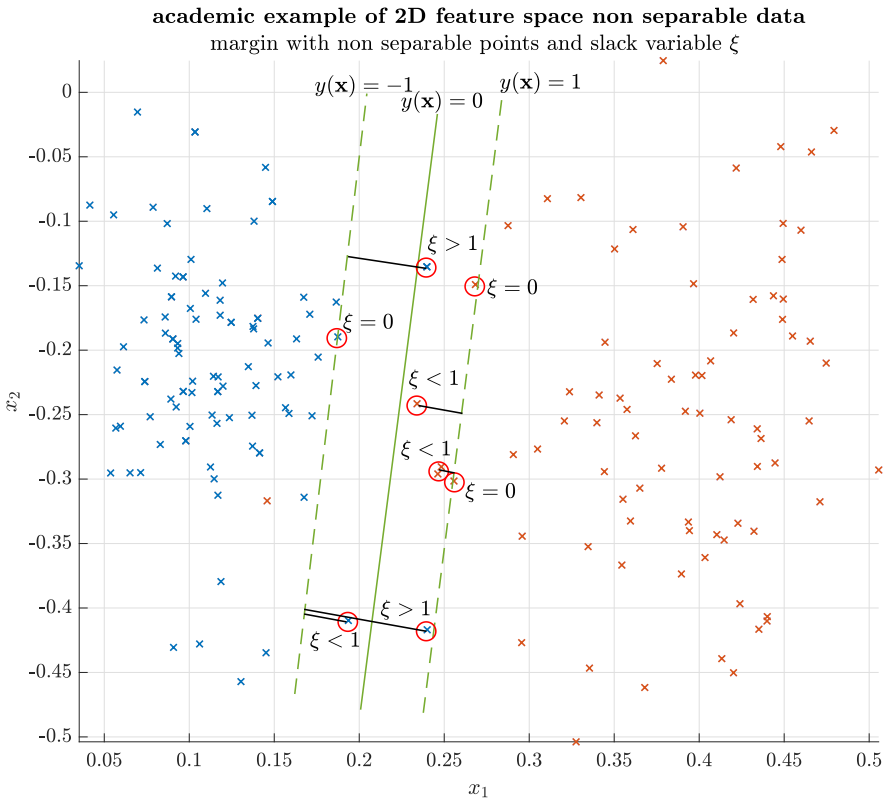


Figure 2.11 *Academic example on margin with non separable data and slack variable ξ, which penalizes the distance of support vectors that lie within the margin, support vectors are circled in red. Source: [5, p.332]*

Introducing the slack variables ξ_n reformulates the margin objective and constraints set in Equation 2.26 such that:

$$\underset{\boldsymbol{w},b,\boldsymbol{\xi}}{\operatorname{argmin}} \left\{ C \sum_{n=1}^{N} \xi_n + \frac{1}{2}||w||^2 \right\} \quad : \quad t_n \left(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b \right) \geq 1 - \xi_n \quad \forall \quad n = 1, \ldots, N \tag{2.35}$$

The introduced parameter $C \geq 0$ controls the trade-off between the penalizing slack variables ξ_n and the margin [5, p.332]. When following the objective formulated in Equation 2.35 the hard margin is relaxed and the goal of the margin objective from Equation 2.26, is now to maximze the margin whilst points that lie inside the margin are softly penalized. This is the reason why this is also often referred to as a *soft margin* SVM [5]. The maximization of the soft margin using the objective of Equation 2.35 is again solvable with Lagrangian multipliers as done in Equation 2.27, with the only difference that now a third partial derivative with respect to $\partial\xi_n$ has to be added to Equation 2.28, before the dual representation can be formulated. In the end, the dual Lagragian for soft margin classification is of the same form as in Equation 2.29 [5, p.333]. However, the first constraint of the maximization with regards to $\tilde{L}(\boldsymbol{\alpha})$ denoted in Equation 2.30 changes to:

$$0 \leq \alpha_n \leq C, \quad \forall\, n = 1, \ldots, N \tag{2.36}$$

The constraint is also known as box constraint and is controlled through the parameter C, if C is chosen as a very large value, the margin becomes harder and for $C \to \infty$ a hard margin classification with

the inital objective from Equation 2.26 is obtained. However, if the data is non-separable an increase of C leads to a SVM classifier that will be overfitted. Higher values of C suppress misclassifications within the training process and the SVM class boundaries are adapted specifically towards the training data, which is followed by poor generalization towards new data as visualized in Figure 2.1.

2.6 Plus-L Minus-R Feature Selection

In order to avoid *the curse of dimensionality* (see section 2.1) it is important to analyze the present feature space for overfitting with regards to feature space dimensionality. This is done with the Plus-L Minus-R Feature selection algorithm (L-R selection). Its aim of is to reduce the dataset to a smaller presupposed number of samples. However, some features are more informative than others and thus the L-R selection incooperates a more sophisticated process, rather than just randomly selecting fewer features. The Plus-L Minus-R feature selection algorithm consists of two steps. The first step is the L-times execution of a sequential forward selection (SFS) and the second step the R-times execution of the sequential backward selection (SBS) [63]. If $L > R$ the algorithm starts with an empty feature set, in which L features are added, using SFS and then the R worst features are discarded using SBS until the required number of features is achieved. If $L < R$ the algorithm starts with a full feature set for which R features are removed with SBS and then L features are added with SFS. In comparison to SFS and SBS the L-R selection overcomes the problem of *nesting*, which means that selected features that have been chosen once cannot be removed. [63, p.316].

Sequential forward selection (SFS)

The sequential forward selection starts with an empty feature set $\mathscr{Y}_0 = \{\emptyset\}$. For each iteration k a feature $\tilde{x}$ from the full feature set $\mathscr{X}$ is selected according to:

$$\tilde{x} = \underset{x \in \mathscr{X} \setminus \mathscr{Y}_k}{\operatorname{argmax}} \left\{ J\left(\mathscr{Y}_k + x\right) \right\} \tag{2.37}$$

and added to the subset of selected features $\mathscr{Y}_k$:

$$\mathscr{Y}_{k+1} = \mathscr{Y}_k + \tilde{x} \tag{2.38}$$

, where $J\left(\mathscr{Y}_k + x\right)$ is a selection criterion that is evaluated for each iteration and $x \in \mathscr{X} \setminus \mathscr{Y}_k$ denotes that the selected features belong to the subset of features in $\mathscr{X}$, that have not been selected yet. SFS chooses the features that contribute most towards the selection criterion, if they are added to the feature selection subset $\mathscr{Y}_k$, until the required number of features is reached. This bears the disadvantage that once features have been selected the selection is irreversible (*nesting*) [63, p.315].

Sequential backward selection (SBS)

In contrast to the SFS, the sequential backward selection (SBS) starts with a full dataset $\mathscr{Y}_0 = \mathscr{X}$. In each iteration the features are selected according to:

$$\tilde{x} = \underset{x \in \mathscr{Y}_k}{\operatorname{argmax}} \left\{ J\left(\mathscr{Y}_k - x\right) \right\} \tag{2.39}$$

and discarded from the subset of selected features $\mathscr{Y}_k$:

$$\mathscr{Y}_{k+1} = \mathscr{Y}_k - \tilde{x} \tag{2.40}$$

Again the selection criterion $J(\mathscr{Y}_k - x)$ is evaluated for each iteration and the features which contribute least to the criterion are discarded until the required number of features is achieved. If SBS is used alone, it has the disadvantage that once a feature is discarded, it cannot be added back to the feature selection and also the selection criterion has to be evaluated over a larger number of features, as it starts with a full feature set which is gradually diminished [63, p.315].

2.6.1 Class Separability Measure

There are several possibilities to choose the selection criterion mentioned in section 2.6, which is crucial to the execution of the L-R selection. The chosen class separability measure is based on the Fisher criterion for multiple classes, originating from the linear discriminant analysis (LDA) [5]. The Fisher criterion of multiple classes is given with:

$$J(\boldsymbol{W}) = \frac{\boldsymbol{W}^T \boldsymbol{S}_B \boldsymbol{W}}{\boldsymbol{W}^T \boldsymbol{S}_W \boldsymbol{W}} \tag{2.41}$$

where $\boldsymbol{S}_B$ is the *between-class* covariance/scatter matrix, $\boldsymbol{S}_W$ is the *within-class* covariance/scatter matrix and if $J(\boldsymbol{W})$ is viewed from the LDA perspective, $\boldsymbol{W}$ holds projection vectors in its columns. In case of LDA these projection vectors are used to project the data to a lower dimensional space (dimensionality reduction), whilst maintaining the best possible separability between classes [5, p.189-191]. However, for the usage of the Fisher criterion as a separability measure in feature selection, $\boldsymbol{W}$ can be understood as the subset of features, for which the selection criterion is evaluated during the L-R selection. To calculate the *between-class* and *within-class* scatter matrices, the global covariance matrix $\boldsymbol{\Sigma}$ defined through the global mean $\boldsymbol{\mu} = \frac{1}{N} \sum_{n=1}^{N} x_n$ is necessary [63, p.375].

$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{n=1}^{N} (\boldsymbol{x}_n - \boldsymbol{\mu}) (\boldsymbol{x}_n - \boldsymbol{\mu})^T \tag{2.42}$$

With N number of observations/data samples and M classes the *within-class* scatter matrix $\boldsymbol{S}_W$ is defined with [63, p.375]:

$$\begin{aligned} \boldsymbol{S}_W &= \frac{1}{N} \sum_{m=1}^{M} \sum_{n=1}^{N} z_{mn} (\boldsymbol{x}_n - \boldsymbol{\mu}_m) (\boldsymbol{x}_n - \boldsymbol{\mu}_m)^T \\ z_{mn} &= \begin{cases} 1, & t_n = t_m \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{2.43}$$

The function z_{mn} distinguishes if the class of $\boldsymbol{x}_n$ (defined by its target value t_n) belongs to the the class t_m, $\boldsymbol{\mu}_m = \frac{1}{M} \sum_{n=1}^{N} z_{nm} \boldsymbol{x}_n$ is the m^{th} class' mean and $N_m = \sum_{n=1}^{N} z_{mn}$ is the number of samples in the m^{th} class. The *between class* scatter matrix $\boldsymbol{S}_B$ is then calculated using Equation 2.42 and Equation 2.43 [63, p.375].

$$\boldsymbol{S}_B = \boldsymbol{\Sigma} - \boldsymbol{S}_W = \sum_{m=1}^{M} \frac{N_m}{N} (\boldsymbol{\mu}_m - \boldsymbol{\mu}) (\boldsymbol{\mu}_m - \boldsymbol{\mu})^T \tag{2.44}$$

The main idea behind the Fisher Criterion noted in Equation 2.41 is that good separability is given, if the between-class covariance is large and the within-class covariance is small, which is not only represented by the matrix product in Equation 2.41 but also true for the trace[1] ratio of the corresponding matrices as proven in [46]. Thus, the selection criterion from Equation 2.41 is replaceable with:

$$J(\boldsymbol{W}) = \frac{\text{trace}\left(\boldsymbol{W}^T \boldsymbol{S}_B \boldsymbol{W}\right)}{\text{trace}(\boldsymbol{W}^T \boldsymbol{S}_W \boldsymbol{W})} = \frac{\text{trace}\left(\tilde{\boldsymbol{S}}_B\right)}{\text{trace}\left(\tilde{\boldsymbol{S}}_W\right)} \tag{2.45}$$

The trace of $\tilde{\boldsymbol{S}}_W$ holds information on the average variance amongst all classes, and the trace of $\tilde{\boldsymbol{S}}_B$ holds information on the average distance towards the global mean $\boldsymbol{\mu}$. Because Equation 2.45 is used as the selection criterion during feature selection with the Plus-L Minus-R mentioned in section 2.6, the matrices $\tilde{\boldsymbol{S}}_W$ and $\tilde{\boldsymbol{S}}_B$ can be viewed as the scatter matrices of all data points which are represented by the subset of features, selected in the current L-R selection iteration.

[1] The trace of matrix is calculated as the sum of its main diagonal elements e.g. $\text{trace}(A) = \sum_i A_{ii}$ [49, p.6]

3 Analysis and Classification

The methods previously discussed in chapter 2 lay the foundation for the executed classification of a novel dataset, created at the Institute of Electronic Music and Acoustics at the University of Music and Performing Arts Graz. Before the classification and analysis of sung vocal signals with regards to the sung voice quality is discussed, a brief overview on the dataset, its creation and the classification possibilities that come with it, is given.

3.1 Dataset

The dataset underlying the analysis carried out in this thesis, consists of recored vocal samples of 10 different professional singers (6 female and 4 male singers). The singers were asked to sing a vowel in one of the voice qualities defined in subsection 1.2.2 (*normal*, *breathy*, or *pressed*). The sung vocals were recorded by use of of the spherical microphone array proposed in [6], which also allowed an approximation of the singers directivity pattern. Additionally, the mouth opening of the singers were measured with a tracking system proposed in [21]. As the focus of this thesis lies in the analysis and classification of phonation types, the measurements regarding the directivity and mouth opening are not relevant. Besides the voice quality and the vowel, the third instruction that was given was the pitch. The pitch range of the samples is depicted in Table 3.1 and the approximate frequencies of these pitches, assuming equal temperament, can be read from Table 3.1. The reference pitch is colored in orange. The instructions that were given to the singers were then evaluated in a listening experiments, in which experts rated the perceived voice quality and vowel.

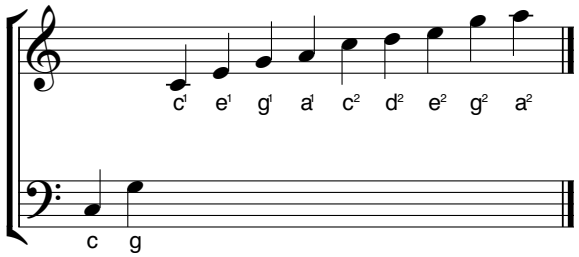


Figure 3.1 *Pitch range of samples in the database.*

Table 3.1 *Pitches and frequencies for equal temperament. Source: [55]*

pitch	frequency in Hz
c	130.813 Hz
g	195.998 Hz
c^1	261.626 Hz
e^1	329.628 Hz
g^1	391.995 Hz
a^1	440 Hz
c^2	523.251 Hz
d^2	587.330 Hz
e^2	659.226 Hz
g^2	783.991 Hz
a^2	880 Hz

The vowels the singers were instructed to sing are */a/*, */e/*, */i/*, */o/* or */u/*. The results of the listening experiment allows a distinction of the vowel with regard to primary and secondary cardinal vowels

as depicted in the International Phonetic Alphabet (IPA) Chart [3], but due to the fact that this thesis deals with the classification and analysis of phonation types no further distinction between the vowels are made and the 5 vowels are referred to as */a/*, */e/*, */i/*, */o/* or */u/*. If the instructions given to the singers are viewed as the groundtruth, the full dataset consists of 1140 samples. The distribution of the 1140 samples with respect to the voice-quality, singers, vowels and pitches are listed in Table 3.2 - 3.4. The singers' gender is marked in Table 3.2 with (f) for female and (m) for male singers.

Table 3.2 *Dataset distribution voice quality vs. singers.*

voice quality	**singer**										**total**
	S1 (f)	**S2** (f)	**S3** (m)	**S4** (f)	**S5** (f)	**S6** (f)	**S7** (m)	**S8** (m)	**S9** (m)	**S10** (f)	
normal	45	45	30	45	45	45	30	30	20	45	380
breathy	45	45	30	45	45	45	30	30	20	45	380
pressed	45	45	30	45	45	45	30	30	20	45	380
Σ	135	135	90	135	135	135	90	90	60	135	1140

Table 3.3 *Dataset distribution voice quality vs. vowels.*

voice quality	**vowel**					**total**
	/a/	***/e/***	***/i/***	***/o/***	***/u/***	
normal	76	76	76	76	76	380
breathy	76	76	76	76	76	380
pressed	76	76	76	76	76	380
Σ	228	228	228	228	228	1140

Table 3.4 *Dataset distribution voice quality vs. pitches.*

voice quality	**pitches**											**total**
	c	**g**	**c^1**	**e^1**	**g^1**	**a^1**	**c^2**	**d^2**	**e^2**	**g^2**	**a^2**	
normal	20	20	50	50	45	45	30	30	30	30	30	380
breathy	20	20	50	50	45	45	30	30	30	30	30	380
pressed	20	20	50	50	45	45	30	30	30	30	30	380
Σ	60	60	150	150	135	135	90	90	90	90	90	1140

As visible in the **total** columns of Table 3.2 - 3.4 the dataset is balanced with regard to the instructed voice quality. There are exactly 380 samples of each requested voice quality. With regard to the vowels the dataset is perfectly balanced for each voice quality. There are 76 samples per vowel. When looking at the distribution across the singers it is noticeable that singers S3, S7, S8 and S9 are the ones with the lowest number of samples in the database. Singers S3 and S7-9 are male singers, who due to their lower register are also the ones responsible for the samples at the lower pitches c

and g. The other singers were female singers. So it is worth noticing that the dataset consists of samples predominantly sung by female singers, which is an important aspect in regards to filling the gender gap that is present in scientific data, due to the fact that a lot of the fundamental research carried out in the second half of the 20^{th} century is based on data retrieved by male experimentees; e.g., the correlates proposed in [15] are all based on calculations that were executed based on samples recorded with a male speaker, which is, as shown in [67], not completely generalizeable, as there are gender specific aspects that have to be adressed especially in the context of speech signal processing. Returning to the composition of the processed dataset, it is visible in Table 3.4, that the fewest samples are given for the lowest pitches at around $130.813\,\mathrm{Hz}$ (c) and $195.998\,\mathrm{Hz}$ (g), whereas most samples are given for the pitches c^1 and e^1.
After recording the sung vocal samples, a listening experiment in which voice quality and vowels were rated by professionals, e.g. singers, professors and students of linguistic or engineers specialized in speech signal processing, was conducted. The voice quality rating was executed with the help of a continuous scale in which the listeners had to rate the perceived voice quality on a range from -1 (*breathy*) to 1 (*pressed*). In the listening experiment each recorded sample was rated multiple times to achieve statistical stability. Now the median voice quality ratings are computed and a k-mediod cluster analysis performed in `Matlab` using the `kmediods()` - command [34] is carried out. With the k-mediod analysis results the class boundaries of $N_{clust} = 3$ clusters were calculated, which allows a distinction of the relevant voice quality *breathy*, *normal* and *pressed*. By comparing the median voice quality rating of each sample to the class boundaries obtained from the k-mediod cluster analysis, class labels for the three phonation types are created and each sample can be labelled. Henceforth the labels retrieved from the listening experiment are called *experiment labels*, the voice quality that the singers were initially instructed with, during the recording process, are referred to as *instruction labels*. A comparison of the *instruction labels* with the *experiment labels* in form of a confusion matrix created with [27] is shown in Figure 3.2.

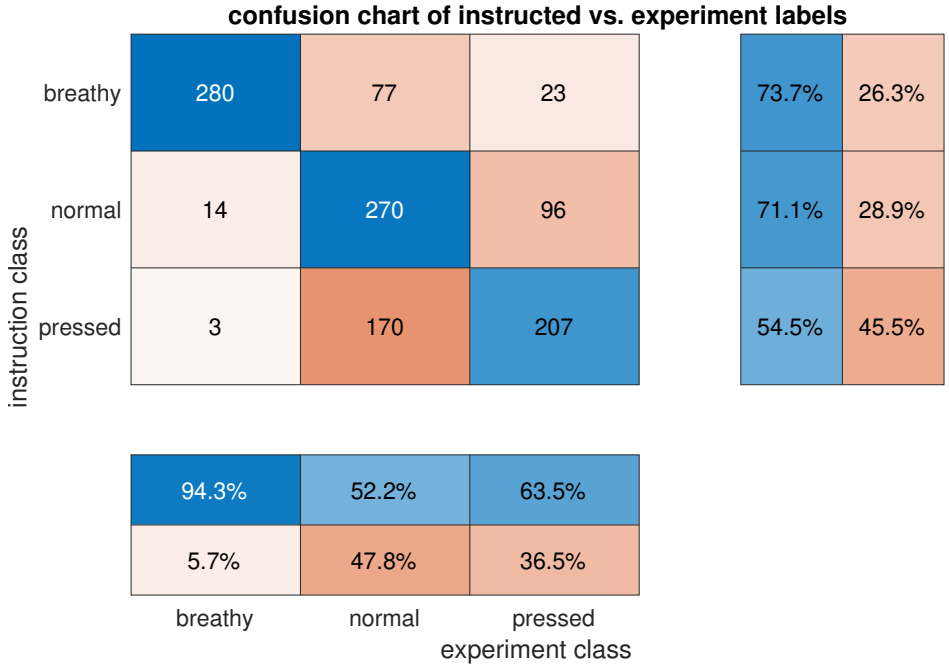


Figure 3.2 *Confusion matrix comparison of experiment labels and instruction labels.*

When looking at Figure 3.2 it becomes clear that the samples in which the singers were instructed to sing with *breathy* phonation were also the ones that were overwhelmingly ranked with the same voice quality. Most differences can be detected between the classes *normal* and *pressed*. 96 samples, where the singers were instructed to use *normal* voice quality were ranked with *pressed* voice quality and 170 samples where the singers were instructed to use *pressed* phonation are ranked as *normal* voice quality. The deviation between the ratings and the instructions becomes clearer when the column summary located at the bottom of Figure 3.2 is looked at. $73.7\,\%$ of the samples that were instructed with *breathy* were also ranked as *breathy*, $71.1\,\%$ of the samples where the singers were instructed to use *normal* phonation were also ranked to be of *normal* voice quality and $54.5\,\%$ of the instructed *pressed* samples were also ranked as *pressed* in the listening experiment. The row summaries on the right hand side of Figure 3.2 give information on how many percent of the *experiment labels* exhibit the same voice quality within the *instruction labels*, i.e. the most notable value is observable for *breathy* voice quality. Here $94.3\,\%$ of the *breathy experiment labels* are labelled as *breathy* within the *instruction labels*.

This means that when looking at the dataset with *experiment labels* the balance with regards to the number of samples for each voice-quality is skewed. When accumulating the number of samples per voice quality for the *experiment labels*, each class has the following number of samples:

Table 3.5 *Number of samples per class when using experiment labels.*

- *breathy* voice quality:	$N_{\mathrm{b}}^{\mathrm{exp}} = 297$ samples
- *normal* voice quality:	$N_{\mathrm{n}}^{\mathrm{exp}} = 517$ samples
- *pressed* voice quality:	$N_{\mathrm{p}}^{\mathrm{exp}} = 326$ samples
	$N^{\mathrm{exp}} = 1140$ samples

For the sake of completeness the sample distribution across the classes, when using *instruction* labels, is summed up in Table 3.6.

Table 3.6 *Number of samples per class when using instruction labels.*

- *breathy* voice quality:	$N_{\mathrm{b}}^{\mathrm{inst}} = 380$ samples
- *normal* voice quality:	$N_{\mathrm{n}}^{\mathrm{inst}} = 380$ samples
- *pressed* voice quality:	$N_{\mathrm{p}}^{\mathrm{inst}} = 380$ samples
	$N^{\mathrm{inst}} = 1140$ samples

The two sets of voice quality labels and an additional k-mediod analysis of 5 clusters are used to vary and reduce the dataset. The versatile variation possibilities that this dataset brings about is discussed in the next chapter.

3.2 Possible Dataset Variations

Five different dataset variations are subject to the analysis presented in this chapter. With the *instruction* and *experiment labels* it is possible to use the full dataset with all 1140 samples with either set of labels, yielding the first and second dataset variation. In this chapter these two variations are referred to by the name of the respective set of labels (*instruction labels* and *experiment labels*). The interchange from *instruction* to *experiment labels* allows a performance evaluation of the executed classification task with regards to the instructions and the results of the listening experiment. Furthermore, a comparison between the behaviour of the machine learning algorithm and the ranking behaviour of the participants of the listening experiment is possible.

In addtion to the two sets of labels and the three class categorization, the k-mediod algorithm mentioned in section 3.1 was also used to obtain class boundaries for $N_{clust} = 5$ clusters. These five clusters can be viewed as five classes namely, *breathy*, *slightly breathy*, *normal*, *slightly pressed* and *pressed*. Within the five cluster analysis the *pressed* and *breathy* class now contain samples that were more confidently rated with the respective voice quality. The labels obtained from this cluster analysis and the comparison of *instruction* and *experiment labels* enable the reduction of the dataset for three voice quality classes, which contain more unambigious samples. This results in 3 additional dataset variations, which are mentioned in subsection 3.2.1. The balancing step of the dataset variations with regard to the number of samples per voice quality class using the random undersampling (RUS) method, is mentioned in subsection 3.2.2.

3.2.1 Reduced Dataset Variations

First dataset reduction

The comparison of *instruction* and *experiment labels* allows a reduction of the data to the part where the *experiment* and *instruction labels* coincide. This leads to a dataset which is also validated by the listening experiment. If this is done, the number of samples of the dataset is reduced to 757 samples from the inital 1140 samples. When looking at the sample's distribution across the three classes the number of samples per class are given with:

Table 3.7 *Number of samples for the first dataset reduction containing only samples where the experiment and instruction labels coincide.*

- *breathy* voice quality:	$N_{\mathrm{b}}^{\mathrm{Red\,I}} = 280$ samples
- *normal* voice quality:	$N_{\mathrm{n}}^{\mathrm{Red\,I}} = 270$ samples
- *pressed* voice quality:	$N_{\mathrm{p}}^{\mathrm{Red\,I}} = 207$ samples
	$N^{\mathrm{Red\,I}} = 757$ samples

This first reduction stage of the dataset is further on referred to as the *first dataset reduction*.

Second dataset reduction

Further reduction can be executed by discarding the *slightly* classes that are given through the 5-cluster-analysis, resulting in an overall amount of 472 samples, which in this thesis is adverted as the *second dataset reduction*. Across the three classes the 472 samples are distributed with:

Table 3.8 *Number of samples for the second dataset reduction containing only samples where the experiment and instruction labels coincide and no samples of the "slightly" classes.*

- *breathy* voice quality:	$N_{\mathrm{b}}^{\mathrm{Red\ II}} = 161$ samples
- *normal* voice quality:	$N_{\mathrm{n}}^{\mathrm{Red\ II}} = 211$ samples
- *pressed* voice quality:	$N_{\mathrm{p}}^{\mathrm{Red\ II}} = 100$ samples
	$N^{\mathrm{Red\ II}} = 472$ samples

Third dataset reduction

Even though there is little to non correlation between the discarded samples in the *first* and *second dataset reduction* and the singer's gender (see Table A.2 and A.6), a gender homogenous dataset reduction is introduced. Additionally this provieds a more compact pitch range, within the dataset, which is often a crucial aspect in signal processing based analysis of singing voice. There often exist correlations towards the fundamental frequency, e.g. as shown for the vocal tract filter estimation executed in [4]. Hence, the *third dataset reduction* is introduced.

The *third dataset reduction* is achieved by taking the previously discussed *second dataset reduction* and limiting it to the samples of the female singers. Thus results in samples with a more compact fundamental frequency range and neglects the lower pitches c and g. This leaves 345 samples, which are split into the three classes according to the following numbers:

Table 3.9 *Number of samples for the third dataset reduction containing only samples by female singers, where the experiment and instruction labels coincide and no samples belonging to the slightly classes.*

- *breathy* voice quality:	$N_{\mathrm{b}}^{\mathrm{Red\ III}} = 121$ samples
- *normal* voice quality:	$N_{\mathrm{n}}^{\mathrm{Red\ III}} = 156$ samples
- *pressed* voice quality:	$N_{\mathrm{p}}^{\mathrm{Red\ III}} = 68$ samples
	$N^{\mathrm{Red\ III}} = 345$ samples

In order to provide some context on how the discarded samples for the *first* and *second dataset reduction* are distributed over voice-quality, singers, vowels and pitches, section A.1 is added in the appendix. In it, tables containing the number of discarded samples in absolute and relative values and their relation towards the voice-quality, the singers, the vowels or the pitches are listed. This also indirectly provides insight into the voice quality rating behaviour summarized in Figure 3.2, because possible accumulations of discarded samples with regards to voice-quality, singers, vowels or pitches, indicate a reason for the discarding of these samples. Concerning the instructed voice quality Table A.1 reveals the same as Figure 3.2. The most samples are discarded for the *pressed* instruction class, as this is also the class with the most samples where *experiment* and *instruction* lables do not match. Also, it is visible that the highest numbers for the *first* and *second datset reduction* mostly occur for the same singers, vowels and pitches. When looking at the vowels it is visible that there is no prominent vowel for which the most samples are discarded. For the *first* and *second dataset reduction* the discarded samples are evenly discarded over all vowels. The discarded samples in context with the sung pitches reveal that the relative values in percent are mostly evenly distributed, only the highest two pitches, g^2 and a^2 stand out. Concerning the singers S1 exhibits the most discarded samples in terms of absolute values and S8 shows the highest relative value. Because S1 is a female singer and S8 is a male singer, the indication is given that gender did not seem to play a vital role in the phonation type ranking, which yielded deviating *instruction* and *experiment* labels. Further analysis targeting the sources of the deviation between *instruction* and *experiment* labels, e.g. through a thor-

ough analysis of each listeners rating behaviour separately, is not included in this thesis, because this thesis focuses on the analysis and classification of the resulting dataset and its variation possibilities.

Concluding, depending on the used dataset variation the number of samples N_{samples} is given as:

$$N_{\text{samples}} = \begin{cases} N^{\text{inst}} = 1140 \text{ samples, for } \textit{instruction labels} \\ N^{\text{exp}} = 1140 \text{ samples, for } \textit{experiment labels} \\ N^{\text{Red I}} = 757 \text{ samples, for } \textit{the first dataset reduction} \\ N^{\text{Red II}} = 472 \text{ samples, for } \textit{the second dataset reduction} \\ N^{\text{Red III}} = 345 \text{ samples, for } \textit{the third dataset reduction} \end{cases} \tag{3.1}$$

3.2.2 Dataset Balancing

In the classification task described in this thesis, phonation type descriptive features are calculated for all underlying recorded sung vocal samples, resulting in a so-called *feature set*, presented in section 3.1. The *feature set* consist of observations or observation samples which describe a certain aspect of the sung vocal recordings mentioned in section 3.1 and the term *dataset* refers to the sung vocal recordings. However in the context of the dataset reductions presented in subsection 3.2.1 and the balancing step proposed in this section the terms can be used synonymously. Because the proposed reduction steps and balancing steps, solely depend on the voice quality labels assigned to each feature set observation or sung vocal recording in the dataset. The voice quality labels of an observation and of the sung vocal recording are equivalent. This means that the proposed dataset reductions are also applicable for the feature set, where the discarded samples now refer to an observation sample contained in the feature set.

The usage of the mentioned dataset variations result in imbalanced feature- or datasets. “Imbalanced“ means a dataset has an unequal number of data samples per class [13]. This is the case for all the dataset variations mentioned in section 3.2, except if *instruction* labels are used. The handling of such imbalanced datasets is a crucial aspect of ML based classification tasks and therefore this section provides information on how the topic was handled during the course of this project.

With regards to the *instruction labels* the dataset is balanced, meaning the same amount of samples is observable for each voice quality class. However, when the *experiment labels* come into play, either in form of labels for the full dataset or in form of the dataset reductions mentioned in section 3.2, the equal distribution of samples per class gets skewed, which introduces a bias towards the class containing the majority of the samples [13]. One could argue that the model focuses more on the data with a higher number of samples per class. This is a common problem encountered in machine learning and one of the easiest ways to counteract is a method called random undersampling (RUS), mentioned in [13, p.83], which basically states that a dataset can be balanced by randomly discarding the majority class examples, thus ensuring that the number of samples in all classes is limited to the minimal number of samples occuring within a class. Once more this yields a reduction of samples. A summary of the number of samples before and after random undersampling is listed in Table 3.10. The number of samples after random undersampling (RUS) is indicated in the symbol variables’ indices.

Table 3.10 *Number of samples summary for all dataset variations before and after random undersampling in order to balance the dataset, the units of all numbers in this table are given as number of samples.*

dataset variation	**voice quality**			**total**
	breathy	***normal***	***pressed***	
instruction labels	$N_{\mathrm{b}}^{\mathrm{inst}} = 380$	$N_{\mathrm{n}}^{\mathrm{inst}} = 380$	$N_{\mathrm{p}}^{\mathrm{inst}} = 380$	$N^{\mathrm{inst}} = 1140$
instruction labels **(RUS)**	$N_{\mathrm{b,RUS}}^{\mathrm{inst}} = 380$	$N_{\mathrm{n,RUS}}^{\mathrm{inst}} = 380$	$N_{\mathrm{p,RUS}}^{\mathrm{inst}} = 380$	$N_{\mathrm{RUS}}^{\mathrm{inst}} = 1140$
experiment labels	$N_{\mathrm{b}}^{\mathrm{exp}} = 297$	$N_{\mathrm{n}}^{\mathrm{exp}} = 517$	$N_{\mathrm{p}}^{\mathrm{exp}} = 326$	$N^{\mathrm{exp}} = 1140$
experiment labels **(RUS)**	$N_{\mathrm{b,RUS}}^{\mathrm{exp}} = 297$	$N_{\mathrm{n,RUS}}^{\mathrm{exp}} = 297$	$N_{\mathrm{p,RUS}}^{\mathrm{exp}} = 297$	$N_{\mathrm{RUS}}^{\mathrm{exp}} = 891$
first dataset reduction	$N_{\mathrm{b}}^{\mathrm{Red\ I}} = 280$	$N_{\mathrm{n}}^{\mathrm{Red\ I}} = 270$	$N_{\mathrm{p}}^{\mathrm{Red\ I}} = 207$	$N^{\mathrm{Red\ I}} = 757$
first dataset reduction **(RUS)**	$N_{\mathrm{b,RUS}}^{\mathrm{Red\ I}} = 207$	$N_{\mathrm{n,RUS}}^{\mathrm{Red\ I}} = 207$	$N_{\mathrm{p,RUS}}^{\mathrm{Red\ I}} = 207$	$N_{\mathrm{RUS}}^{\mathrm{Red\ I}} = 621$
second dataset reduction	$N_{\mathrm{b}}^{\mathrm{Red\ II}} = 161$	$N_{\mathrm{n}}^{\mathrm{Red\ II}} = 211$	$N_{\mathrm{p}}^{\mathrm{Red\ II}} = 100$	$N^{\mathrm{Red\ II}} = 472$
second dataset reduction **(RUS)**	$N_{\mathrm{b,RUS}}^{\mathrm{Red\ II}} = 100$	$N_{\mathrm{n,RUS}}^{\mathrm{Red\ II}} = 100$	$N_{\mathrm{p,RUS}}^{\mathrm{Red\ II}} = 100$	$N_{\mathrm{RUS}}^{\mathrm{Red\ II}} = 300$
third dataset reduction	$N_{\mathrm{b}}^{\mathrm{Red\ III}} = 121$	$N_{\mathrm{n}}^{\mathrm{Red\ III}} = 156$	$N_{\mathrm{p}}^{\mathrm{Red\ III}} = 68$	$N^{\mathrm{Red\ III}} = 345$
third dataset reduction **(RUS)**	$N_{\mathrm{b,RUS}}^{\mathrm{Red\ III}} = 68$	$N_{\mathrm{n,RUS}}^{\mathrm{Red\ III}} = 68$	$N_{\mathrm{p,RUS}}^{\mathrm{Red\ III}} = 68$	$N_{\mathrm{RUS}}^{\mathrm{Red\ III}} = 204$

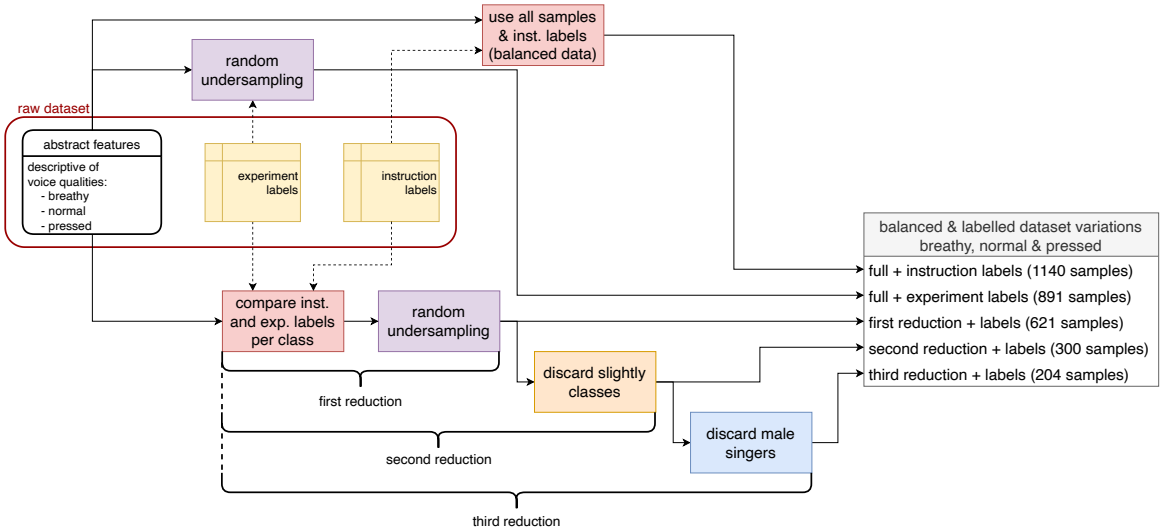


Figure 3.3 *Processing of raw data in order to obtain a balanced and labelled dataset.*

The creation of the different dataset variations including the balancing of the dataset resulting in a balanced & labelled dataset is summed up in Figure 3.3. The raw dataset consisting of the calculated features mathematically describing the voice quality and the two possible label sets, the *experiment* and *instruction* labels are marked in red. The three reduction steps are clearly marked at the bottom of Figure 3.3. At the top, the creation of the two full dataset variations, either with *experiment* or *instruction* labels, is visualized. At the right end of Figure 3.3 the six possible dataset variations are depicted as a list. The depiction of a processable, balanced and labelled dataset in Figure 3.3 is as a list is continued throughout Figure 3.3 - 3.5.

Due to different implementation structures of one ML-classifier presented in section 3.3 an additional processing step of the balanced and labelled dataset shown in Figure 3.3 is necessary. The certain classifier structure requires a distinction into a *breathy* and *rest* class, whereas the *rest* consists of sung vocal samples with *normal* and *pressed* voice quality. The *rest* class holds the majority of the samples, as it is made up of two classes (*normal* and *pressed*). Therefore, the underlying sub-dataset needs to be balanced in such a way that it consists of the same number of *rest* and *breathy* samples. In order to do so, the balancing can be understood as an additional processing step applied onto the balanced and labelled dataset resulting from Figure 3.3. This is shown in the processing chain located at the top of Figure 3.4. The resulting number of samples are summarized in the top half of Table 3.11.

Table 3.11 *Number of samples summary for both sub-datasets with and without RUS, enabling the separate analysis of breathy vs. rest and normal vs. pressed classification.*

dataset variation	**voice quality**		**total**
	breathy	***rest*** (*normal & pressed*)	
instruction labels	$N_{\mathrm{b}}^{\mathrm{inst}} = 380$	$N_{\mathrm{r}}^{\mathrm{inst}} = 760$	$N_{\mathrm{b\text{-}r}}^{\mathrm{inst}} = 1140$
instruction labels **(RUS)**	$N_{\mathrm{b}}^{\mathrm{inst}} = 380$	$N_{\mathrm{r}}^{\mathrm{exp}} = 380$	$N_{\mathrm{b\text{-}r,RUS}}^{\mathrm{inst}} = 760$
experiment labels	$N_{\mathrm{b}}^{\mathrm{exp}} = 297$	$N_{\mathrm{r}}^{\mathrm{exp}} = 843$	$N_{\mathrm{b\text{-}r}}^{\mathrm{exp}} = 1140$
experiment labels **(RUS)**	$N_{\mathrm{b}}^{\mathrm{inst}} = 297$	$N_{\mathrm{r}}^{\mathrm{exp}} = 297$	$N_{\mathrm{b\text{-}r,RUS}}^{\mathrm{exp}} = 594$
first dataset reduction	$N_{\mathrm{b}}^{\mathrm{Red\ I}} = 280$	$N_{\mathrm{r}}^{\mathrm{Red\ I}} = 477$	$N_{\mathrm{b\text{-}r}}^{\mathrm{Red\ I}} = 757$
first dataset reduction **(RUS)**	$N_{\mathrm{b,RUS}}^{\mathrm{Red\ I}} = 280$	$N_{\mathrm{r,RUS}}^{\mathrm{Red\ I}} = 280$	$N_{\mathrm{b\text{-}r,RUS}}^{\mathrm{Red\ I}} = 560$
second dataset reduction	$N_{\mathrm{b}}^{\mathrm{Red\ II}} = 161$	$N_{\mathrm{r}}^{\mathrm{Red\ II}} = 311$	$N_{\mathrm{b\text{-}r}}^{\mathrm{Red\ II}} = 472$
second dataset reduction **(RUS)**	$N_{\mathrm{b,RUS}}^{\mathrm{Red\ II}} = 161$	$N_{\mathrm{r,RUS}}^{\mathrm{Red\ II}} = 161$	$N_{\mathrm{b\text{-}r,RUS}}^{\mathrm{Red\ II}} = 322$
third dataset reduction	$N_{\mathrm{b}}^{\mathrm{Red\ III}} = 121$	$N_{\mathrm{r}}^{\mathrm{Red\ III}} = 224$	$N_{\mathrm{b\text{-}r}}^{\mathrm{Red\ III}} = 345$
third dataset reduction **(RUS)**	$N_{\mathrm{b,RUS}}^{\mathrm{Red\ III}} = 121$	$N_{\mathrm{r,RUS}}^{\mathrm{Red\ III}} = 121$	$N_{\mathrm{b\text{-}r,RUS}}^{\mathrm{Red\ III}} = 242$

dataset variation	**voice quality**		**total**
	normal	***pressed***	
instruction labels	$N_{\mathrm{n}}^{\mathrm{inst}} = 380$	$N_{\mathrm{p}}^{\mathrm{inst}} = 380$	$N_{\mathrm{n\text{-}p}}^{\mathrm{inst}} = 760$
instruction labels **(RUS)**	$N_{\mathrm{n,RUS}}^{\mathrm{inst}} = 380$	$N_{\mathrm{p,RUS}}^{\mathrm{inst}} = 380$	$N_{\mathrm{n\text{-}p,RUS}}^{\mathrm{inst}} = 760$
experiment labels	$N_{\mathrm{n}}^{\mathrm{exp}} = 517$	$N_{\mathrm{p}}^{\mathrm{exp}} = 326$	$N_{\mathrm{n\text{-}p}}^{\mathrm{exp}} = 843$
experiment labels **(RUS)**	$N_{\mathrm{n,RUS}}^{\mathrm{exp}} = 326$	$N_{\mathrm{p,RUS}}^{\mathrm{exp}} = 326$	$N_{\mathrm{n\text{-}p,RUS}}^{\mathrm{exp}} = 652$
first dataset reduction	$N_{\mathrm{n}}^{\mathrm{Red\ I}} = 270$	$N_{\mathrm{p}}^{\mathrm{Red\ I}} = 207$	$N_{\mathrm{n\text{-}p}}^{\mathrm{Red\ I}} = 477$
first dataset reduction **(RUS)**	$N_{\mathrm{n,RUS}}^{\mathrm{Red\ I}} = 207$	$N_{\mathrm{p,RUS}}^{\mathrm{Red\ I}} = 207$	$N_{\mathrm{n\text{-}p,RUS}}^{\mathrm{Red\ I}} = 414$
second dataset reduction	$N_{\mathrm{n}}^{\mathrm{Red\ II}} = 211$	$N_{\mathrm{p}}^{\mathrm{Red\ II}} = 100$	$N_{\mathrm{n\text{-}p}}^{\mathrm{Red\ II}} = 311$
second dataset reduction **(RUS)**	$N_{\mathrm{n,RUS}}^{\mathrm{Red\ II}} = 100$	$N_{\mathrm{p,RUS}}^{\mathrm{Red\ II}} = 100$	$N_{\mathrm{n\text{-}p,RUS}}^{\mathrm{Red\ II}} = 200$
third dataset reduction	$N_{\mathrm{n}}^{\mathrm{Red\ III}} = 156$	$N_{\mathrm{p}}^{\mathrm{Red\ III}} = 68$	$N_{\mathrm{n\text{-}p}}^{\mathrm{Red\ III}} = 156$
third dataset reduction **(RUS)**	$N_{\mathrm{n,RUS}}^{\mathrm{Red\ III}} = 68$	$N_{\mathrm{p,RUS}}^{\mathrm{Red\ III}} = 68$	$N_{\mathrm{n\text{-}p,RUS}}^{\mathrm{Red\ III}} = 136$

The second sub-dataset which the special ML-classifier structure mentioned in section 3.3 requires, has to consist of only *normal* and *pressed* voice quality. The number of samples before and after balancing using random untersampling are listed in the lower half of Table 3.11. The additional processing steps applied onto the balanced and labelled dataset from Figure 3.3, to derive the subdataset with *normal* and *pressed* voice quality is shown in the lower processing chain of Figure 3.4.

This is pointed out in Figure 3.4, where the origin of the balanced and labelled full dataset with *experiment labels* for the differentiation of *normal* and *pressed* phonation consisting of 652 samples, is again found within the raw dataset. For the other dataset variations of the second classification stage the resulting datasets from Figure 3.3 are used and the samples belonging to the *breathy* class are discarded. This is possible because for the dataset reductions the distinction of *experiment* and *instruction labels* is not relevant anymore, as a reduced dataset only contains samples where the *experiment* and *instruction labels* are identical and the random undersampling which has been executed beforehand and is visualized in Figure 3.3, leads to an already balanced origin dataset for all three classes.

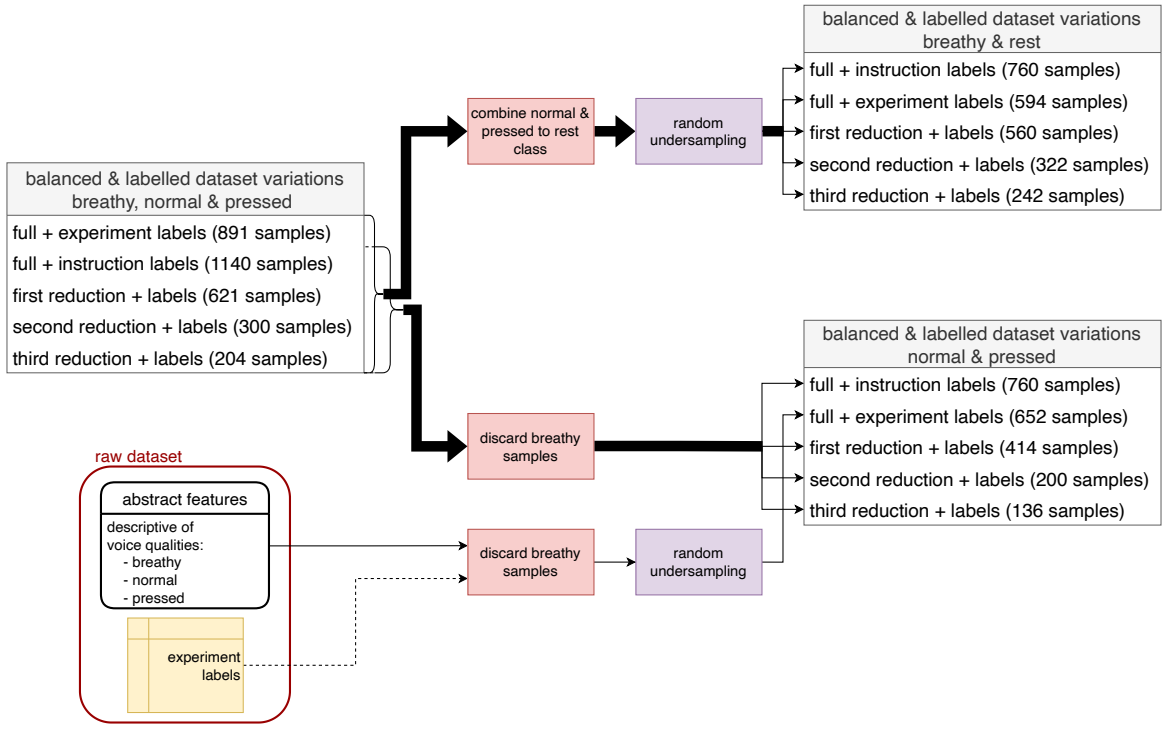


Figure 3.4 *Separation of the balanced and labelled dataset into two balanced and labelled subdatasets containing samples for distinction of breathy vs. rest and normal vs. pressed.*

In summary, it can be stated that the task of the dataset balancing, including, random undersampling, is to ensure that the underlying dataset and its variations always contain the same number of samples across each class, by looking for the smallest number of samples available for each class, independent from the number of classes.

3.3 Implementation of the SVM Classifier

As already mentioned in section 3.2, different ML-classifier structures are used in this thesis. Based on theoretical considerations the support vector machine (SVM) was chosen as the underlying classifier with regard to the phonation type. The theory behind SVMs is formulated in section 2.5. The binary classification structure of SVMs is utilized in different configurations in order to classifiy the previously described dataset variations as well as the two sub-datasets illustrated in Figure 3.4.

SVM structure

Concerning the structure of the classifier, two different approaches are analyzed in this thesis. On the one hand, a multiclass SVM-model is created with three binary classifiers. This leads to a classification of all three voice quality classes within a single classification stage. On the other hand, a two stage classification is also executed, this means that the classification into three classes happens in two stages. More precisely the first classification stage deals with the distinction of *breathy* samples. In the second stage the classes *normal* and *pressed* are distinguished. This is done to enable the usage of different feature sets for the different classification stages. So for *breathy* samples different features can then be used for the classification of *normal* and *pressed* samples. Concerning the implementation using the software application `Matlab` each classification stage requires a classifier-model, which for both structures, the single stage SVM and the two stage SVM, is done with `fitcecoc()` [31]. In order to discuss the features and their potential, different performance measures quantifying their classification capabilities are necessary. These performance measures are discussed in subsection 3.3.1.

Matlab implementation

In this paragraph the practical realisation of the SVM, its structure and its parameters are explained. Regarding the executed analysis, the classifier model itself is created using the `Matlab` command `fitcecoc()` [31]. As mentioned in section 2.5, a single support vector machine is a binary classifier allowing a classification of only two classes. Therefore, multiple support vector machines are needed, if more than two classes are distinguished. The task executed in this thesis requires a separation into three classes, this is why `fitcecoc()` was used, as it also enables multiclass SVM classifier models. In order to create a multiclass SVM-model, a SVM-template has to be created using the command `templateSVM()`, thus creating a binary "Learner", which serves as the classifier template [42]. The classifier template is handed to `fitcecoc()` with the function argument `'Learners'`. For a three class classification with one mutliclass SVM-model, three binary classifiers, one for each class, are needed. These three binary classifiers are then subject to a certain comparison strategy in order to make a three class separation possible. In this project the strategy pursued is the "one-versus-all" strategy. This means that in the case of three classes, each binary learner is used to set up a binary classification problem and one class is assigned to be the positive class, the other two classes are negative classes meaning that this method tries every possible combination of positive classes [31]. For the three voice quality classes the three binary classification problems would be:

- *normal* vs. [*breathy* & *pressed*]
- *breathy* vs. [*normal* & *pressed*]
- *pressed* vs. [*normal* & *breathy*]

In order to avoid ambiguities where a sample is assigned to multiple classes the raw output of the k^{th} SVM $y'_k(\boldsymbol{x})$ is required. The raw output of a SVM is the model output $y(\boldsymbol{x})$ before it is handed to the activation function as discussed in section 2.5. $y'_k(\boldsymbol{x})$ is then processed using the $\mathrm{argmax}(\cdot)$ or $\max(\cdot)$ operator [5, p.338] in order to find the highest raw output $y(\boldsymbol{x})$ which also determines the most likely class label for the processed sample. Each binary classifier outputs a positive $y'_k(\boldsymbol{x})$ if

the processed sample belongs to the positive class. Thus, according to [5, p.338], the class labels $k_{\max} \in \{$*'normal'*, *'breathy'*, *'pressed'*$\}$ of the multiclass SVM and its raw output $y(\boldsymbol{x})$ for the one-versus-all approach of multiples classes is predicted with:

$$\begin{aligned} k_{\max} &= \underset{k}{\operatorname{argmax}} \left(y'_k(\boldsymbol{x})\right) \\ y'(\boldsymbol{x}) &= y'_{k_{\max}}(\boldsymbol{x}) = \max_k \left(y'_k(\boldsymbol{x})\right) \end{aligned} \tag{3.2}$$

Within `fitcecoc()` this method is chosen by using `'onevsall'` in combination with the function argument `'Coding'`.

Box constraint

As discussed in subsection 2.5.3, the box constraint C from Equation 2.36 controls the penalization of misclassifications in a SVM. For all SVM models used in the analysis mentioned in this chapter, C is set to the default value 1. Initially, a hyperparameter optimization including a grid search, provided with `Matlab's fitcecoc()` [31], which iteratively adapts the box constraint and the kernel scale (mentioned in the next paragraph), was carried out. However, the results achieved with the parameters resulting from this optimization procedure, produced comparable results to the results achieved with the default valued box constraint and the kernel scale estimation method mentioned in the next paragraph. Additionally, the runtime, especially the training time, is reduced abundantly when using a fixed box constraint value. This is of importance as the evaluation process according to the measures discussed in subsection 3.3.1 includes multiple executions of the classification process in order to be able to assess the statistical deviations of the calculated measures.

Kernel function and kernel scale

A vital parameter of an SVM is the *kernel function* which is extensively discussed in subsection 2.5.2. For the voice quality classification a radial basis function (rbf) kernel is used, using the function parameter `'KernelFunction'` when the binary learner template is created with `templateSVM()` of [42]. The parameter that comes with a rbf kernel is the kernel-scale γ (see Equation 2.33). In `Matlab` it is possible to set the kernel scale value to `'auto'` using the `'KernelScale'` function argument of `templateSVM()` [42]. Using `'auto'` automatically fixes the kernel scale by executing a heuristic procedure, where the used training set is subsampled. Unfortunately, a detailed description of this heuristic method is not mentioned in [42]. However, the `Matlab` code which is executed when using the kernel scale value `'auto'`, can be viewed by entering the command "`edit classreg.learning.svmutils.optimalKernelScale`" into `Matlab's` command window. Within the code it becomes visible that the heuristic method for two class learning involves the calculation of the median Eucedian distance towards the other class' nearest neighbours. More precisely, this means that the Eucledian distance of a sample towards the nearest neighbour of the opposite class is calculated for 100 observations per class. These 100 observations are subsampled from the underlying feature set. The kernel-scale is then ultimately chosen as the median value of the the subsampled data points' Eucledian distances.

When fitting a SVM-model with `'auto'` it is important to know that the results can deviate from one function call to another. Therefore, a function was written which allows the iterative execution of the SVM fitting process with `fitcecoc()` and the return of the kernel scale to obtain several estimations for choosing the kernel scale. The mean of the kernel scale can then be handed to the actual SVM-template that is fitted to perform the phonation type classification. This is done to surpress the statistic variance that would originate from the kernel-scale variation, if the same classification process with the same features would be carried out a multiple times. For the feature selection procedure described

in section 3.5, 50 iterations were used to determine the kernel scale. 500 iterations were used in the performance overview depicted in Figure 3.8 and Figure 3.9 of subsection 3.4.1.

3.3.1 Performance Measures

The dataset or one of the variations mentioned in section 3.2 are the starting point of a ML-based classification task whose processing steps are outlined in section 2.1. Depending on the number of classification stages the measures are calculated once or twice. Thus, the distinction achieved by the two stage classification model results in separate performance measures for each stage. Allowing an evaluation of each classification stage and also a comparison on which classes are more distinctly identifiable. The calculation of the performance measures using the single stage classifier model is visualized in Figure 3.5. For the two stage SVM-model the illustrated schematic and the depicted processes are simply duplicated but only differ in the underlying dataset and its variations, as the stages of the two stage SVM only classify two classes. This means that the depicted balanced and labelled dataset variations of Figure 3.5 would have to be replaced by the ones shown on the right end of Figure 3.4 indicating that the depicted calculation process of Figure 3.5 is built modularily as only the underlying data varies but the performance measure calculation remains unchanged.

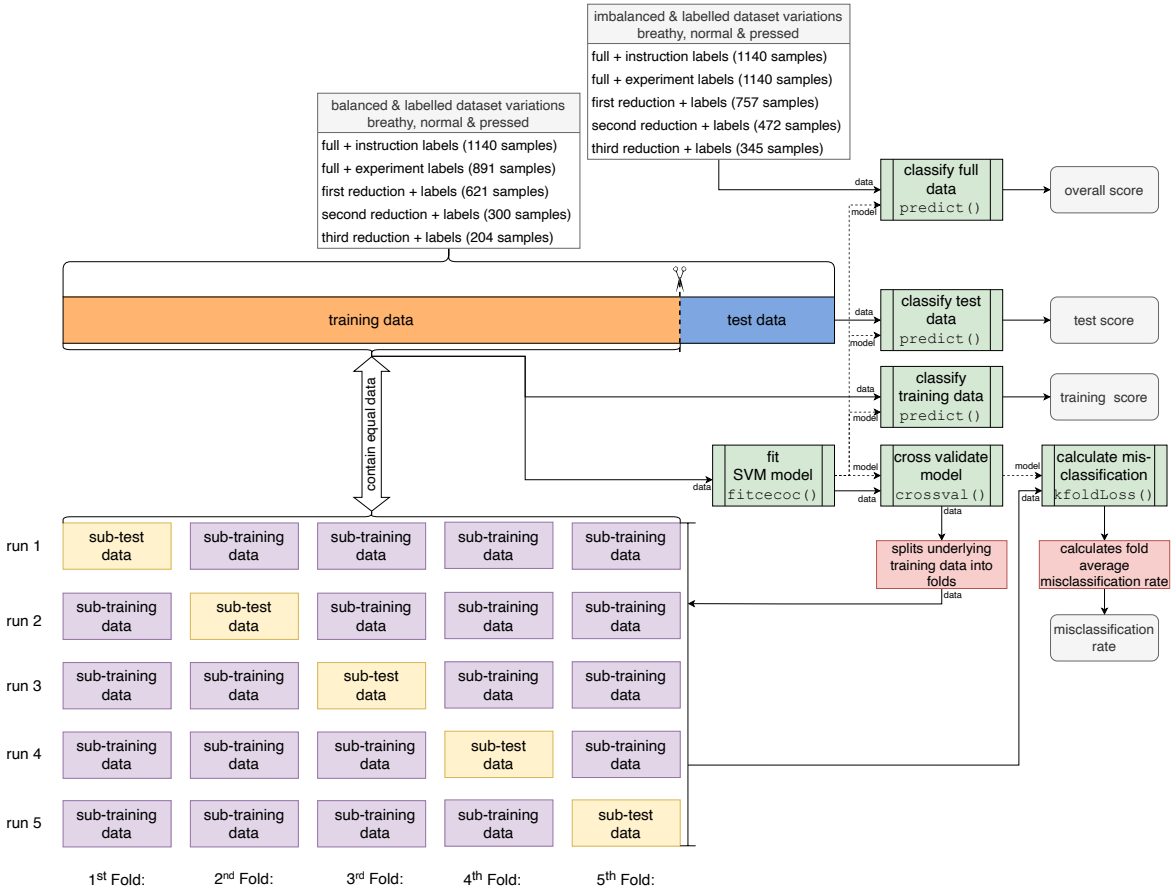


Figure 3.5 *Data split and calculation of performance measures.*

Training and *test score*

The underlying dataset (or a variation of it) is randomly split into 80 % training data and into 20 % test data, also referred to as the hold-out set. The training data is used to fit the SVM-model using

`Matlab's fitcecoc()`. The trained model is then further processed using the `Matlab`-command `predict()` [38], this command takes data and a trained SVM-model and assigns each sample to a class. The *training* and *test score* is then calculated as the percentage of correctly classified samples of the respective set, which is formulated in Equation 3.3 and 3.4.

$$p_{\text{train}} = \frac{\#\,(\text{correctly classified training samples})}{\#\,(\text{training samples})} \cdot 100\,\% = \frac{TP + TN}{TP + TN + FP + FN} \cdot 100\,\% \tag{3.3}$$

$$p_{\text{test}} = \frac{\#\,(\text{correctly classified test samples})}{\#\,(\text{test samples})} \cdot 100\,\% = \frac{TP + TN}{TP + TN + FP + FN} \cdot 100\,\% \tag{3.4}$$

The *training* and *test scores* always have to be viewed in relation e.g. a model with a relatively large deviation between both scores exhibits overfitted behaviour [5, p.25-26]. Ultimately the *test score* also holds information on the generalizability of the classification, it shows how well a model performs on data that the model hasn't seen before. The operator $\#(\cdot)$ denotes the "number of operator" and in order to identify the number of correctly classified training or test samples, the predicted labels (class predictions) are simply compared with the previously assigned true labels defined by the used dataset variation mentioned in section 3.2. The correcly classified training or test samples are given as the sum of the true positives (TP) and true negatives (TN), denoting the correct classifications of the positive and negative class for either training or test data. The overall amount of training or test samples is calculated as the sum of all true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN), where FP and FN respectively denote the false classifications within the positive and negative class.

Misclassification rate

Another measure that strongly correlates with the *test score* is the *misclassification rate*. This measure allows the assessment on the percentage of wrongly classified samples, which are to be expected of a trained model. In the context of this thesis the *misclassification rate* is calculated through a 5-fold crossvalidation, using the commands `crossval()` [28] and `kfoldLoss()` [33] as indicated in Figure 3.5. The function `crossval()` only obtains the SVM model that has been fitted with the training data. In `Matlab` the SVM-structures are cached with the underlying data, so if `crossval()` obtains the trained SVM model, the underlying training data is also handed to it inherently. With `crossval()` a cross validation object is created. Within the crossvalidation object the underlying training data is separated into five equally sized data blocks, also called folds as indicated in Figure 3.5. The crossvalidation object, which again inherently includes the training data split into 5 folds, is then handed to the function `kfoldLoss()`. It executes 5 runs, in which 4 folds are used as "sub-training data" and 1 fold is used as the "sub-test data". The sub-training data is used to train a SVM-model with the same specifications as the initally fitted SVM-model, which was handed to `crossval()`. The *misclassification rate* is evaluated for each sub-test data fold in each run [5, p.33]. The *misclassification rates* calculated in each are then averaged across the 5 runs, leading to what in [31] is referred to as the generalized classification error. As the terms *classification error* and *misclassification rate* are often used to describe equal measures, it is important to note that in this thesis the measure referred to as the *misclassification rate* is an averaged measure and for one fold it is defined as the percentage of incorrectly classified samples, in accorrdance with [48, p.10]. The term "classification error" is slightly more general and it is also often used synonymously with the term "classification loss", especially in the `Matlab` documentation [31], [33] and [28]. Mathematically, the *misclassification rate* is calculated according to Equation 3.5 and 3.6, where $p_{\text{misclass},\,j}$ describes the

misclassification rate of the j^{th} fold and S is the number of folds.

$$p_{\text{misclass},\,j} = \frac{\#\,(\text{incorrectly classified test fold samples})}{\#\,(\text{test fold samples})} \cdot 100\,\% = \frac{FP + FN}{TP + TN + FP + FN} \cdot 100\,\% \tag{3.5}$$

Equation 3.6 denotes the averaginge of $p_{\text{misclass},\,j}$ across the folds, resulting in the third performance measure $\bar{p}_{\text{misclass}}$, referred to as the *misclassification rate*.

$$\bar{p}_{\text{misclass}} = \frac{1}{S} \sum_{j=1}^{S} p_{\text{misclass},\,j} \tag{3.6}$$

For the evaluations carried out in this thesis the number of folds was chosen with $S = 5$ folds. The *test score* $p_{\text{test},j}$ and *misclassification rate* of one fold show the following relation:

$$p_{\text{misclass},j} = 100\,\% - p_{\text{test},j} \tag{3.7}$$

As visualized in Figure 3.5 it has to be kept in mind that $p_{\text{test},j}$ is calculated for each test fold of the training set, whereas p_{test} as noted in Equation 3.4 is a seperate measure and is calculated for the hold-out set which takes no part in the calculation of the *misclassification rate*. Nevertheless, as both measures describe the generalizabilty, but are subject to different data, Equation 3.8 does not hold for p_{test} and $\bar{p}_{\text{misclass}}$, but it can be seen as an approximation:

$$\bar{p}_{\text{misclass}} \approx 100\,\% - p_{\text{test}} \tag{3.8}$$

Overall score

The third measure calculated to evaluate the performance of the classification task is the *overall score* which indicates the overall percentage of correctly classified samples:

$$p_{\text{all}} = \frac{\#\,(\text{correctly classified samples})}{\#\,(\text{samples})} \cdot 100\,\% = \frac{TP + TN}{TP + TN + FP + FN} \cdot 100\,\% \tag{3.9}$$

The underlying samples are all contained in the processed dataset variation, meaning the underlying trained classification model (either single stage or two stage SVM) has processed all available data within the used imbalanced dataset variation and predicted their classes.

To conclude the performance measures it is important to note, that apart from the *misclassification rate*, all measures introduced in this chapter are presented as values, which are calculated once. However, from this point on each calculated performance measure mentioned in this thesis, is subject to an averaging process. The random split into training and test data as well as the random undersampling mentioned in subsection 3.2.2 introduce a random component into the SVM-classification procedure. In order to present statistically stabilized performance measures each classification, carried out to calculate the measures, is executed multiple times. For the *misclassification rate* this means, that each *misclassification rate* evaluation presented in the following chapters is subject to a further averaging processes, in addition to the inherent averaging procedure, which is already included in the *misclassification rate's* crossvalidation calculation process, depicted in Figure 3.5.

3.4 Abstract Features and their Parameters

Possible variations enabled through the comparison of *experiment* and *instruction labels* are extensively discussed in section 3.2. So far the descriptive feature dataset has not been discussed, it was only mentioned that they consist of an abstract set of descriptive voice quality measures. What features these are and what augmentations as well as configurations were considered are subject of this section.

3.4.1 Mel Frequency Cepstral Coefficients

The underlying theory on the Mel frequency cepstral coefficients (MFCCs) is mentioned in section 2.2 and its modifications in section 2.2. This section deals with actual feature extraction from the recorded vocal samples, which settings are used and in what way they are modified. Resulting in a list of augmented MFCC Features, whose capabilities with respect to voice quality classification are then compared in an overview. This allows to narrow down multiple possible feature sets to one MFCC feature set whose performance is in detail analyzed in terms of a feature selection algorithm in section 3.5. In general the MFCCs are derived with the calculation steps listed in section 2.2. Yielding the potential of modification for the following aspects:

- filterbank modification
- filterbank center frequencies modification
- cepstral liftering

The main outline for the calculation of the MFCCs and the variants in this thesis is provided by the well established HTK "[...]a toolkit for building Hidden Markov Models (HMMs). " [66, p.2], with the aim of automatic speech recognition. One of the main features used in HTK are MFCCs but as HTK provides its own software architecture and is run through a commando line interface, a Matlab implementation of the way MFCCs are computed within HTK was created. The functions provided in [64] served as a template for the calculation of the MFCC variants that are discussed in this section. Before a MFCC variant is calculated the following pre-processing steps are executed:

1. signal scaling
    - the signal is scaled with 2^{15} in order to use samples in the range of 16-bit shorts as this is the underlying datatype in of HTK proposed in [66].
2. signal blocking
    - the input signal is blocked into $80\,\mathrm{ms}$ blocks with an overlap of $90\,\%$. The first $0.5\,\mathrm{s}$ of each signal are discarded. This is due to the fact that the singers started singing with a closed mouth, which was necessary for the directivity pattern estimation, that has also been carried out in the recording process. The opening phase of the mouth is included in the recorded sample, resulting in a perceptible humming sound which is transforming into the sung vowels. This humming sound, which resembles a voiced */m/*, lasts approximately $0.5\,\mathrm{s}$ and, thus is cut off. This blocking is later also required for the f_0-estimation mentioned in subsection 3.4.2, in order to provide a statistical foundation, by creating multiple estimates for one signal. The blocking parameters chosen in accordance with [4], as the same f_0-estimation procedure was used in it. Concerning the MFCC calculation the coefficients are calculated for each signal block, but for the following classification process no time dependency is investigated. Therefore, the mean MFCCs are determined reducing the time domain.

3. mean subtraction
    - Before the MFCC calculation process is carried out on the signal blocks they are centered by subtracting their means.
4. pre-emphasis filtering
    - In order to counteract the natural decline of energy in voiced speech or sung vocal signals towards higher frequencies, a pre-emphasis filter is applied [52]. The filter is given by a simple first order highpass with a transfer function of

$$H(z) = 1 - \alpha z^{-1} \tag{3.10}$$

whereby the slope of the filter is controlled with α and is chosen as the default value $\alpha = 0.97$ mentioned in [66]. The filter's frequency response is illustrated in Figure 3.6

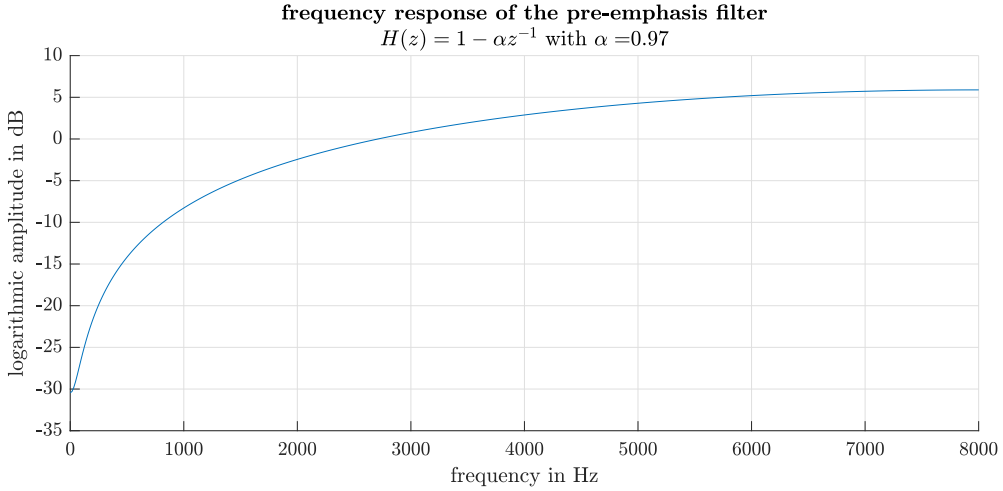


Figure 3.6 *Frequency response of pre-emphasis filter.*

Filterbank modification

The spectral averaging of a frequency spectrum into melbands is carried out using triangular shaped filterbanks as described in subsection 2.2.1. In this project five different filterbanks were used to calculate variants of MFCCs. The frequency response of these filterbanks are depicted in Figure 2.2. The five different filterbanks and the corresponding abbreviations used in this thesis are:

1. classic mel filterbank with constant amplitude: `ConstAmp`
2. classic mel filterbank with decaying amplitude: `DecayAmp`
3. linearily spaced filterbank with constant amplitude: `linear`
4. inverse mel filterbank with constant amplitude (inverse first filterbank): `InvConst`
5. inverse mel filterbank with decaying amplitude (inverse second filterbank): `InvDecay`

The basis of the filterbank design was enabled by modification of the function `fft2melmx.m`, which is contained in D. Ellis' rastamat library [11]. For the mentioned cepstral coefficient variants $N_{\text{MEL}} = 40$ filters are used. This leads to the condensing of a sung vocal signal's frequency spectrum of a sung vocal signal into 40 frequency bands which are then further processed using the DCT as indicated in Equation 2.4

Filterbank center frequencies modification

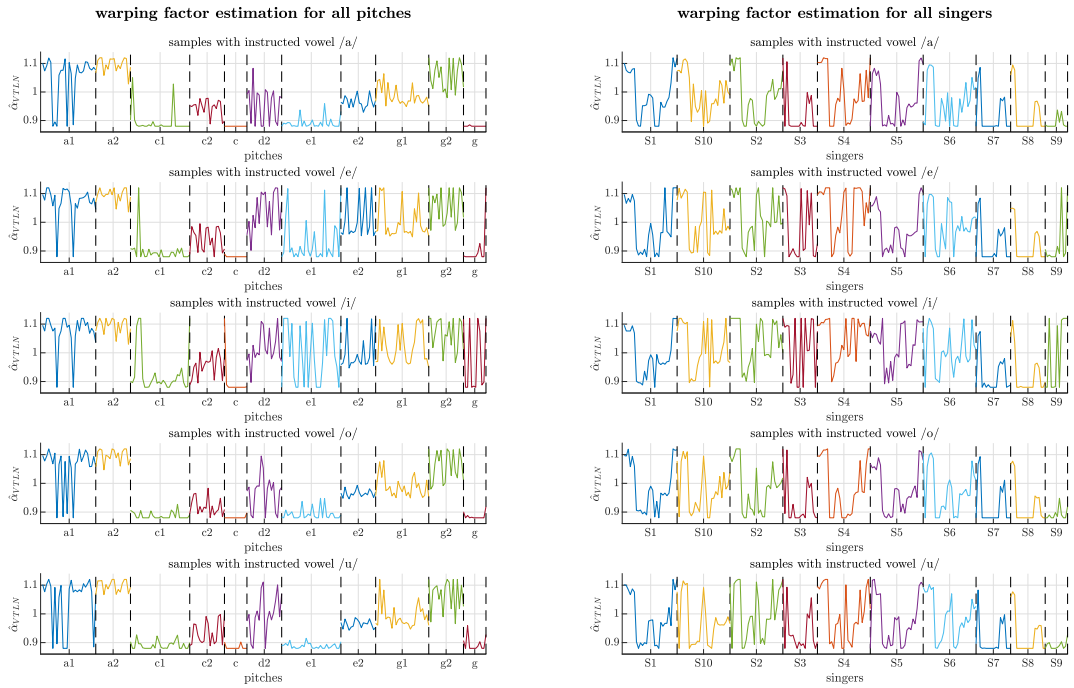
Two possibilities of modifing the filterbank's center frequencies are dealt with in this thesis. The two procedures called Vocal Tract Length Normalization (VTLN) [23] and Vocal Tract Length Pertubation (VTLP) [16] are used to modifiy the center frequencies of a used filterbank in order to attenuate the influence of the vocal length of different singers. VTLN uses the frequency warping approach, from [23] follows the idea of normalizing the vocal tract of each singer, by mapping the vocal length difference between a singer's vocal tract and a reference vocal tract by means of a frequency shift of the filterbank's center frequencies using the frequency warping factor α_{VTLN}. VTLP on the other hand follows the principle of choosing the frequency warping factor randomly, introducing a random component into the center frequency shift, which can be understood as a vocal tract length whitening which according to [16] can improve the classification performance. A mathematical description and analysis of the frequency warping approach proposed in [23] underlying VTLN, which is also used in the HTK implementation [66] and the VTLP approach from [16], can be found in subsection 2.2.2.
For the realization of VTLN a frequency warping factor is estimated for each sample in the range of $\alpha_{\text{VTLN}} \in [0.88, 1.12]$ as proposed in [23]. The estimation process including a MMSE estimation using MFCCs, described in subsection 2.2.2, is executed using reference MFCCs $\boldsymbol{c}_{\text{ref}}$ that has been calculated as the average MFCCs of all samples where:

- voice quality : *normal*
- vowel: /a/
- pitch: $\text{a}^1 \approx 440\,\text{Hz}$

is requested. This is the case for exactly nine files contained in the full dataset presented in section 3.1. An assessment of the frequency warping factor estimation results is possible with Figure 3.7. The estimated frequency warping factors $\hat{\alpha}_{\text{VTLN}}$ for all samples are sorted and set in relation with all pitches and singers. The course of $\hat{\alpha}_{\text{VTLN}}$ across the singers and pitches allow interpretation on how stable the estimation is and if there are dependencies between $\hat{\alpha}_{\text{VTLN}}$ and the singer or pitch of a sample. The results are separated for all vowels by using subplots in order to show deviations in the estimation results across vowels and visualize correlations between $\hat{\alpha}_{\text{VTLN}}$ and the sung pitch or the singer.

If the frequency warping factor estimation would work as pointed out in subsection 2.2.2 little fluctuations for the estimation within the samples sung by the same singer should be anticipated, as $\hat{\alpha}_{\text{VTLN}}$ should show similar results for the same vocal tract length. However, this is not the case. When comparing Figure 3.7(a) and 3.7(b) it is visible that the frequency warping factor estimation is way more stable when looking at the estimation results from a pitch point of view (Figure 3.7(a)) rather than from a singer point of view (Figure 3.7(b)). This indicates that the presumed correlation of the frequency warping factor with the vocal tract length (and hence the singer) does not hold for the present sung vocal signals and the average MFCC reference that is used. A correlation between $\hat{\alpha}_{\text{VTLN}}$ and the present pitch seems more likely as 3.7(b) presents more stable results. But it is also important to point out that when comparing the subplots of Figure 3.7(a) and 3.7(b), $\hat{\alpha}_{\text{VTLN}}$ exhibits similar courses over all pitches and singers across the different vowels, indicating that the influence of the sung vowel as pointed out in subsection 2.2.2 and more precisely in Figure 2.4 does not seem to be as drastic.

Concerning the VTLP, the frequency warping factor is chosen randomly as a uniformly distributed random variable within the interval $\alpha_{\text{VTLP}} \in [0.88, 1.12]$ using `Matlab's rand()` function [39].



(a) $\hat{\alpha}_{\text{VTLN}}$ vs. pitches

(b) $\hat{\alpha}_{\text{VTLN}}$ vs. singers

Figure 3.7 *Estimated frequency warping factor $\hat{\alpha}_{VTLN}$ for all samples set into relation with all pitches & singers, separated for all different vowels.*

Cepstral liftering

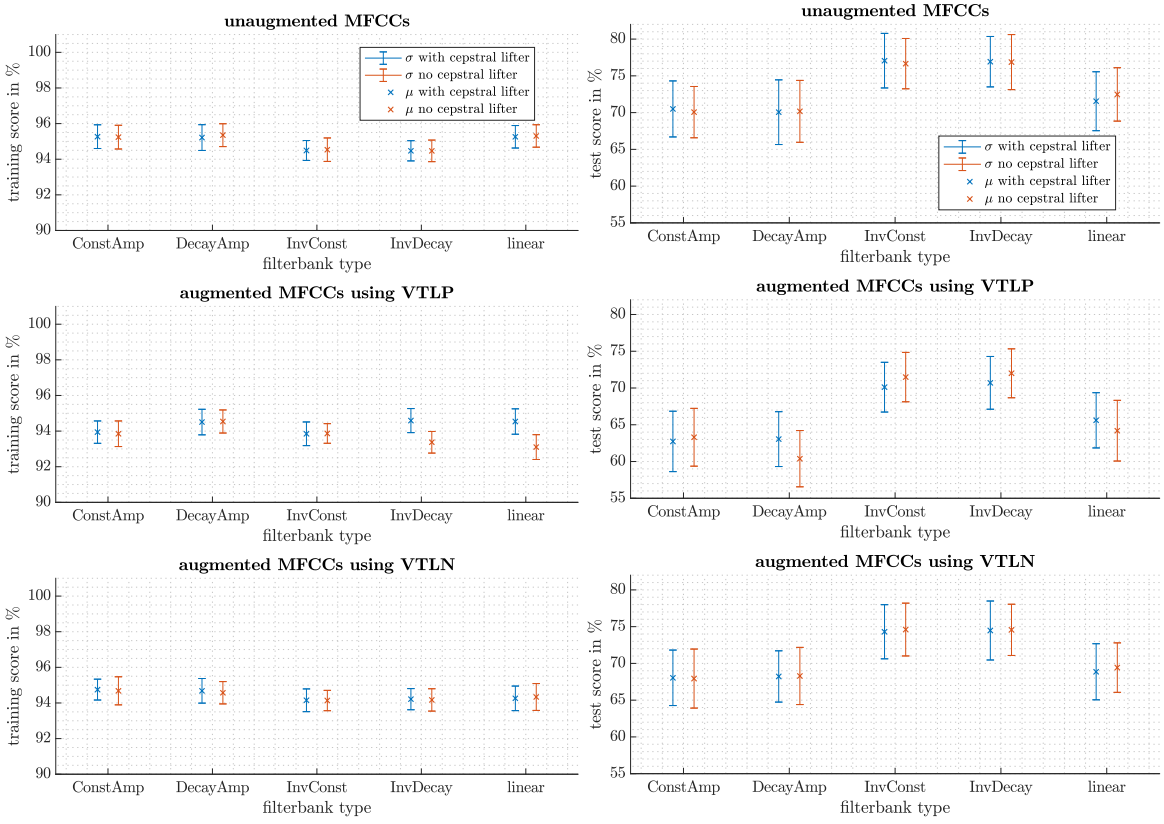
The last modification possibilty mentioned for the cepstral coefficient variants dealt within this thesis is given in form of a possible cepstral lifter. Again HTK [66] acts as a template and a cepstral lifter is implemented. The effects of the cepstral lifter were already indicated in Figure 2.5. The lifter curve depicted in Figure 2.5 is created by the proposed default lifter of [66, p.94], where a lifter according to Equation 2.9 with a lifter parameter $L = 22$ for $N_{\text{coeffs}} = 13$ coefficients is proposed. In order to adapt equal lifter behaviour for $N_{\text{coeffs}} = 36$ coefficients, which are used in the feature comparison carried out in this thesis, the lifter response is interpolated from 13 to 36 coefficients using the `Matlab` function `interp()` [32]. This lifter response is the one depicted in Figure 2.5. The liftering itself is then carried out by multiplication of cepstral coeffiecients and the lifter response, resulting in the scaling properties mentioned in subsection 2.2.3.

Performance overview

5 possible filterbank modifications as shown in Figure 2.2. 3 filterbank center frequency modifications (no modification, VTLN and VTLP) and a cepstral lifter that can be switched on and off, result in 30 available MFCC feature set variations. For every MFCC variant $N_{\text{coeffs}} = 36$ coefficients are calculated, but the zeroth coefficient is neglected, as it only provides information on the average log-energy of the signal, which does not hold much relevant information on the voice quality or the singer [52, p.87]. This leaves an overall number of $N_{\text{MFCCs}} = N_{\text{coeffs}} - 1 = 35$ coefficients which are further processed.

In order to limit the following analysis, which includes a feature selection algorithm, to one coefficient

feature set, a performance overview is created where the performance measures from subsection 3.3.1 are evaluated for all possible MFCC variations with 35 coefficients, using the dataset resulting from *the first dataset reduction* (see section 3.2). Due to random undersampling applied to balance the dataset as described in subsection 3.2.2 the underlying dataset is randomly composed. This introduces a random component into the classification process and hence the previously mentioned performance measures also exhibit a statistical variation. This is why the performance measures are evaluated by repeatedly executing the classification process presented in section 2.1 with a single stage SVM, whose kernel scale is determined with 500 iterations, as mentioned in section 3.3, the box constraint is fixed with 1. The classification itself using the SVM-classifier model (single stage) is executed with 100 iterations in order to create Figures 3.8 and 3.9, meaning the performance measures are caluclated 100 times. The mean μ and standard deviation σ are calculated using the `Matlab` commands `mean()` [35] and `std()` [41]. The first subplot of Figure 3.8 and Figure 3.9 depicts the resulting performance measures for the cepstral coefficients without center frequency modification (VTLN & VTLP). The second subplot depicts the performance measures for the cepstral coefficients, where the filterbank center frequencies are shifted using VTLP and in the third subplot, the results for the modified filterbank center frequencies using VTLN are depicted. The results dependent on the cepstral lifter are visualized in different colors. This enables a comparison of all 30 possible feature set variations within one plot. For each performance measure a separate plot is added. Figure 3.8 shows the *training* and *test score* in form of $\mu \pm \sigma$ achieved with each of the MFCC variants. The mean and standard deviation of the performance measures, indicated by markers and bars are placed over the filterbank variation abbreviation. Figure 3.9 shows the *misclassification rate* and the *overall score*.



(a) *training score* (b) *test score*

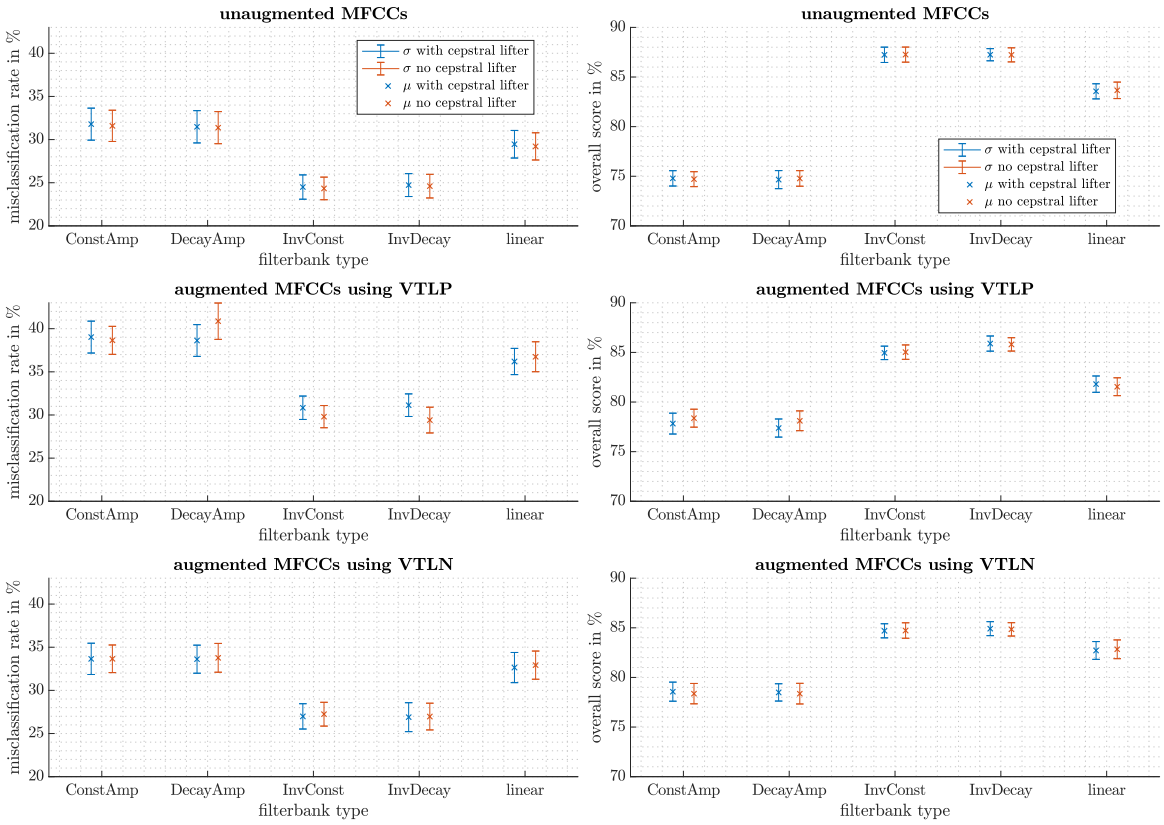
Figure 3.8 *Performance overview on MFCC variations: $\mu \pm \sigma$ of training (a) and test score (b) are evaluated with 100 classification iterations for the first dataset reduction, considering 35 augmented and unaugmented MFCCs respectively for various filterbanks.*

It is obvious that the center-frequency warping with VTLP worsens the performance. The train, test and *overall scores* in the second subplots of both Figure 3.8 and 3.9 are visibly lower in comparison to the unaugmented MFCCs, depicted in the first subplots. The *misclassification rate* is also higher for the VTLP augmented MFCCs, spanning a range from a maximum mean value of around $41\,\%$ (`DecayAmp` without cepstral liftering) to a minimum mean value of approximately $30\,\%$ (inverse filterbanks) for the VTLP augmented MFCCs, whereas the *misclassification rate* unaugmented MFCCs depicted in the first subplot of Figure 3.9(a), spans a range of $32\,\%$ - $24\,\%$. A lower *misclassification rate* and higher *test score* indicate better generalizability, which is given for the unaugmented MFCC variants, depicted in the first subplots of Figure 3.8 and 3.9, in comparison to the augmented MFCCs. The VTLP and VTLN augmented MFCCs, whose results are visualized in the second and third subplot of Figure 3.8 and 3.9, show lower *test scores* and higher *misclassification rate*s than the unaugmented MFCCs. The coefficients augmented with VTLN perform better than the coefficients augmented with VTLP. But compared to the unaugmented MFCC variants the VTLN does not exhibit improved performance. The VTLN augmentation yields a *misclassification rate* range from $34\,\% \pm 1.5\,\%$ to $27\,\% \pm 1.5\,\%$, whereas the unaugmented MFCCs result in a *misclassification rate* range of $33\,\% \pm 1.5\,\%$ to $24\,\% \pm 1.5\,\%$. The value ranges intersect, demonstrating no significant difference. But considering the extra computational effort behind the VTLN, the unaugmented coefficients are preferred.
Furthermore, when comparing the feature set variations in terms of the filterbank type for the unaugmented case (first subplot of Figure 3.8 and 3.9), the inverse filterbanks (`InvConst` & `InvDecay`) clearly outperform the other filterbanks. The train, test and *overall scores* exhibit higher percentages whereas the *misclassification rate* is way lower, compared to the other filterbank types. The inverse filterbanks reach a minimal *misclassification rate* mean of approx. $24\,\%$, which is $5\,\%$ less than the mean *misclassification rate* of the LFCCs (`linear`) and $8\,\%$ less than the mean *misclassification rate* of the MFCCs with the classic filterbank using a constant or decaying amplitude (`ConstAmp` & `DecayAmp`). Naturally, the maximal *test score* is also found for the inverse filterbanks with a score of $77\,\% \pm 3\,\%$, whereby σ is slightly deviating, which can be traced back to the statistical variation brought into the peformance measure by randomly picking and separating the data as mentioned in subsection 3.2.2. Generally, when looking at the random component within the values, it is visible that the highest standard deviations are found for the *test score*, which is due to the fact that the *test score* is only evaluated once for the hold-out set after the dataset is split with the ratio 80/20, as illustrated in subsection 3.2.2. The *misclassification rate*, on the other hand, is calculated through the crossvalidation process and is subject to averaging across the different runs, which gives the measure itself more statistical stability, resulting in a smaller standard deviation margin.

As the best performance is achieved with the inverse filterbanks and only one feature set variation is further processed and compared to the feature set derived from the modulation power spectrum presented in the next chapter, one feature set is chosen. Looking at the *misclassification rate*, the train score and the *overall score*, the results of the inverse filterbanks for the unaugmented coefficents are almost identical, also the influence of the cepstral lifter is very limited. But when looking at the *test score* of the unaugmented coefficients calculated with the inverse filterbank with a constant amplitude shows a slight improvement, if cepstral liftering is applied. Even if the reason of this increase lies in the previously mentioned emphasized statistical variation within the *test score* calculation, the inverse filterbank with constant amplitude in unaugmented form is used for further analysis, using the feature selection algorithm, whose results are presented in section 3.5. The cepstral lifter is also kept as this is also a default property of MFCCs calculated within the HTK implementation [66].

Another aspect that is analyzed and discussed is the rather large mismatch between the test and train

score observable in Figure 3.8, indicating overfitting of the used classification model.



(a) *misclassification rate* (b) *overall score*

Figure 3.9 *Performance overview on MFCC variations: $\mu \pm \sigma$ of misclassification rate (a) and overall score (b) are evaluated with 100 classification iterations for the first dataset reduction, considering 35 augmented and unaugmented MFCCs respectively for various filterbanks.*

The feature set containing the $N_{\text{MFCCs}} = 35$ cepstral coefficients, created using the inverse filterbank with constant amplitude and a cepstral lifter, are calculated for all N_{samples} according to Equation 3.1. In order to allow consistent usage of mathematical variables, the chosen MFCC variant from this point out are referred to as $\tilde{c}_{1-35}$ in accordance with Equation 2.9, and the dataset holding the 35 MFCCs calculated for each sample of a chosen dataset variation are noted as:

$$\mathscr{X}_{\text{MFCC}} = \{\tilde{\boldsymbol{c}}_1, \tilde{\boldsymbol{c}}_2, \ldots, \tilde{\boldsymbol{c}}_{35}\} = \begin{Bmatrix} \tilde{c}_1^{(1)}, & \tilde{c}_2^{(1)}, & \ldots, & \tilde{c}_{35}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{c}_1^{(N_{\text{samples}})}, & \tilde{c}_2^{(N_{\text{samples}})}, & \ldots, & \tilde{c}_{35}^{(N_{\text{samples}})} \end{Bmatrix} \tag{3.11}$$

The bold notation, indicating the compendium of all 35 coefficients for all N_{samples}, depending on the used dataset variation as presented in Equation 3.1, is also upheld in section 3.5, where the results of the feature selection algorithm are presented.

3.4.2 Fundamental Frequency

The fundamental frequency is vital for the calculation of the MPS based features mentioned in the next chapter. Although the information on the requested pitch during the recording process of the dataset

mentioned in section 3.1 is present, a pitch tracking algorithm is implemented in order to intercept slight pitch deviations that might have occured during recording. Another point, why a fundamental frequency estimation was chosen, rather than using the reference pitches, is that for a possible future real-time implementation a pitch estimation algorithm might be crucial and the implemented pitch tracker can be viewed as a template implementation. As mentioned in section 2.4 the underlying pitch tracker was implemented as proposed in [9]. The limitations of the used SRH pitch estimation technique is extensively discussed in [4]. In order to assess the performance of the F0-tracker on the audio samples contained in the dataset discussed in section 3.1, an estimation error measure is introduced in Equation 3.12.

$$e_{f_0} = \left| \frac{\hat{f}_0 - f_{0,\text{true}}}{f_{0,\text{true}}} \right| \cdot 100\,\% \tag{3.12}$$

$\hat{f}_0$ denotes the estimated mean fundamental frequency of a sung vocal sample and $f_{0,\text{true}}$ is one of the reference pitches listed in Table 3.1. The error measure e_{f_0} is compared to a tolerated error threshold, which was fixed as an approximate halftone deviation[1], resulting in a threshold of:

$$e_{\text{tol}} = \left| 1 - 2^{\frac{1}{12}} \right| \cdot 100\,\% \approx 5.95\,\% \tag{3.13}$$

Figure 3.10 shows the calculated estimation error of each file. The various pitches of the files are coloured differently and the file indices, entered on the x-axis, indicate the last sample, before the error measures of samples containing different pitches are plotted.

Figure 3.10 *Pitch-tracking performance before correction.*

There are exactly 32 files for which the tolerated error threshold is exceeded. Considering the halftone error threshold this yields $p_{f_0} = \left(1 - \frac{32 \text{ samples}}{1140 \text{ samples}}\right) \cdot 100\,\% \approx 97.2\,\%$ of the samples where the estimated pitch lies below the error threshold. The estimated pitches of the 32 samples that exceed the halftone threshold mostly exhibit error measures of $50\,\%$ or even $200\,\%$ and interestingly enough the

[1] The threshold in percent does not exactly correspond to $\pm$ one semitone, as $5.95\,\%$ amount to a little bit more than a semitone downwards, because: $|1-2^{\frac{1}{12}}| \approx 5.95\,\% \neq |1-2^{-\frac{1}{12}}| \approx 5.61\,\%$. However, the slight deviation is neglected.

samples which exhibit the highest errors are of pitch a^2, the highest pitch contained in the dataset. It can be anticipated that such high deviations in pitch are the result of misestimations of the pitch tracker and were not caused during the recording process by the professional singers. If one takes the pitches listed in Table 3.1 as the groundtruth, a $50\,\%$ error corresponds to a pitch that is estimated an octave too low and an error of $200\,\%$ indicates that the fundamental frequency has been estimated as 3 times the reference pitch, indicating that the third harmonic probably has been mistaken for the fundamental frequency.

Due to the fact that the information on the reference pitch of each sample is provided with the dataset, each of the 32 estimated pitches, exhibiting exceeding errors are corrected with the reference pitch. If the error measure from Equation 3.12 is now computed again and the results are illustrated in Figure 3.11. It is evident that there are now no more outliers and all of the estimated fundamental frequencies are now below the halftone error threshold. When looking at the calculated error mea-

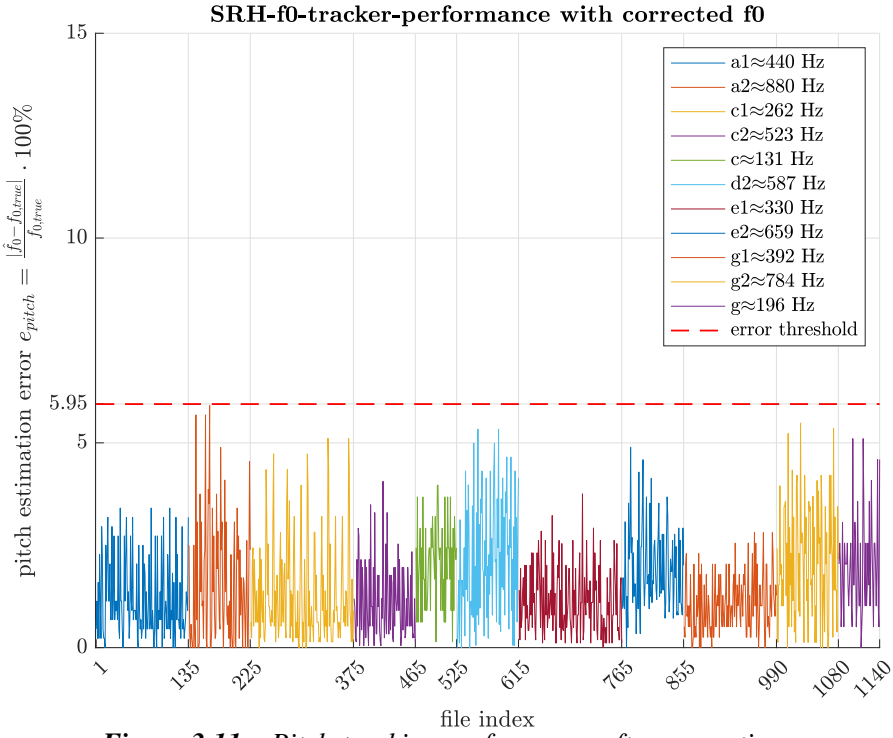


Figure 3.11 *Pitch-tracking performance after correction.*

sures in Figure 3.11, it becomes visible that the pitches a^1, c^2, e^1 and g^1, which span a frequency range of approx. $330\,\text{Hz}, \ldots, 520\,\text{Hz}$, exhibit the lowest errors. This is explainable with the fact that the dataset consists of samples predominantly sung by female singers, and the mentioned pitch range corresponds to the moderate lower-mid pitch range of a mezzosoprano, with neither extremely high, nor extremely low pitches, yielding better pitch controllability [14, p. 132]. However, the SRH pitch tracker also comes with limitations in regards to higher fundamental frequencies as shown in [4] and thus, the higher errors for higher pitches, might also be partially caused by the chosen pitch tracker. A more thorough analysis of correlations between pitch deviations and singers or voice qualities is not carried out, so the origins of the present pitch deviations are not fully uncovered.
Regarding the notation, the estimated fundamental frequencies for N_{samples} of a analysed dataset variation are noted with $\hat{\boldsymbol{f}}_0 = \left\{\hat{f}_{0,1}, \hat{f}_{0,2}, \ldots, \hat{f}_{0,N_{\text{samples}}}\right\}$. In addition to the usage of the estimated fundamental frequencies within the MPS feature calculation, mentioned in subsection 3.4.3 $\hat{\boldsymbol{f}}_0$ is also incooperated in the classification procedure of section 3.5 as a voice quality feature.

3.4.3 Summed Modulation Power Spectrum Features

In section 2.3 the modulation power spectrum is theoretically and mathematically described and it ends with the definition of the summed modulation power spectra which build the basis of the features that are derived in this section. The MPS, summed along the temporal modulation and spectral modulation axis results in:

- the summed temporal modulation power spectrum (STMPS): $\hat{S}_\Sigma\left(f_{\text{t}_{\text{mod}}}\right)$
- the summed spectral modulation power spectrum (SSMPS): $\hat{S}_\Sigma\left(\tau\right)$

An exemplary depiction of the $\hat{S}_\Sigma\left(f_{\text{t}_{\text{mod}}}\right)$ and $\hat{S}_\Sigma\left(\tau\right)$ can be found in Figure 2.8. As shown in Figure 2.8, the summed modulation power spectra both in the temporal and spectral domain exhibit decreasing trend for higher modulations. The underlying idea on the feature extraction is to focus on the discussion of the summed MPS' peaks, e.g. differences in height or a voice quality dependent decrease etc. This is done more easily if the peaks are placed on an equal level and the decreasing trend towards higher modulations is compensated.

Polynomial fitting

The decreasing trend is approximated by application of polynomial fitting using `Matlab's polyfit()` command [37]. With a polynomial order $N_{\text{poly}} = 3$, the function `polyfit()` solves for the polynomial:

$$y(x) \approx p(x) = p_1 x^3 + p_2 x^2 + p_3 x + p_4 \tag{3.14}$$

by solving the linear equation system

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{p}$$

$$\begin{pmatrix} y(x_1) \\ y(x_2) \\ \vdots \\ y(x_M) \end{pmatrix} = \begin{bmatrix} x_1^3 & x_1^2 & x_1^1 & 1 \\ x_2^3 & x_2^2 & x_2^1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_M^3 & x_M^2 & x_M^1 & 1 \end{bmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} \tag{3.15}$$

with regards to the polynomial coefficients contained in $\boldsymbol{p}$ [37]. $\boldsymbol{X}$ is a so-called Vandermode matrix [49]. Equation 3.15 is solved for the polynomial coefficients $\boldsymbol{p}$, within `Matlab's polyfit()` using the backslash operator [36]:

$$\boldsymbol{p} = \boldsymbol{X} \backslash \boldsymbol{y} \tag{3.16}$$

`Matlab's` backslash operator from [36] employs a QR matrix decomposition in order to obtain the solution of this equation system. The solution delivers a least-squares polynomial fit of points defined by $\boldsymbol{x} = [x_1, \ldots, x_M]^T$, which are the x-axis values and $\boldsymbol{y} = [y(x_1), \ldots, y(x_M)]^T$, the corresponding y-axis values that are to be approximated with a polynomial. M defines the number of the available points [37], that are used in the polynomial fitting process.

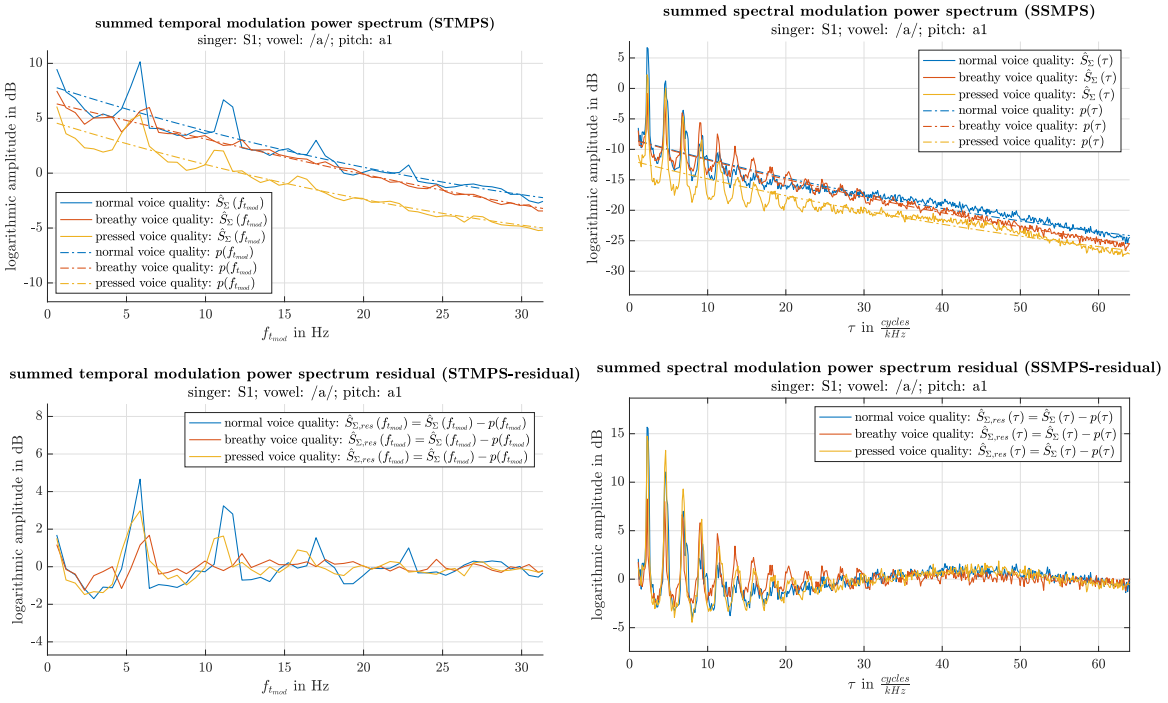
The polynomial fitting is carried out for both the STMPS and the SSMPS. This is done by solving Equation 3.16 with $\boldsymbol{x} = [f_{\text{t}_{\text{mod}}1}, \ldots, f_{\text{t}_{\text{mod}},M}]^T$ and $\boldsymbol{y} = \left[\hat{S}_\Sigma\left(f_{\text{t}_{\text{mod}},1}\right), \ldots, \hat{S}_\Sigma\left(f_{\text{t}_{\text{mod}},M}\right)\right]^T$ for the polynomial fit of the summed temporal modulation power spectrum $p(f_{\text{t}_{\text{mod}}})$. If $\boldsymbol{x} = [\tau_1, \ldots, \tau_M]^T$ and $\boldsymbol{y} = \left[\hat{S}_\Sigma\left(\tau_1\right), \ldots, \hat{S}_\Sigma\left(\tau_M\right)\right]^T$ are used, the polynomial fit of the summed spectral modulation power spectrum $p(\tau)$ is retrieved. The resulting estimated polynomials are then given with:

$$\begin{aligned} p(f_{\text{t}_{\text{mod}}}) &= p_1 f_{\text{t}_{\text{mod}}}^3 + p_2 f_{\text{t}_{\text{mod}}}^2 + p_3 f_{\text{t}_{\text{mod}}} + p_4 \\ p(\tau) &= p_1 \tau^3 + p_2 \tau^2 + p_3 \tau + p_4 \end{aligned} \tag{3.17}$$

It is important to note that for the fitting of both polynomials $p(f_{\text{t}_{\text{mod}}})$ and $p(\tau)$ the zero modulation components were discarded, as they would add an unnecessary offset to the fitted polynomial, distorting the summed modulation power spectra's trend estimation. Concerning the summed temporal modulations, all points for which $f_{\text{t}_{\text{mod}}} < 0.5\,\text{Hz}$ are neglected. For the summed spectral modulation power spectrum's polynomial fitting the points for $\tau < 1.1\frac{\text{cycles}}{\text{kHz}}$ were neglected. To remove the trend and to place the peaks on a relatively equal level, enabling a better assessment of several peak parameters, the approximated trend using polynomial fitting and the summed modulation power spectra (STMPS & SSMPS) are subtracted, resulting in the STMPS- and SSMPS-residual $\hat{S}_{\Sigma,\text{res}}\left(f_{\text{t}_{\text{mod}}}\right)$ and $\hat{S}_{\Sigma,\text{res}}\left(\tau\right)$. The summed modulation power spectrum residuals are formulated as:

$$\begin{aligned}\hat{S}_{\Sigma,\text{res}}\left(f_{\text{t}_{\text{mod}}}\right) &= \hat{S}_{\Sigma}\left(f_{\text{t}_{\text{mod}}}\right) - p(f_{\text{t}_{\text{mod}}}) \\ \hat{S}_{\Sigma,\text{res}}\left(\tau\right) &= \hat{S}_{\Sigma}\left(\tau\right) - p(\tau)\end{aligned} \tag{3.18}$$

An exemplary visualization is given in Figure 3.12. In the first subplot of Figure 3.12 (a) the STMPS is shown, whereas the first subplot of Figure 3.12 (b) shows the SSMPS. The fitted polynomials are plotted as dashed dotted lines. The residuals $\hat{S}_{\Sigma,\text{res}}\left(\tau\right)$ and $\hat{S}_{\Sigma,\text{res}}\left(f_{\text{t}_{\text{mod}}}\right)$ are visualized in the lower subplots of Figure 3.12. It is clearly visible that all peaks are now located on the same reference level and emerge from ca. $0\,\text{dB}$. The STMPS-residual is also calculated for negative modulation



(a) STMPS, polynomial fitting and STMPS-residual

(b) SSMPS, polynomial fitting and SSMPS-residual

Figure 3.12 *Exemplary depiction of STMPS and SSMPS with polynomial fitting and resulting residuals.*

frequencies ($f_{\text{t}_{\text{mod}}} < 0\,\text{Hz}$) but it is not displayed in Figure 3.12 (a), because only the peaks of the positive half are further processed.

It is important to note that the correlation between the MPS-based features and the voice qualities described in this paragraph are not explicitly analyzed for the combination of all vowels, pitches and singers, as this would result in a lot of combinations. Nevertheless, an exemplary analysis of several samples is carried out, highlighting observations that indicate a correlation towards the voice

quality. This is the reason why the MPS based features, dealt with in this section, can be viewed as educated guesses, describing the exemplarily observed relations between the sung voice quality and observations made within the summed modulation power spectrum residuals. The discussed observations are mostly based on the description and dicussion of the summed modulation power spectrum residuals' peaks, which serve as the foundation for the MPS-based feature extraction. To retrieve the peak height and position of the STMPS- and SSMPS-residual, depicted in Figure 3.12, different peak picking strategies are executed for each of the residuals.

STMPS-residual peak based features

The underlying assumption, which the features derived from the STMPS-residuals are based on, is the assumption that the voice qualities have an effect on the perceived vibrato, e.g. when using *breathy* phonation there is less tension on the vocal folds and the surrounding muscular structure, which makes it more difficult to produce a distinct vibrato. As the temporal modulations describe the vibrato present in a sung vocal recording, it is assumed that features derived from STMPS-residuals exhibit a discriminatory power, towards the sung phonation type. Hence, the STMPS-residuals and their peaks are analyzed to derive 3 features. With the `Matlab` command `findpeaks()` [30] the 6 most prominent peaks of the STMPS-residual are determined, over the whole range of temporal modulations (positive and negative). However, the six peaks are reduced to the peaks located on the positive half within the interval $0\,\mathrm{Hz} < f_{\mathrm{t_{mod}}} < 20\,\mathrm{Hz}$ of the STMPS-residual. The peaks are then sorted in descending order, which allows further processing into the phonation descriptive abstract features. The number of the picked STMPS-residual peaks can vary from sample to sample, as there are not always as many peaks in $-20\,\mathrm{Hz} < f_{\mathrm{t_{mod}}} < 0\,\mathrm{Hz}$ as there are in the interval $0\,\mathrm{Hz} < f_{\mathrm{t_{mod}}} < 20\,\mathrm{Hz}$. Mathematically, the mentioned set of peaks containing $\check{N}_{\mathrm{temp}}$ peaks are denoted as:

$$\begin{aligned}\hat{\boldsymbol{S}}^{\mathrm{pk}}_{\Sigma,\,\mathrm{temp}} &= \left\{ \hat{S}^{\mathrm{pk},1}_{\Sigma,\,\mathrm{temp}}, \hat{S}^{\mathrm{pk},2}_{\Sigma,\,\mathrm{temp}}, \ldots, \hat{S}^{\mathrm{pk},\check{N}_{\mathrm{temp}}}_{\Sigma,\,\mathrm{temp}} \right\} = \\ &= \left\{ \hat{S}_{\Sigma,\mathrm{res}}\left(f^{\mathrm{pk},1}_{\mathrm{t_{mod}}}\right), \hat{S}_{\Sigma,\mathrm{res}}\left(f^{\mathrm{pk},2}_{\mathrm{t_{mod}}}\right), \ldots, \hat{S}_{\Sigma,\mathrm{res}}\left(f^{\mathrm{pk},\check{N}_{\mathrm{temp}}}_{\mathrm{t_{mod}}}\right) \right\}\end{aligned} \tag{3.19}$$

and the temporal modulation frequencies at which the STMPS-residual peaks occur are written as $f^{\mathrm{pk},i}_{\mathrm{t_{mod}}}\ \forall i = 1, 2, \ldots, \check{N}_{\mathrm{temp}}$.

The picked peaks of the STMPS-residual for a sung vocal sample containing the vowel */a/* and a pitch of $a1 \approx 440\,\mathrm{Hz}$, for two singers (S1 & S6) are visualized in the first subplots of Figure 3.13. The temporal modulations which comprise the modulations, defining a vibrato, namely amplitude (shimmer) and frequency modulation (jitter), exhibit one aspect that suggests a correlation with the phonation type. When comparing the STMPS-residual depicted in Figure 3.13 with regards to the voice quality, it is clearly visible that for both singers no distinct peaks arise for *breathy* phonation. This suggests that a vibrato is only present in a non-distinct or weakened state if *breathy* voice quality is used while singing. On the other hand *normal* voice quality seems to exhibit the most distinct peaks, there are even distinct peaks visible, at the double and three time multiple of the vibrato frequency, alongside with the first peak. This indicates that the vibrato does not occur as a single sinusoidal component within the sung vocal signal, but rather as a combination of multiple sinusoidal components. The STMPS-residuals and the picked peaks for the sung vocal samples, with the same vowel and pitch, but for different singers can be found in the Appendix in section B.2. With the exception of singer S7 the STMPS-residuals shown in section B.2, mostly exhibit the same properties. These exemplarily discussed observations, seem to correlate more with *breathy* phonation, the distinction of *normal* and *pressed* phonation using STMPS-residuals is not as clear as the one for *breathy* voice quality. However, the feature selection results presented in section 3.5 allow an evaluation of the calculated features. Concerning the STMPS-residual peaks the following features were calculated:

1. The height of the first (highest) STMPS-residual peak:

$$\hat{S}^{\text{pk,1}}_{\Sigma,\text{temp}} = \hat{S}_{\Sigma,\text{res}}\left(f^{\text{pk,1}}_{\text{t}_\text{mod}}\right) \tag{3.20}$$

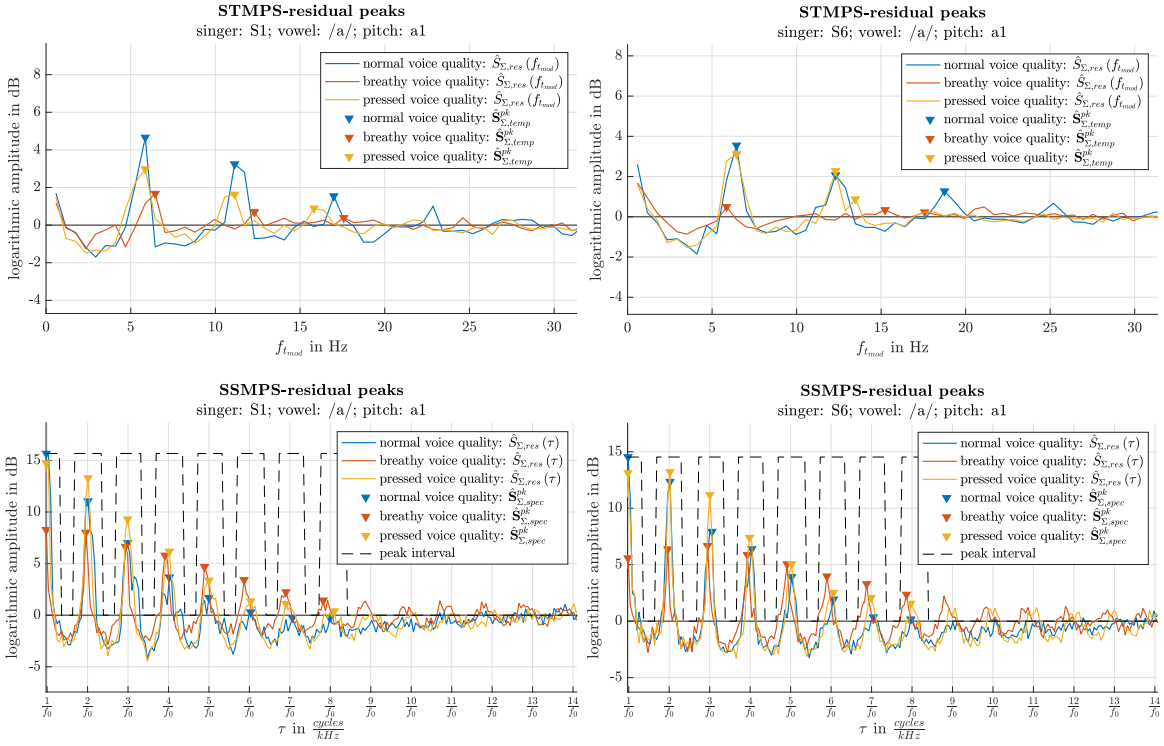
2. Difference between the first (highest) and second STMPS-residual peak:

$$\Delta_{\text{temp}} = \hat{S}^{\text{pk,1}}_{\Sigma,\text{temp}} - \hat{S}^{\text{pk,2}}_{\Sigma,\text{temp}} = \hat{S}_{\Sigma,\text{res}}\left(f^{\text{pk,1}}_{\text{t}_\text{mod}}\right) - \hat{S}_{\Sigma,\text{res}}\left(f^{\text{pk,2}}_{\text{t}_\text{mod}}\right) \tag{3.21}$$

3. The sum of all amplitude values of the STMPS-residual within the previously defined positive interval of interest:

$$\Sigma_{\text{temp}} = \sum_{f_{\text{t}_\text{mod}}} \hat{S}_{\Sigma,\text{res}}\left(f_{\text{t}_\text{mod}}\right) \;:\; 0\,\text{Hz} < f_{\text{t}_\text{mod}} < 20\,\text{Hz} \tag{3.22}$$

In order to provide transparent notation in this thesis, it is important to note that the usage of bold notation on feature variables e.g. $\hat{\boldsymbol{S}}^{\text{pk,1}}_{\Sigma,\text{temp}}$, $\boldsymbol{\Delta}_{\text{temp}}$ or $\boldsymbol{\Sigma}_{\text{temp}}$, indicates the comprisal of the respective features for all samples contained in a dataset variation. This leads to a number of N_{samples} features, which depends on the chosen dataset variation, according to Equation 3.1.



(a) $\hat{S}_{\Sigma,\text{res}}(\tau)$ and $\hat{S}_{\Sigma,\text{res}}(f_{\text{t}_\text{mod}})$ with picked peaks for singer: S1, vowel: */a/*

(b) $\hat{S}_{\Sigma,\text{res}}(\tau)$ and $\hat{S}_{\Sigma,\text{res}}(f_{\text{t}_\text{mod}})$ with picked peaks for singer: S6, vowel: */a/*

Figure 3.13 *Picked peaks of STMPS and SSMPS-residual for exemplarily chosen vocal samples.*

SSMPS-residual peak based features

Because the STMPS-based features seem to correlate more with the *breathy* voice quality the second axis of the modulation power spectrum is also considered.

Analogous to the STMPS-residual features, the first step of the SSMPS-residual feature extraction is peak picking. The peak picking strategy applied on the SSMPS-residuals differs from the peak picking carried out for the STMPS-residuals, because in case of spectral modulation residuals, the fundamental frequency information is usable. The peaks in the summed spectral modulation power spectrum residuals form around multiples of the fundamental period, which makes it possible to localize peaks more specifically, if a fundamental frequency estimate exists. Note that the SSMPS-residual is no frequency spectrum and the values depcited on the x-axis are no frequencies, but the cepstral equivalent called quefrencies (see section 2.3). The spectral modulations building the y-axis of a MPS hold information on the composition and relations of the overtone spectrum. More details on the interpretation of the temporal and spectral modulations are found in section 2.3.

In order to determine the peaks of the SSMPS-residual, peak intervals centered around the values $\tau = \frac{1}{\hat{f}_0} \cdot n \,\forall\, n \in \mathbb{N}$ are scanned for the maximum value. The peak intervals for SSMPS-residual calculated for one sample are defined as:

$$\begin{aligned} \tau_{\text{start},n} &= \frac{n}{\hat{f}_0} - \frac{1}{3\hat{f}_0} \,\forall\, n = 1, 2, \ldots, \check{N}_{\text{spec}} \\ \tau_{\text{stop},n} &= \frac{n}{\hat{f}_0} + \frac{1}{3\hat{f}_0} \,\forall\, n = 1, 2, \ldots, \check{N}_{\text{spec}} \end{aligned} \tag{3.23}$$

To derive the SSMPS-residual based features $\check{N}_{\text{spec}} = 8$, peaks are picked within the interval defined by τ_{start} and τ_{stop} using the max-operator:

$$\begin{aligned} \hat{\boldsymbol{S}}^{\text{pk}}_{\Sigma,\,\text{spec}} &= \max\left(\hat{S}_{\Sigma,\text{res}}\left(\tau\right)\right)\Big|_{\tau_{\text{stop}}}^{\tau_{\text{start}}} \quad \forall n = 1, 2, \ldots, 8 \\ \hat{\boldsymbol{S}}^{\text{pk}}_{\Sigma,\,\text{spec}} &= \left\{\hat{S}^{\text{pk},1}_{\Sigma,\,\text{spec}}, \hat{S}^{\text{pk},2}_{\Sigma,\,\text{spec}}, \ldots, \hat{S}^{\text{pk},8}_{\Sigma,\,\text{spec}}\right\} = \left\{\hat{S}_{\Sigma,\text{res}}\left(\tau^{\text{pk},1}\right), \hat{S}_{\Sigma,\text{res}}\left(\tau^{\text{pk},2}\right), \ldots, \hat{S}_{\Sigma,\text{res}}\left(\tau^{\text{pk},8}\right)\right\} \end{aligned} \tag{3.24}$$

The spectral modulations, within the peak intervals, at which the peaks are located are written as:

$$\tau^{\text{pk},n} \in [\tau_{\text{start},n}, \tau_{\text{stop},n}] \,\forall\, n = 1, 2, \ldots, \check{N}_{\text{spec}} \tag{3.25}$$

In order to carry through the exemplary analysis of the SSMPS-residuals, the second subplots of Figure 3.13 and Figure B.2-B.5 contain the SSMPS-residuals of samples, where the requested vowel /a/ was sung with *normal* voice quality at a pitch of $\text{a}^1 \approx 440\,\text{Hz}$. If one takes a look at the peak formed at $\tau^{\text{pk},2} = \frac{2}{\hat{f}_0}$ it is visible that the highest peak amongst all three voice qualities is given for *pressed* phonation, followed by the peak height of *normal* and *breathy* phonation. This is visible for the SSMPS-residuals of both singers depicted in Figure 3.13, as well as for most of the plots added in section B.2. Another aspect that is observable for the peaks of the SSMPS-residual is that the peak heights display different levels of decrease depending on the spectral modulation. The SSMPS-residual's peak amplitudes of *normal* voice quality decrease faster for lower modulations ($\frac{2}{\hat{f}_0}$ to $\frac{5}{\hat{f}_0}$) in contrast to the peak height decrease for higher modulations ($\frac{5}{\hat{f}_0}$ to $\frac{8}{\hat{f}_0}$). The peak amplitudes for *breathy* voice quality, decrease less pronounced over all spectral modulations, they evidently show lower peak heights than the peaks visible for other phonation types. *Pressed* voice quality starts

with higher amplitude peaks at lower modulations ($\frac{2}{\hat{f}_0}$ to $\frac{5}{\hat{f}_0}$) than for higher modulations ($\frac{5}{\hat{f}_0}$ to $\frac{8}{\hat{f}_0}$), as shown in Figure 3.13 but also in Figures B.2, B.5 and B.3, thus indicating that the peak-height decrease for *pressed* voice quality viewed over all picked peaks is different from the overall difference for other phonation types. Based on these observations concerning the SSMPS-residual's peak height, peak decreases as well as the relation of the peak differences towards lower and higher modulations, the following six abstract features are calculated in addition to the three STMPS-residual peak based features, mentioned in the previous paragraph.

1. The second peak of the SSMPS-residual located at $\tau^{\text{pk},2} = \frac{2}{\hat{f}_0}$

$$\hat{S}^{\text{pk},2}_{\Sigma,\text{spec}} = \hat{S}_{\Sigma,\text{res}}\left(\tau^{\text{pk},2}\right) \tag{3.26}$$

2. The average peak difference of the peak located at $\tau^{\text{pk},2} = \frac{2}{\hat{f}_0}$ to $\tau^{\text{pk},5} = \frac{5}{\hat{f}_0}$

$$\bar{\Delta}_{\text{spec},1} = \frac{\left|\hat{S}^{\text{pk},2}_{\Sigma,\text{spec}} - \hat{S}^{\text{pk},3}_{\Sigma,\text{spec}}\right| + \left|\hat{S}^{\text{pk},3}_{\Sigma,\text{spec}} - \hat{S}^{\text{pk},4}_{\Sigma,\text{spec}}\right| + \left|\hat{S}^{\text{pk},4}_{\Sigma,\text{spec}} - \hat{S}^{\text{pk},5}_{\Sigma,\text{spec}}\right|}{3} \tag{3.27}$$

3. The average peak difference of the peak located at $\tau^{\text{pk},5} = \frac{5}{\hat{f}_0}$ to $\tau^{\text{pk},8} = \frac{8}{\hat{f}_0}$

$$\bar{\Delta}_{\text{spec},2} = \frac{\left|\hat{S}^{\text{pk},5}_{\Sigma,\text{spec}} - \hat{S}^{\text{pk},6}_{\Sigma,\text{spec}}\right| + \left|\hat{S}^{\text{pk},6}_{\Sigma,\text{spec}} - \hat{S}^{\text{pk},7}_{\Sigma,\text{spec}}\right| + \left|\hat{S}^{\text{pk},7}_{\Sigma,\text{spec}} - \hat{S}^{\text{pk},8}_{\Sigma,\text{spec}}\right|}{3} \tag{3.28}$$

4. The logarithmized ratio of the previously defined peak difference measures $\bar{\Delta}_{\text{spec},1}$ and $\bar{\Delta}_{\text{spec},2}$

$$\bar{\Delta}_{\text{ratio}} = \log_{10}\left(\frac{\bar{\Delta}_{\text{spec},2}}{\bar{\Delta}_{\text{spec},1}}\right) \tag{3.29}$$

5. The overall peak difference calculated as the difference between the second and eighth peak of the SSMPS-residual

$$\bar{\Delta}_{\text{overall}} = \hat{S}^{\text{pk},2}_{\Sigma,\text{spec}} - \hat{S}^{\text{pk},8}_{\Sigma,\text{spec}} \tag{3.30}$$

6. The sum of all eight SSMPS-residual peaks that were picked.

$$\Sigma^{pk}_{\text{spec}} = \sum_{i=1}^{8} \hat{S}^{\text{pk},i}_{\Sigma,\text{spec}} = \sum_{i=1}^{8} \hat{S}_{\Sigma,\text{res}}\left(\tau^{\text{pk},i}\right) \tag{3.31}$$

This completes the MPS based feature set consisting of 9 descriptive measures that are calculated on voice quality dependent observations, which are made from the summed temporal and spectral modulation power spectrum. It is worth noting again: that if the feature variables are written in bold notation, they are to be seen as variables which contain all N_{samples} of the dataset variation according to Equation 3.1. For the feature selection analysis stated in the next chapter, the MPS-based feature set, calculated for all N_{samples} according to Equation 3.1 of a dataset variation, is written as:

$$\mathscr{X}_{\text{MPS}} = \left\{\hat{\boldsymbol{S}}^{\text{pk},1}_{\Sigma,\text{temp}}, \boldsymbol{\Delta}_{\text{temp}}, \boldsymbol{\Sigma}_{\text{temp}}, \hat{\boldsymbol{S}}^{\text{pk},2}_{\Sigma,\text{spec}}, \bar{\boldsymbol{\Delta}}_{\text{spec},1}, \bar{\boldsymbol{\Delta}}_{\text{spec},2}, \bar{\boldsymbol{\Delta}}_{\text{ratio}}, \bar{\boldsymbol{\Delta}}_{\text{overall}}, \boldsymbol{\Sigma}^{pk}_{\text{spec}}\right\} \tag{3.32}$$

In order to shortly explain the usage of the bold notation in Equation 3.32, Equation 3.33 is added. Equation 3.33 uses $\bar{\boldsymbol{\Delta}}_{\text{ratio}}$ as an example and it shows that $\bar{\boldsymbol{\Delta}}_{\text{ratio}}$ is calculated for each sample of a dataset variation. This also holds for all the other features contained in the feature set $\mathscr{X}_{\text{MPS}}$ from Equation 3.32, but also for the MFCC feature set $\mathscr{X}_{\text{MFCC}}$ from Equation 3.11.

$$\bar{\boldsymbol{\Delta}}_{\text{ratio}} = \left\{\bar{\Delta}^{(1)}_{\text{ratio}}, \bar{\Delta}^{(2)}_{\text{ratio}}, \ldots, \bar{\Delta}^{(N_{\text{samples}})}_{\text{ratio}}\right\} \tag{3.33}$$

3.5 Performance Overview with Feature Selection

The feature sets $\mathscr{X}_{\text{MFCC}}$ from Equation 3.11 and $\mathscr{X}_{\text{MPS}}$ from Equation 3.32, as well as a combined version comprising both the 9 MPS-based features as well as the 35 MFCC features. The combined feature set is denoted as:

$$\mathscr{X}_{\text{combo}} = \{\mathscr{X}_{\text{MPS}}, \mathscr{X}_{\text{MFCC}}\}$$

$$\mathscr{X}_{\text{combo}} = \left\{\hat{\boldsymbol{S}}^{\text{pk},1}_{\Sigma,\text{temp}}, \boldsymbol{\Delta}_{\text{temp}}, \boldsymbol{\Sigma}_{\text{temp}}, \hat{\boldsymbol{S}}^{\text{pk},2}_{\Sigma,\text{spec}}, \bar{\boldsymbol{\Delta}}_{\text{spec},1}, \bar{\boldsymbol{\Delta}}_{\text{spec},2}, \bar{\boldsymbol{\Delta}}_{\text{ratio}}, \bar{\boldsymbol{\Delta}}_{\text{overall}}, \boldsymbol{\Sigma}^{pk}_{\text{spec}}, \tilde{\boldsymbol{c}}_1, \ldots, \tilde{\boldsymbol{c}}_{35}\right\} \tag{3.34}$$

Before $\mathscr{X}_{\text{MFCC}}$, $\mathscr{X}_{\text{MPS}}$ and $\mathscr{X}_{\text{combo}}$ are analysed using the Plus-L Minus-R feature selection algorithm mentioned in section 2.6. The feature sets are normalized to standard z-scores, using `Matlab's zscore()` [43]. The standardization using `zscore()` effectuates that the feature sets' mean equals 0 and the standard deviation amounts to 1.
For the single stage SVM model, all three feature sets are processed in order to evaluate each feature set's capabilities on the classification of all three voice qualities, seperately. This enables a ranking of features which contribute most towards the distinction of the three voice qualities: *breathy*, *normal* and *pressed*. The analysis with regards to the two stage classification model is carried out soley with the combined feature set $\mathscr{X}_{\text{combo}}$, with the idea that through the feature selection algorithm, applied on both SVM stages the Plus-L Minus-R algorithm chooses the most suitable features for each classification problem separately.

Finally, it is important to note that the curves depicted in the plots of subsections 3.5.2 and 3.5.3 are all subject to a certain statistical variance. This is due to the situation that the balancing of the data presented in subsection 3.2.2 is subject to random undersampling processes, which introduce a statistical variation into classification process. Due to this the classification process necessary to calculate the performance measures as explained in subsection 3.3.1 was carried out 10 times and the mean value is depicted in the figures of subsection 3.5.2 and subsection 3.5.3. This is done to present statistically stabilized measures. Nevertheless, the highest statistical variation is present for the *test score* but at no point in the analysis, a standard deviation of $\pm 5\,\%$ is exceeded. Due to the fact that this is the grid spacing of the following figures the standard deviation of each measure is neglected and also the deviation decreases with an increasing number of selected features or data samples. The other measure which holds information on the generalizabilty of the classification model is the *misclassification rate* which due to the intrinsic averaging process necessary for its calculation as depicted in Figure 3.5, already proves a certain degree of statistical stability amounting to deviations of less than $\pm 2\,\%$.

After a short introduction on how the Plus-L Minus-R algorithm is implemented using `Matlab` and which parameters are chosen for the algorithm, the feature selection results are summarized in the figures inserted in subsection 3.5.2 and subsection 3.5.3. The results are evaluated for the *first*, *second* and *third* dataset reduction.

3.5.1 Implementation of the Plus-L Minus-R feature selection algorithm

As mentioned in section 2.6 the Plus-L Minus-R feature selection (L-R selection) algorithm can be viewed as the L-times execution of a sequential forward selection (SFS) and afterwards the R-times execution of a sequential backward selection (SBS). The aim of the feature selection, carried out in this thesis, is to exploit every possible dimensionality reduced feature set and calculate the resulting performance measures introduced in subsection 3.3.1. This means that the L-R selection is executed iteratively for $N_{\text{FS}} = 1, 2, \ldots, (N_{\text{feat}} - 1)$, where N_{FS} is the reduced feature set's number of features

and N_{feat} is the number of features in the complete feature set, which for the three possible feature sets is given with:

$$N_{\text{feat}} = \begin{cases} 35, & \text{for} \mathscr{X}_{\text{MFCC}} \\ 9, & \text{for} \mathscr{X}_{\text{MPS}} \\ 44, & \text{for } \mathscr{X}_{\text{combo}} \end{cases} \tag{3.35}$$

Note that N_{feat} does not describe the number of samples for which the features are calculated, as this number is denoted with N_{samples} in Equation 3.1. The main drawback of the L-R selection method, is that there is no empirical way of estimating optimal values for L and R [61, p.6]. In case of this thesis, L and R are chosen to fullfill the condition:

$$R \stackrel{!}{=} L - 1 \tag{3.36}$$

Besides the fulfillment of the condition mentioned in Equation 3.36, L was also chosen to be higher than the half of N_{feat} of the respective feature sets. Depending on N_{feat} L was chosen with:

$$L_{\text{MFCC}} = \begin{cases} 19, & \text{for } N_{\text{FS}} < 18 \\ 18, & \text{for } N_{\text{FS}} = 18 \\ 17, & \text{for } N_{\text{FS}} = 19 \\ N_{\text{FS}} - (2 + 2 \cdot \delta) \text{ with } \delta = \left|19 - N_{\text{FS}}\right|, & \text{else} \end{cases}$$

$$L_{\text{MPS}} = \begin{cases} 6, & \text{for } N_{\text{FS}} < 5 \\ 5, & \text{for } N_{\text{FS}} = 5 \\ 4, & \text{for } N_{\text{FS}} = 6 \\ 3, & \text{for } N_{\text{FS}} = 7 \\ 2, & \text{else} \end{cases} \tag{3.37}$$

$$L_{\text{combo}} = \begin{cases} 24, & \text{for } N_{\text{FS}} < 22 \\ 23, & \text{for } N_{\text{FS}} = 22 \\ 22, & \text{for } N_{\text{FS}} = 23 \\ N_{\text{FS}} - (2 + 2 \cdot \delta) \text{ with } \delta = \left|19 - N_{\text{FS}} - \frac{1}{2}\right|, & \text{else} \end{cases}$$

Given the fact that L is chosen to be larger than R the algorithm explained in section 2.6 starts with an empty feature selection set, which is then filled feature by feature, depending on Fisher Criterion from Equation 2.45. The composition of the reduced feature set, selected by the Plus-L Minus-R algorithm, is listed in the tables inserted in Appendix C.
The figures inserted in subsection 3.5.2 and subsection 3.5.3 contain the performance measures for the single and two stage SVM models, from subsection 3.3.1, for each of the reduced feature sets holding $N_{\text{FS}} = 1, \ldots, (N_{\text{feat}} - 1)$ features picked by the L-R selection. Due to the circumstance that the L-R selection is executed iteratively, reduced feature sets with an increasing number of selected features for each iteration are obtained and the performance measures for each of the reduced feature sets are evaluated. This results in a performance measure progression over an increasing number of selected features N_{FS}, as depicted in all figures of subsection 3.5.2 and subsection 3.5.3. In addition to the L-R selections, an extended selection including the L-R selection and the fundamental frequency estimate $\hat{\boldsymbol{f}}_0$ are also evaluated. Just like the feature sets $\mathscr{X}_{\text{MFCC}}$, $\mathscr{X}_{\text{MPS}}$ and $\mathscr{X}_{\text{combo}}$ the estimated fundamental frequencies are also normalized using `zscore()` [43]. The fundamental frequency estimate is not included in the feature selection process. $\hat{\boldsymbol{f}}_0$ is added afterwards and has to be seen as an

extension of the reduced feature sets provided by the L-R selection. This is also the reason why the curves depicting the performance measures for the L-R selection extended with $\hat{\boldsymbol{f}}_0$, colored in red in Figure 3.14-3.19, start at $N_{\text{FS}} = 2$ selected features and end at $N_{\text{FS}} = N_{\text{feat}}$ features. The unextended L-R selection reaches a maximum number of $N_{\text{FS}} = (N_{\text{feat}} - 1)$ selected features. This is important when analyzing the selected features picked with the L-R selection. For instance, the features selected for $N_{\text{FS}} = 8$ of the unextended feature selection are the same features contained in the $\hat{\boldsymbol{f}}_0$ extended selection of $N_{\text{FS}} = 9$.

As each figure of subsection 3.5.2 and subsection 3.5.3 contains the results of different reduced feature sets whose content is listed in the tables inserted in Appendix C, the subsequent Table 3.12 provides insight into which figure corresponds to which appended table. This gives an overview of the performance analysis carried out in the figures and which features and their corresponding tables are contained in the underlying reduced feature sets.

Table 3.12 *Performance analysis figures and corresponding L-R selection tables.*

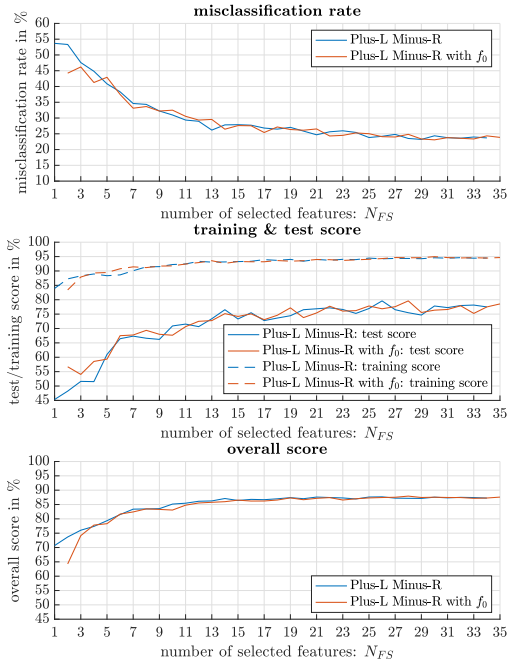
single stage SVM		
figure	**corresponding table**	
Figure 3.14 (a)	Table C.4	*first dataset reduction*
Figure 3.14 (b)	Table C.1	
Figure 3.14 (c)	Table C.7	
Figure 3.15 (a)	Table C.5	*second dataset reduction*
Figure 3.15 (b)	Table C.2	
Figure 3.15 (c)	Table C.8	
Figure 3.16 (a)	Table C.6	*third dataset reduction*
Figure 3.16 (b)	Table C.3	
Figure 3.16 (c)	Table C.9	
two stage SVM		
figure	**corresponding table**	
Figure 3.17 (a)	Table C.10	*first dataset reduction*
Figure 3.17 (b)	Table C.11	
Figure 3.18 (a)	Table C.12	*second dataset reduction*
Figure 3.18 (b)	Table C.13	
Figure 3.19 (a)	Table C.14	*third dataset reduction*
Figure 3.19 (b)	Table C.15	

3.5.2 L-R Selection: Single Stage SVM

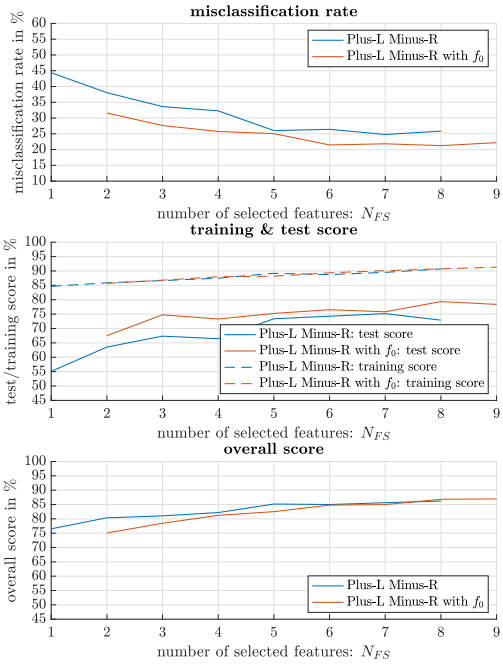
In the next paragraphs the evaluated measures derived using a single stage SVM, for the reduced L-R selection with increasing N_{FS}, with and without $\hat{\boldsymbol{f}}_0$-extension, are illustrated. Figure 3.14 holds the measures for the *first dataset reduction*, followed by the measures obtained for the *second dataset reduction* in Figure 3.15 and the results of the *third dataset reduction*, depicted in Figure 3.16.

First dataset reduction

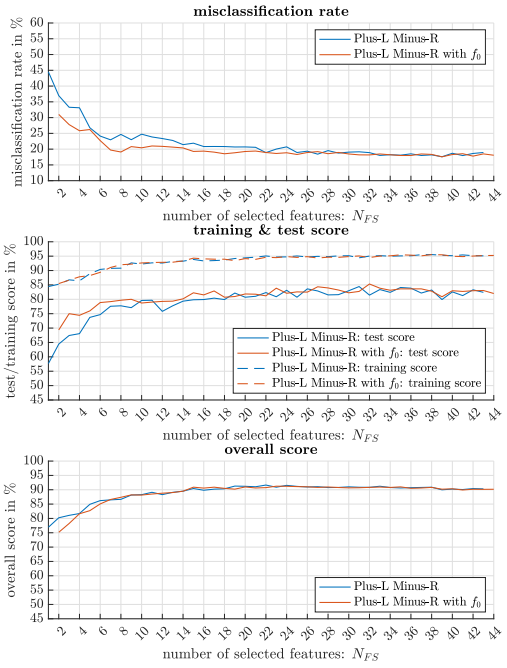
Looking at each performance figure separately, the feature set $\mathscr{X}_{\text{MFCC}}$ whose results are shown in Figure 3.14 (a), shows a *misclassification rate* progression from $55\,\%$ to around $25\,\%$ with increasing N_{FS}. The *training score* starts at already relatively high values and ranges from $85\,\%$ to $95\,\%$ for higher numbers of selected features. On the other hand, the *test score* starts relatively low with $45\,\%$ for $N_{\text{FS}} = 1$ feature and ends between $75\,\%$ to $80\,\%$ at $N_{\text{FS}} = 35$. In general, a saturating behaviour is visible, which indicates that, for a number of selected features $N_{\text{FS}} > 13$, no visible improvement in all scores can be detected, e.g. at $N_{\text{FS}} = 13$ the *overall score* of Figure 3.14 (a) amounts to $85\,\%$ and for $N_{\text{FS}} = 35$, the value only grows with approximately $2.5\,\%$ to a final value of around $87.5\,\%$. The mismatch between the *training* and *test scores* in the saturated state for $N_{\text{FS}} > 13$ settles in at around $15\,\%$- $20\,\%$.The $\hat{\boldsymbol{f}}_0$ extended feature set only shows advantages for very low dimensional feature sets of around $N_{\text{FS}} < 5$, for larger N_{FS} the progressions for the selected features with and without $\hat{\boldsymbol{f}}_0$ exhibit equal courses.



(a) reduced $\mathscr{X}_{\text{MFCC}}$ performance



(b) reduced $\mathscr{X}_{\text{MPS}}$ performance



(c) reduced $\mathscr{X}_{\text{combo}}$ performance

Figure 3.14 *Single stage SVM: L-R feature selection performance for first dataset reduction.*

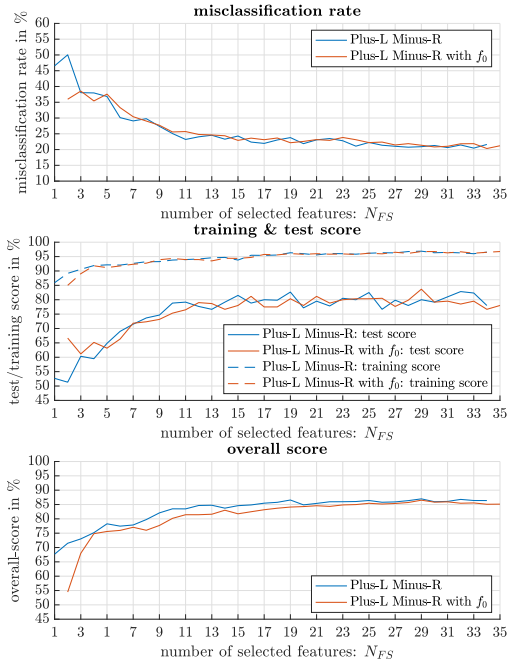
Figure 3.14 (b) depicting the measure progressions of the $\mathscr{X}_{\text{MPS}}$ over an increasing number of selected features contains progressions which already show improved results. The *misclassification rate* for the feature set without extension starts at $45\,\%$, which is $10\,\%$ lower compared to Figure 3.14(a) and settle at $25\,\%$ for $N_{\text{FS}} > 6$ features. The curve of the feature set with the fundamental frequency extension even scratches the $20\,\%$ mark for $N_{\text{FS}} > 6$ features, which, compared to $\mathscr{X}_{\text{MFCC}}$, is an improvement in the *misclassification rate* of about $5\,\%$. In contrast to $\mathscr{X}_{\text{MFCC}}$, the $\hat{\boldsymbol{f}}_0$ extension, in combination with the feature set $\mathscr{X}_{\text{MPS}}$, seems to improve the *misclassification rate* as well as the *test score*, which for the selected features with $\hat{\boldsymbol{f}}_0$, ranges from $67.5\,\%$ to a maximum value of $80\,\%$ (orange curve). However, there is still a mismatch of around $10\,\%$ between the *training* and *test score*, as the maximum value of the *training score* amounts to around $90\,\%$. The *overall score* reaches the same maximum value of $85\,\%$ as in Figure 3.14(a).

When looking at the results obtained for the combined feature set $\mathscr{X}_{\text{combo}}$ in Figure 3.14 (c), a combination of the positive aspects of both feature sets ($\mathscr{X}_{\text{MFCC}}$ and $\mathscr{X}_{\text{MPS}}$) is detectable. The *misclassification rate* even reaches values below $20\,\%$ for $N_{\text{FS}} > 25$ features. Also, the mismatch between *training* and *test score* is smaller than the one achieved with the MFCC feature set $\mathscr{X}_{\text{MFCC}}$ and is given with around $10\,\%$ for $N_{\text{FS}} > 25$, with a *test score* of $85\,\%$. The benefits of the $\hat{\boldsymbol{f}}_0$ extension disappear for $N_{\text{FS}} > 10$ which could indicate that the first features, chosen at the lower N_{FS} are the MPS-based features. This is confirmed, when the corresponding feature selection table Table C.7 is analyzed. The first MFCC-feature that is picked by the L-R selection is $\tilde{\boldsymbol{c}}_1$ in the 6^{th} iteration (for $N_{\text{FS}} = 6$) and the second MFCC that is picked is $\tilde{\boldsymbol{c}}_5$ in the 9^{th} iteration. Even the picked features for $N_{\text{FS}} = 11$ still consist of 8 MPS-based features and only 3 MFCCs. The *overall score* achievable with the combined feature set, saturates at around $90\,\%$.

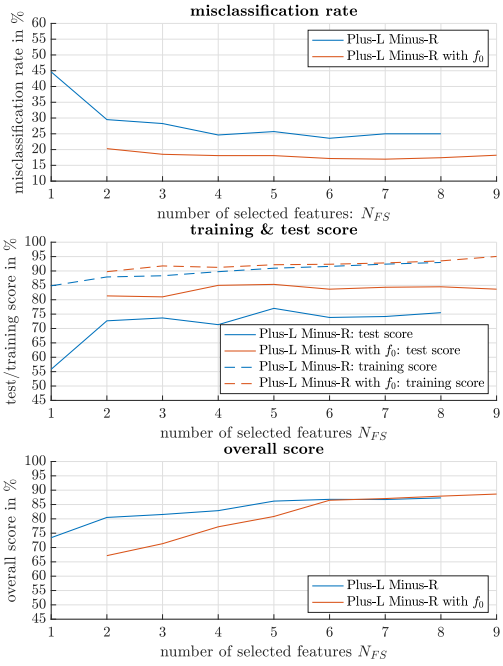
Second Dataset Reduction

The second dataset variation only contains samples that were confidently rated as *breathy*, *normal* or *pressed* and where the *instruction* and *experiment labels* coincide, as presented in section 3.2. Thus, it is anticipated that improved results are noticeable. However, when looking at the results obtained for $\mathscr{X}_{\text{MFCC}}$ in Figure 3.15 (a) in comparison to Figure 3.14 (a), only marginal improvements are present. The *misclassification rate* improves by $5\,\%$, to minimum values of around $20\,\%$ in the first subplot of Figure 3.15 (a). The same improvement is shown for the *test score*, where now a top value of $80\,\%$ is achieved for $N_{\text{FS}} > 13$ features. Also when looking at the mismatch between the *training* and *test scores*, no change is observable. Caused by a higher *training score* (ca. $95\,\%$ for $N_{\text{FS}} > 13$) the mismatch is still given with $15\,\%$. Interestingly, the *overall score* progression of $\mathscr{X}_{\text{MFCC}}$ for the *second dataset reduction* even settles at marginally smaller values than previously. Where in Figure 3.14 (a) an *overall score* of $87.5\,\%$ is displayed, Figure 3.15 (a) only shows an *overall score* of approximately $85\,\%$.

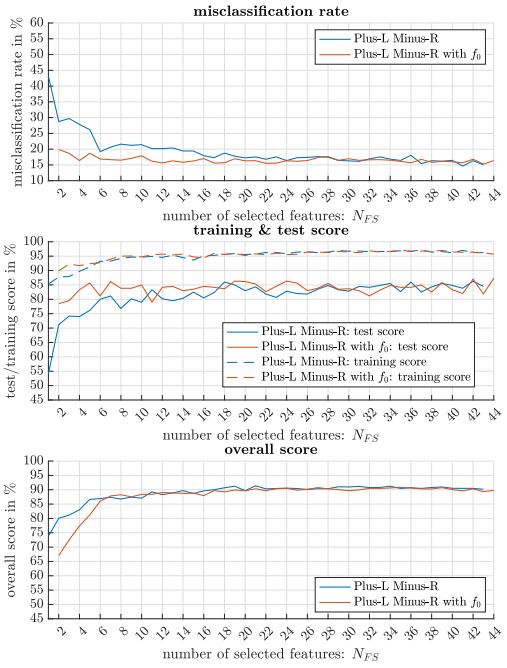
Concerning the MPS based features $\mathscr{X}_{\text{MPS}}$, the anticipated improvements accredited to the *second dataset reduction* are more thoroughly present. Again the feature selection with $\hat{\boldsymbol{f}}_0$ extension outperforms the basic L-R selection, especially when looking at the generalizability measures (*misclassification rate* and *test score*) in Figure 3.15 (b). The *misclassification rate* for the extended feature selection now almost reaches $15\,\%$, but still resides slightly above it, which yields and improvement of $7\,\%$. Concerning the *test score*, the extended feature selection now produces a progression which is located well above $80\,\text{percent}$ with top values of $85\,\%$ for $N_{\text{FS}} > 3$. It is also worth noticing that the distance between the *test score* and the *training score* is diminished. In Figure 3.14 (b) a mismatch of $10\,\%$ is present, whereas Figure 3.15 (b) only exhibits a deviation of *test* and *training score* ranging from $5\,\%$ to $7\,\%$.



(a) reduced $\mathscr{X}_{\text{MFCC}}$ performance



(b) reduced $\mathscr{X}_{\text{MPS}}$ performance



(c) reduced $\mathscr{X}_{\text{combo}}$ performance

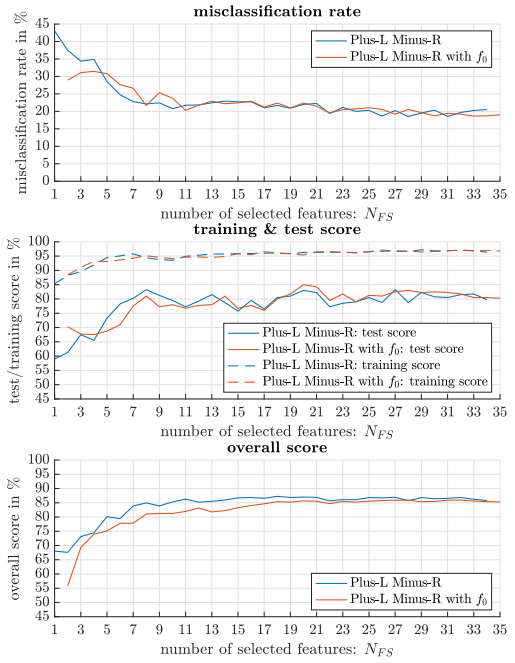
Figure 3.15 *Single stage SVM: L-R feature selection performance for second dataset reduction.*

When looking at $\mathscr{X}_{\text{combo}}$'s *misclassification rate* results with regard to the *second dataset reduction*, the achieved results coincide with the ones obtained from $\mathscr{X}_{\text{MPS}}$ shown in Figure 3.15 (b). A $5\,\%$ improvement of the misclassifcation rate to $15\,\%$ for $N_{\text{FS}} > 6$ features, for the extended feature selection is given. However, the *test* and *training scores* only shows minimal improvement. In Figure 3.15 (c) the *test score* lies marginally closer to $85\,\%$ than in Figure 3.14 (c), but the mismatch towards the *training score* with $10\,\%$ remains. The same circumstance can be observed for the *overall score*, where no change towards the previously analyzed dataset reduction is detectable, meaning a saturated value of $90\,\%$ for $N_{\text{FS}} > 15$ is present.

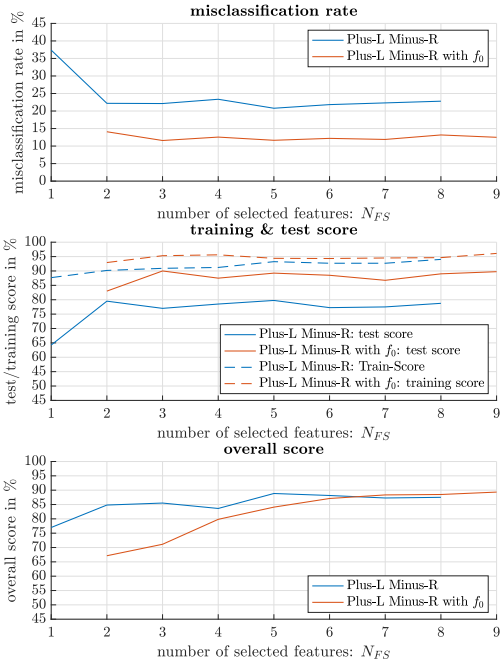
Third Dataset Reduction

The *third dataset reduction*, instills further improvements in the performance of the MPS-based feature set $\mathscr{X}_{\text{MPS}}$ and the combined feature set $\mathscr{X}_{\text{combo}}$. While the performance of $\mathscr{X}_{\text{MFCC}}$ stagnates and does not change in Figure 3.16 (a), compared to Figure 3.15 (a), the MPS-based feature set, used on the *third dataset reduction* brings about the largest advances. A *misclassification rate* of ca. $12\,\%$ for the extended feature selection with $N_{\text{FS}} > 3$ features, as well as a *test score* of ca. $90\,\%$ resulting in, the so far smallest discrepancy towards the *training score* with $5\,\%$ are visible in Figure 3.16 (b). One interesting aspect for $\mathscr{X}_{\text{MPS}}$ comes with the *overall score*. For the extended L-R selection, which starts at a rather low value in comparison to the unextended feature selection. It is only after more than six features are included in the feature selection that the *overall score* curve catches up and resides at $85\,\%$-$87\,\%$. This behaviour is also detectable for the *overall score*, calculated for $\mathscr{X}_{\text{MPS}}$ on the *second dataset reduction*, as illustrated in Figure 3.15 (b). There the curve of the *overall score* over the number of selected features nearly shows an identical progression.

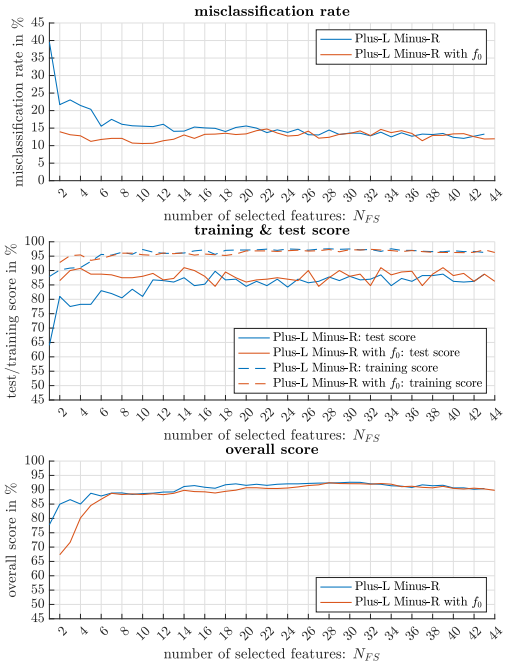
Figure 3.15 (c), which depicts the performance measures of $\mathscr{X}_{\text{combo}}$ used for the *third dataset reduction*, yields to a *misclassification rate* of ca. $10\,\%$ for $5 \leq N_{\text{FS}} \leq 12$. For a number of selected features higher than 12, the *misclassification rate* gradually increases and again saturates at $15\,\%$ for $N_{\text{FS}} > 18$. This is also observable for the *test score* in Figure 3.16 (c), where the $\hat{\boldsymbol{f}}_0$ extended feature selection already starts with a very prominent value of $90\,\%$ for only $N_{\text{FS}} = 3$ features, which is upheld until $N_{\text{FS}} = 10$. Here the deviation between *training* and *test score* is given with approx. $5\,\%$. However, for $N_{\text{FS}} > 10$ features, the *test score* first starts to fluctuate and then drops to values of around $85\,\%$. When looking at Table C.9 which lists the picked features from $\mathscr{X}_{\text{combo}}$ for $N_{\text{FS}} > 10$, it is shown that the algorithm again prefers the MPS-based features contained in $\mathscr{X}_{\text{combo}}$. The *overall score* of Figure 3.16 (c) also exhibits little changes, the $90\,\%$ mark is slightly surpassed for $N_{\text{FS}} > 13$ but keeping in mind that there are slight statistical deviations behind every measure, it can be argued that in comparison to Figure 3.15 (c) the *overall score* does not show significant improvement.



(a) reduced $\mathscr{X}_{\text{MFCC}}$ performance



(b) reduced $\mathscr{X}_{\text{MPS}}$ performance



(c) reduced $\mathscr{X}_{\text{combo}}$ performance

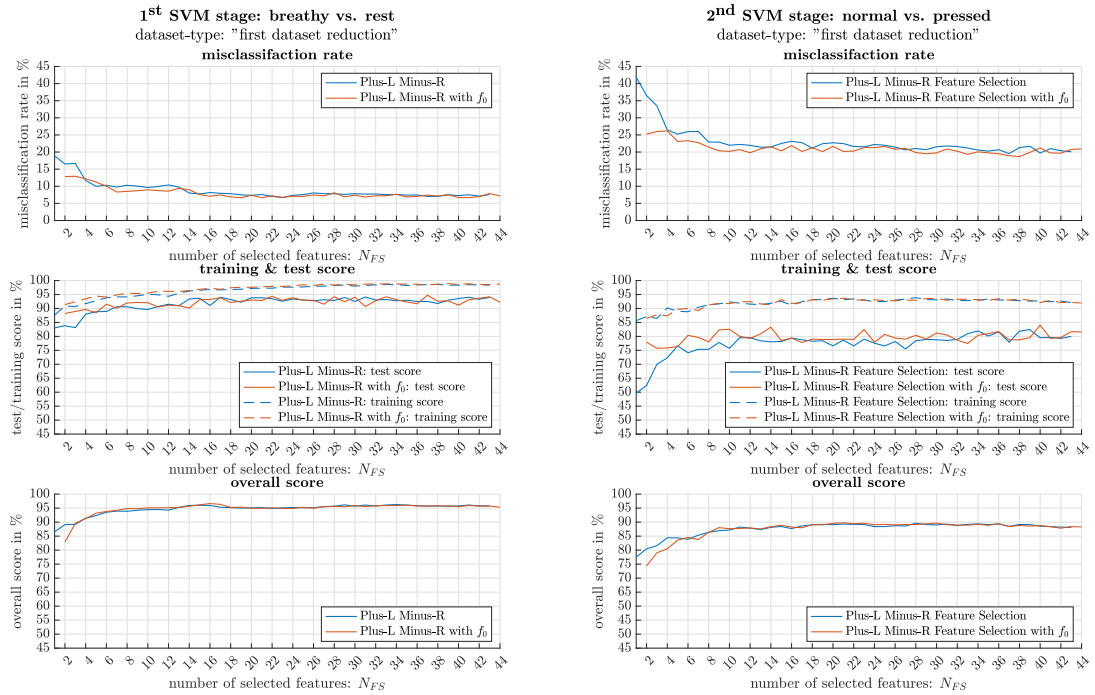
Figure 3.16 *Single stage SVM: L-R feature selection performance for third dataset reduction.*

3.5.3 L-R Selection: Two Stage SVM

The two stage SVM model enables the assessment of each classification task separately. In the first stage *breathy* voice quality is distinguished. The classification task of the first stage is denoted "*breathy* vs. *rest*" and the second one differentiates "*normal* vs. *pressed* " voice qualities. Again all three dataset reductions presented in section 3.2 are analyzed. Different to the one stage SVM, whose results for the three dataset reductions are presented in subsection 3.5.2, for the two stage SVM only the combined feature set $\mathscr{X}_{\text{combo}}$ is analyzed, supposing, that the feature selection algorithm picks the most suitable features for each classification stage separately. Again the picked features for each L-R selection iteration are listed in the tables of Appendix C, which table belongs to which feature selection figure is listed in Table 3.12.

First dataset reduction

It is now clearly shown, when comparing subfigures (a) and (b) of Figure 3.17 that the more problematic classifcation is given at the second SVM stage where *normal* and *pressed* phonation are distinguished. The misclassification at the first SVM stage settles at $7\,\%$ for $N_{\text{FS}} > 13$ features, whereas the second SVM stage performs worse with a $20\,\%$ misclassification for the same number of selected features. Also the deviation between *training* and *test score* for the *breathy* classification is minimized to a little bit less than $5\,\%$, with a *test score* that reaches ca. $93\,\%$ for $N_{\text{FS}} > 13$ and a *training score* that surpasses $95\,\%$ at $N_{\text{FS}} = 10$ and even reaches top values of $98\,\%$ - $99\,\%$ for $N_{\text{FS}} > 26$. In contrast to that, the second SVM stage exhibits a mismatch of around $15\,\%$, with a *test score* that saturates at around $80\,\%$. The better performing *breathy* vs. *rest* classification is also reflected, when the *overall score* of both stages are compared. On the one hand, the *overall score* of the first stage, shown in Figure 3.17, offers values of $95\,\%$ and on the other hand, the *overall score* presented in Figure 3.17 (b) of the second barely reaches $90\,\%$.

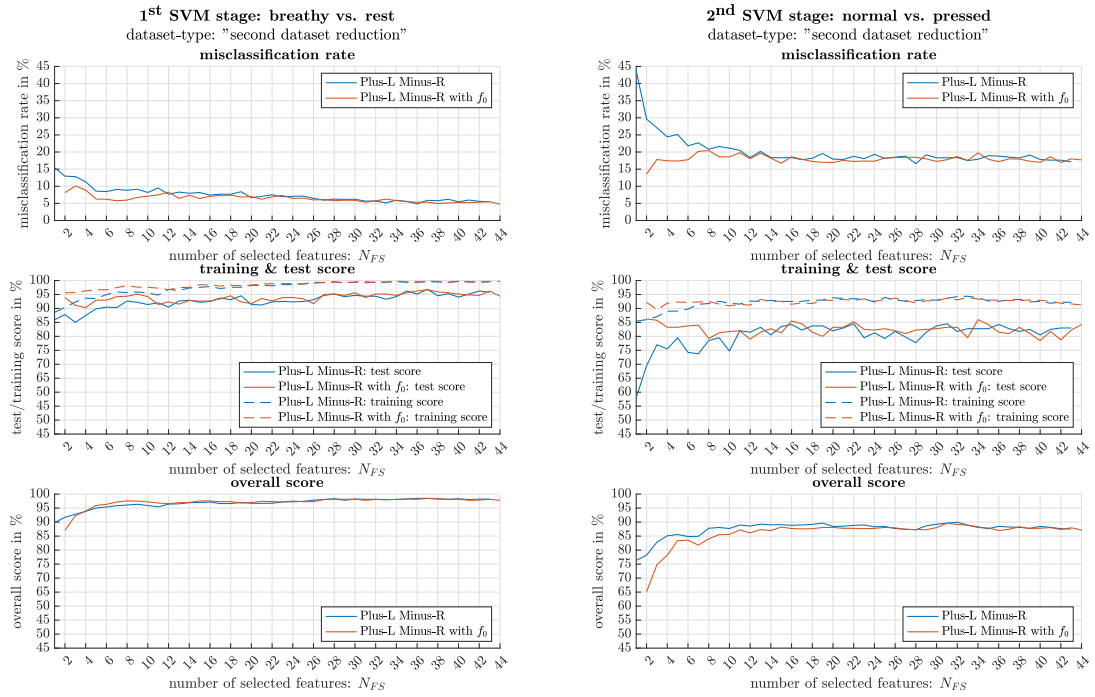


(a) first SVM stage: *breathy* vs. *rest* (b) second SVM stage: *normal* vs. *pressed*

Figure 3.17 *$\mathscr{X}_{combo}$: L-R selection performance results; first dataset reduction.*

Second Dataset Reduction

As previously remarked during the performance discussion of the single stage SVM, the *second dataset reduction* is also used in the two stage SVM and is also accompanied by performance improvements. It can be argued that the $5\,\%$ improvement, which was present in the *misclassification rate* for the single stage SVM model discussed in the previous paragraph and visualized in Figure 3.15, is split equally onto both SVM stages during the two stage classification process, because the *misclassification rate's* improvement visible for both SVM stages in Figure 3.18 amounts to around $2.5\,\%$. Also higher *test* and *training scores* can be observed. The *training score* of the "*breathy vs. rest*" classification executed in the first SVM stage nearly reaches $100\,\%$ for $N_{\text{FS}} > 30$ features and a *test score* of above $95\,\%$ also keeps the *test/training score* mismatch in the vicinity of a marginally less than $5\,\%$. This coincides with performance present on the *first dataset reduction* reduction displayed in Figure 3.17. Finally, also the *overall score* reaches new heights for the first SVM stage depicted in Figure 3.15 (a), where the *overall score* surpasses $95\,\%$ already at $N_{\text{FS}} = 5$ features. The *overall score* of the second stage saturates at $87\,\%$ to $89\,\%$.



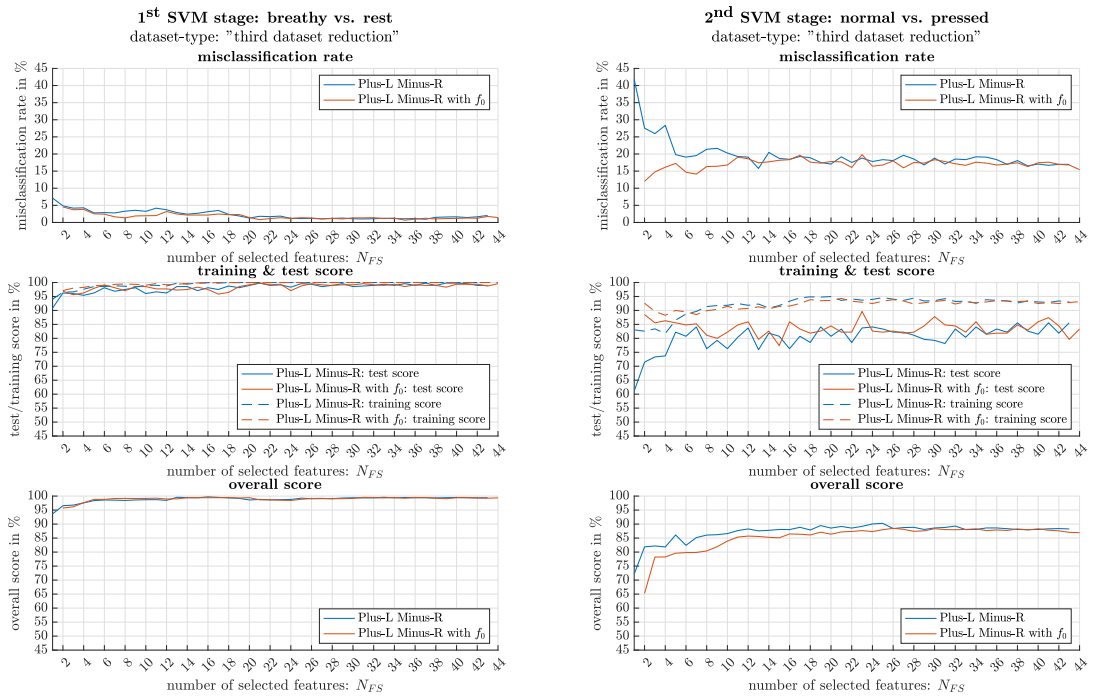
(a) first SVM stage: *breathy* vs. *rest* (b) second SVM stage: *normal* vs. *pressed*

Figure 3.18 *$\mathscr{X}_{combo}$: L-R selection performance results; second dataset reduction.*

Third Dataset Reduction

The *third dataset reduction* takes the improvement even further, especially for the first SVM stage, for which the *misclassification rate* progression already falls below a value of $5\,\%$ for $N_{\text{FS}} > 3$. The lowest values are then reached for $N_{\text{FS}} > 20$ features and reside at ca. $1\,\%$. In accordance with the *misclassification rate* progression the *test score* also exhibits very promising results throughout all the analyzed number of selected features. The deviation towards the *training score* practically vanishes and the *training* and *test scores* reach values of $99\,\%$ - $100\,\%$ for $N_{\text{FS}} > 25$. The same progression is detectable for the *overall score* shown in Figure 3.19 (a). When analyzing the performance measures evaluated for the 2nd SVM stage presented in Figure 3.19 (b), it is perceivable that the progressions are way more fluctuating than the ones of Figure 3.18 (b) and Figure 3.17 (b). This

can be led back to the reduced number of samples contained in the underlying *third dataset reduction* in combination with the random undersampling. The statistical variation brought in by the random undersampling procedure is amplified if fewer samples are contained in the dataset. As visible in Figure 3.4 the underlying data processed in the second stage of the two stage SVM model, after the *third dataset reduction* only contains an overall number of 136 samples, which results in 68 samples per class. Nevertheless, the fluctuations still allow a determination of the performance measure values. A *misclassification rate* of 12 % to approx. 17 % is visible for the $\hat{\boldsymbol{f}}_0$ extended feature selection in Figure 3.19 (b). Interestingly enough the lower values of 12 %- 15 % are achieved for a lower number of selected features e.g. $N_{\mathrm{FS}} < 10$, for larger N_{FS} the *misclassification rate* rises. The same behaviour is observable for the *test score* of the extended feature selection. Higher values of ca. 85 %, are obtained for lower number of selected features ($N_{\mathrm{FS}} < 8$). However, it has to be noted that the observed improvements towards lower N_{FS} lie in the vicinity of the statistical variation brought in by RUS, therefore, it cannot be stated that the classification of the *third dataset reduction* exhibits better performance for a lower number of selected features. For higher numbers of selected features the score then fluctuates between 80 % to 85 %, the *test score* for the unextended feature selection catches up to the results produced by the feature selection with $\hat{\boldsymbol{f}}_0$ extension for $N_{\mathrm{FS}} > 10$. The *overall score* also reaches values varying between 85 % and 90 %, the extended and unextended feature selections exhibit an identical course for $N_{\mathrm{FS}} > 10$. For lower numbers of selected features ($N_{\mathrm{FS}} < 10$) the previously described circumstance that the extended feature selection causes a lower *overall score* than the unextended feature selection, is also noticeable.



(a) first SVM stage: *breathy* vs. *rest* (b) second SVM stage: *normal* vs. *pressed*

Figure 3.19 *$\mathscr{X}_{combo}$: L-R selection performance results; third dataset reduction.*

In conclusion, the two stage SVM performance analysis contains repeating aspects throughout Figure 3.17 to 3.19, the most prominent being the deviating performance of the $\hat{\boldsymbol{f}}_0$-extended feature selection in the second SVM stage. Especially within lower numbers of selected features, e.g. $N_{\mathrm{FS}} < 10$ features, the extended feature selection shows significantly better *misclassification rates* and *test*

scores than the L-R selection without $\hat{\boldsymbol{f}}_0$. As already discussed for Figure 3.14 (c), this can be explained by looking at the selected features listed in the tables of section C.2 for each respective performance measure evaluation. There are differences with respect to the order, but for all cases the MPS-based features $\mathscr{X}_{\text{MPS}}$ are preferred by the Plus-L Minus-R selection, for both classification stages. For the first stage the $\hat{\boldsymbol{S}}^{\text{pk},2}_{\Sigma,\,\text{spec}}$ is picked first, whereas for the second stage the LR-selcetion is started with $\bar{\boldsymbol{\Delta}}_{\text{spec},1}$. The behaviour observable for $N_{\text{FS}} < 10$ of all two stage SVM performance measure evaluations for each dataset reduction (Figure 3.17 to 3.19) coincides with the observations made for MPS-based features used in the single stage SVM to calculate the performance measure progressions shown in Figure 3.14 (b) to 3.16 (b). Although, the first MFCC features are picked for $N_{\text{FS}} > 5$ or $N_{\text{FS}} > 8$, depending on the dataset reduction, it is shown in section C.2 that a feature selection consisting of 10 features is still predominantly composed of MPS-based features. As the MPS-based features contained in $\mathscr{X}_{\text{combo}}$ are picked first, and they show an improved performance if $\hat{\boldsymbol{f}}_0$ is added as an additional feature, a fundamental frequency dependency of the MPS-based features $\mathscr{X}_{\text{MPS}}$ is indicated. This fundamental frequency dependence vanishes, if more MFCC features come into play. For $N_{\text{FS}} > 20$ features the $\hat{\boldsymbol{f}}_0$ dependence vanishes, as the *misclassification rates*, as well as the *test*, *training* and *overall scores* exhibit identical performance measure progressions as for the unextended L-R selection.

3.6 Classification Performance on Full Dataset

In subsection 3.5.3 three dataset variations have been analyzed, but so far the full dataset has been left out. The insights on the performance of the different feature sets $\mathscr{X}_{\text{MFCC}}$, $\mathscr{X}_{\text{MPS}}$ and $\mathscr{X}_{\text{combo}}$ are now important, in order to determine one feature set that is used in the single and two stage SVMs on the full unbalanced dataset with the *instruction* and *experiment labels*. The analysis with regard to switching labels yields an assessment on which labels yield the better ML-based classification results. Additionally, it enables a comparison of the ranking behaviour of the listeners who took part in the listening experiment that was conducted to evaluate the dataset, as mentioned in section 3.1. As the combined dataset $\mathscr{X}_{\text{combo}}$ proved itself to be the best performing dataset for the *first dataset reduction*, which is the dataset variation closest to the full dataset, it is used in this section to classify the full dataset. Regarding the number of samples it helps to look at the highest *test score* and lowest *misclassification rate* which are found for the largest number of selected features ($N_{\text{FS}} = 44$ features for the extended feature selection). Also the deviation between *training* and *test score*, which is the indicator for overfitting behaviour, does not grow for an increasing dimensionality within the selected feature set.
Additionally, Figure 3.20 provides an overview on the separability of different feature set configurations with regards to the *first dataset reduction*. The full MPS-based feature set $\mathscr{X}_{\text{MPS}}$, the combined feature set $\mathscr{X}_{\text{combo}}$ comprising 22 features, which were picked by the L-R selection listed in Table C.7, and the full combined feature set $\mathscr{X}_{\text{combo}}$ with all $N_{\text{FS}} = 44$ features were transformed into a 2D-space using a linear discriminant analysis (LDA) and the eucledian distances between the 2D cluster means are calculated and summarized in Table 3.13. The linear discriminant analysis builds on the *within* and *between scatter matrices*, mentioned in relation with the Plus-L Minus-R algorithm in section 2.6. Using the `Matlab` command `eig()` [29]. The `eig()` command outputs the eigenvectors $\boldsymbol{\phi}$ of the matrix product *between* the inverse *within class scatter matrix* $\boldsymbol{S}_W^{-1}$ and the between class scatter matrix $\boldsymbol{S}_B$. As a precautionary measure, a regularization term of $\epsilon = 10^{-6}$ is added onto the main diagonal of the *within scatter matrix*, in order to ensure stability for the matrix inversion of $\boldsymbol{S}_W$. The executed `Matlab` command used to retrieve the projection vectors $\boldsymbol{\phi}$ is given with Equation 3.38.

$$\boldsymbol{\phi} = \texttt{eig}\left((\boldsymbol{S}_W + \boldsymbol{I} \cdot \epsilon)^{-1}\boldsymbol{S}_B\right) \tag{3.38}$$

The eigenvectors ϕ_1 and ϕ_2 contained in ϕ, corresponding to the largest eigenvalues[2], are used to project the feature-sets into a 2D-feature space, by calculating the inner product between the eigenvectors and a feature set denoted as $\mathscr{X}$, as suggested in Equation 3.39.

$$\overbrace{\boldsymbol{y}_{\text{LDA}}^{(1)}}^{[N_{\text{samples}}\times 1]} = \overbrace{\mathscr{X}}^{[N_{\text{samples}}\times N_{\text{feat}}]} \cdot \overbrace{\boldsymbol{\phi}_1}^{[N_{\text{feat}}\times 1]} \qquad \boldsymbol{y}_{\text{LDA}}^{(2)} = \mathscr{X} \cdot \boldsymbol{\phi}_2 \tag{3.39}$$

The different feature configurations, projected into the two dimensional feature space are all extended with $\hat{\boldsymbol{f}}_0$. Mathematically the extension is indicated by using the symbols: $\mathscr{X}_{\text{MPS\&}\hat{f}_0}$ and $\mathscr{X}_{\text{combo\&}\hat{f}_0}$. The cluster means of the projected feature sets are depicted in Figure 3.20 and are denoted with $\mu_{\text{LDA}}^{(b)}$ for *breathy* phonation and $\mu_{\text{LDA}}^{(n)}$ and $\mu_{\text{LDA}}^{(p)}$ for *normal* and *pressed* voice quality respectively. Additionally, the σ-confidence ellipsoids enclosing 68 % of the data samples are also plotted.

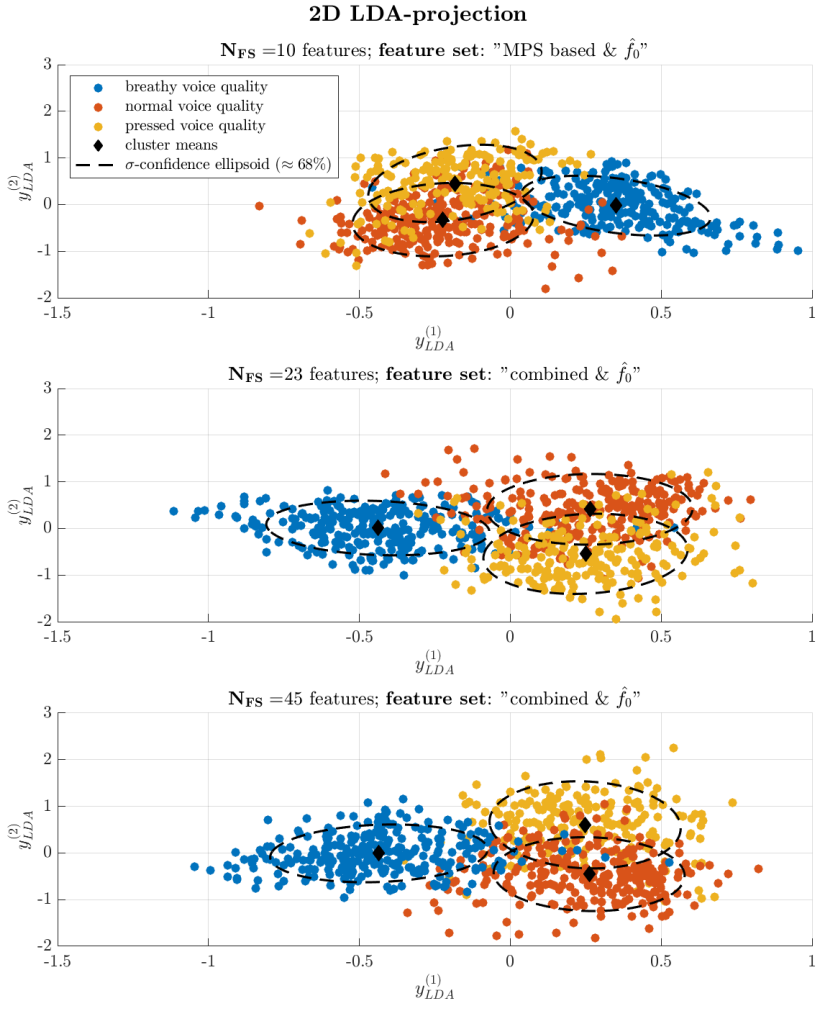


Figure 3.20 *2D LDA projection of the full and reduced combined feature set as well as the full MPS based feature set applied onto the first dataset reduction.*

[2]Note that the `Matlab` command `eig()` does not output the eigenvectors in descending order, with regard to the eigenvalue size. An additional sorting step using `Matlab's sort()` [40] is necessary.

The LDA projection also shows the problematic separability between the classes *normal* and *pressed*, which has already been disclosed by the two stage SVM analyzed in subsection 3.5.3. In Figure 3.20 it is shown that the projected clusters of *normal* and *pressed* phonation are way more intersected than the *breathy* cluster. The Euclidean distances of all cluster means of Table 3.13 are sorted in the same way as the subplots of Figure 3.20, e.g. the first feature configuration corresponds to the feature set whose LDA projection is depicted in the first subplot of Figure 3.20.

Table 3.13 *Euclidean distance of 2D LDA projected cluster means.*

Nr.	**feature configuration:**	**euclidean distance**		
		$\left\|\boldsymbol{\mu}_{\text{lda}}^{(b)} - \boldsymbol{\mu}_{\text{lda}}^{(n)}\right\|$	$\left\|\boldsymbol{\mu}_{\text{lda}}^{(b)} - \boldsymbol{\mu}_{\text{lda}}^{(p)}\right\|$	$\left\|\boldsymbol{\mu}_{\text{lda}}^{(n)} - \boldsymbol{\mu}_{\text{lda}}^{(p)}\right\|$
1.)	$\mathscr{X}_{\text{MPS\&}\hat{f}_0}$; $N_{\text{FS}} = 10$ features	0.6470	0.7153	0.7782
2.)	$\mathscr{X}_{\text{combo\&}\hat{f}_0}$; $N_{\text{FS}} = 23$ features	0.8089	0.8829	0.9540
3.)	$\mathscr{X}_{\text{combo\&}\hat{f}_0}$; $N_{\text{FS}} = 45$ features	0.8268	0.9173	1.0570

In order to maintain the best separability aim of the LDA is to keep a maximum distance between the clusters, i.e. maximizing the between class covariance, whilst keeping the within class covariance to a minimum [5, p.187-189]. This separability can be compared for the three feature set configurations, using the distances listed in Table 3.13 and the confidence ellipsoids in Figure 3.20. It is visible that the worst separability is given for the first feature set configuration $\mathscr{X}_{\text{MPS\&}\hat{f}_0}$ and the second and third configuration show comparable separability, in terms of the position and intersection of the confidence ellipsoids depicted in Figure 3.20. However, the largest distances, between the clusters, are found for the third feature configuration containing the combined feature set $\mathscr{X}_{\text{combo}}$ with all $N_{\text{FS}} = 44$ features, including $\hat{\boldsymbol{f}}_{\boldsymbol{0}}$. This reinforces the assessment completed in subsection 3.5.2, that the combined feature set $\mathscr{X}_{\text{combo}}$ with $\hat{\boldsymbol{f}}_{\boldsymbol{0}}$-extension yields the best performance for the *first dataset reduction* and overfitting due to dimensionality (*curse of dimensionality*) does not occur in a problematic extent.

Therefore, the following subsections provide the performance measures from subsection 3.3.1 as well as confusion matrices, created using `Matlab's confusionchart()` [27], on the classification task fulfilled for the full dataset with both *experiment* and *instruction lables* using the combined feature set $\mathscr{X}_{\text{combo}}$ with $\hat{\boldsymbol{f}}_{\boldsymbol{0}}$ using the single and two stage SVM model.

3.6.1 Single Stage SVM Performance on Full Dataset

According to the implementation presented in section 3.3 the single stage SVM is used with 500 iterations to determine the kernel-scale and 100 classification iterations to provide statistical context for the classification results on the full dataset with varying labels. The evaluated mean (μ) and standard deviation (σ) of the performance measures calculated using the 100 classification iterations are listed as tables in the following paragraphs. Additionally, the trained SVM model achieving the highest *test score* amongst the 100 classification iterations is used to process the full dataset again, delivering estimated labels for the whole dataset and enabling the creation of confusion matrices and thus, presenting a compact overview on the class assignment of each sample. Moreover this allows a comparison of the ML-classification with the rating behaviour observable from the listening experiment, which is summarized in Figure 3.2. Firstly, the results using the full dataset in combination with the *instruction labels* are mentioned and then the performance of the single stage SVM on the full dataset with the *experiment labels* is discussed.

Performance with *instruction labels*

When the full dataset with *instruction labels* is used, the performance of the single stage SVM, with regards to the generalization measures (*misclassification rate* and *test score*), decreases with approximately $5\,\%$ in comparison to the single stage SVM performance on the *first dataset reduction* summarized in Figure 3.14 (c). The mismatch between *test* and *training score* with ca. $15\,\%$ remains equal in comparison to Figure 3.14 (c) indicating the same overfitting behaviour as in the *first dataset reduction*. All determined performance measures for the full dataset with *instruction labels*, using the complete combined feature $\mathscr{X}_{\text{combo}}$ set with $\hat{\boldsymbol{f}}_0$-extension are summarized in Table 3.14.

Table 3.14 *Estimated performance measures using 100 classification iterations for the single stage SVM on a full dataset with instruction labels.*

performance measure:	$\mu \pm \sigma$
training score:	$92.9\,\% \pm 0.51\,\%$
test score:	$76.95\,\% \pm 2.61\,\%$
misclassification rate:	$23.82\,\% \pm 1.12\,\%$
overall score:	$89.71\,\% \pm 0.56\,\%$

The performance measures of Table 3.14 and the confusion matrix depicted in Figure 3.21 always have to be viewed in relation. Although, the confusion matrix shown in Figure 3.21 seems to show very promising results, especially when looking at the row summary of the confusion matrix, which holds the percentage of correctly classified samples per class, it has to be kept in mind that behind the displayed classification results there is a SVM model which has already seen $80\,\%$ of all available data during the fitting process. The result is a train score of around $92.9\,\%$ (see Table 3.14). In terms of generalization, one has to expect $23.82\,\%$ misclassified samples.

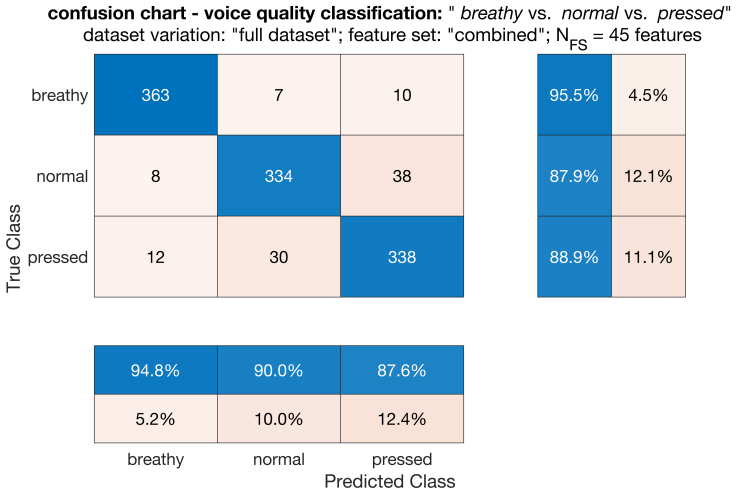


Figure 3.21 ***Confusion matrix****: absolute sample numbers distributed over 3 voice quality classes,*
SVM model configuration*: single stage SVM,*
Feature set*:* $\mathscr{X}_{combo\&\hat{f}_0}$, $N_{FS} = 45$,
Dataset*: full dataset with instruction labels.*

Most mix-ups within the class assignments occur for *normal* and *pressed* voice quality. 30 *pressed* samples are classified as *normal* and 38 samples with *normal* phonation are classified as *pressed*. A normalized version of the confusion matrix is shown in Figure B.6, where the absolute sample num-

bers are normalized with the underlying total number of samples (1140 samples), resulting in relative values in percent. The row and column summaries depicted within the confusion charts contain information on how many correctly classified samples are within the predicted classes (column summary) and how many samples of the true classes are correctly classified (row summary), thus, a normalization towards the amount of samples contained in a row or column holds limited new information. The normalization with the underlying total number of samples on the other hand, provides additional information, e.g. the total percentage of correctly classified samples, calculatable by summing the main elements of a confusion chart with percentage values.

Performance with *experiment labels*

If the *experiment labels* are used, the classification performance decreases. The mean *misclassification rate*, calculated from 100 classification iterations, drops to $25.59\,\%$ and the mean *test score* to $75.41\,\%$. The most prominent decrease, however, can be found with the *overall score*. When comparing the mean *overall score* of Table 3.14 and Table 3.15 a drop of approximately $9\,\%$ is visible.

Table 3.15 *Estimated performance measures using 100 classification iterations for the single stage SVM using $\mathscr{X}_{combo}$ on a full dataset with experiment labels.*

performance measure:	$\mu \pm \sigma$
training score:	$93.39\,\% \pm 0.56\,\%$
test score:	$75.41\,\% \pm 3.09\,\%$
misclassification rate:	$25.59\,\% \pm 1.28\,\%$
overall score:	$80.32\,\% \pm 0.73\,\%$

The declining *overall score* is also noticeable in Figure 3.22, where even more misclassification between *normal* and *pressed* phonation occurs. Also the row summary of Figure 3.22 displays that only $70.6\,\%$ of *normal* samples are correctly identified, which corresponds to 111 *normal* samples that are wrongly classified as *pressed*. The best results are achieved for *breathy* voice quality, which underpins the observation made in the feature selection analysis of the two stage SVM in subsection 3.5.3.

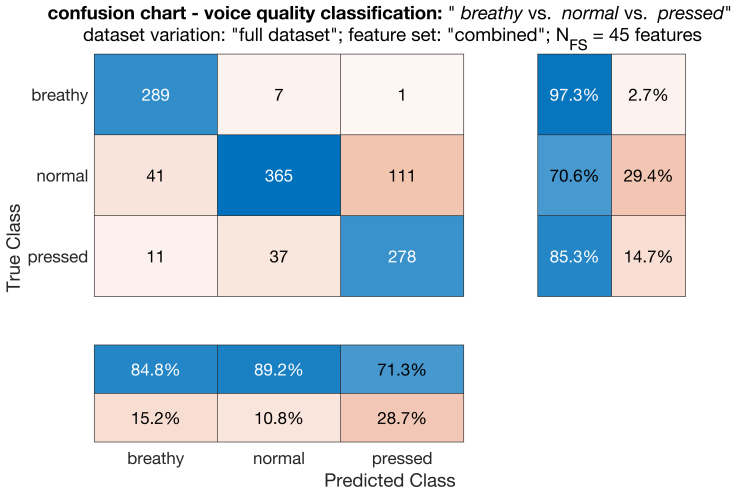


Figure 3.22 ***Confusion matrix****: absolute sample numbers distributed over 3 voice quality classes,*
SVM model configuration*: single stage SVM,*
Feature set*: $\mathscr{X}_{combo\&\hat{f}_0}$; $N_{FS} = 45$,*
Dataset*: full dataset with experiment labels.*

3.6.2 Two Stage SVM Performance on Full Dataset

For the two stage SVM analysis mentioned in section 3.3 used for the full dataset with *instruction* and *experiment labels*, the same strategy as for the single stage SVM is followed. For each classification stage 500 iterations are executed to determine the kernel-scale of the corresponding SVM classifier. Then in each stage 100 classification iterations are used to determine the mean (μ) and standard deviation (σ) of the performance measures for both SVM stages. The first stage is responsible for the distinction of *breathy* and the second SVM stage for the distinction of *normal* vs. *breathy*. For both stages, tables containing the performance measures and confusion matrices are created[3]. In order to provide a figure which can be compared to the previous confusion matrices in Figure 3.21 and Figure 3.22, the full dataset is processed through both stages, classifying every sample, whereby the second stage is only reached by those samples that were assigned to the *rest* class within the first SVM stage.

Performance with *instruction labels*

Table 3.16 depicts the performance measures achieved in both voice quality stages for the full dataset with *instruction labels*. A very clear trend is detectable. With a *misclassification rate* of $9.1\,\%$ and a *test score* of $91.72\,\%$ the *breathy* vs. *rest* classification performs very well. The problem for the *normal* vs. *pressed* classification remains and with a *misclassification rate* of $24.86\,\%$, one has to anticpiate that a quarter of samples are misclassified, if the model is presented with new data, hence a *test score* of $76.83\,\%$.

Table 3.16 *Estimated performance measures using 100 classification iterations for the two stage SVM on a full dataset with instruction labels.*

1st SVM stage: "*breathy* vs. *rest*"		**2nd SVM stage:** "*normal* vs. *pressed*"	
performance measure:	$\mu \pm \sigma$	**performance measure:**	$\mu \pm \sigma$
training score:	$97.8\,\% \pm 0.39\,\%$	*training score*:	$90.86\,\% \pm 0.74\,\%$
test score:	$91.72\,\% \pm 2.07\,\%$	*test score*:	$76.83\,\% \pm 3.3\,\%$
misclassification rate:	$9.1\,\% \pm 0.85\,\%$	*misclassification rate*:	$24.86\,\% \pm 1.4\,\%$
overall score:	$90.39\,\% \pm 0.57\,\%$	*overall score*:	$88.06\,\% \pm 0.87\,\%$

The confusion matrix of the two stage SVM depicted in Figure 3.23 is fairly similar to the one of the single stage SVM depicted in Figure 3.21, especially with regards to the classification of *breathy* voice quality. Concerning the distinguishability of *normal* and *pressed* phonation the confusion chart in Figure 3.23 exhibits slight drawbacks in comparison to Figure 3.21. Another aspect that stands out in comparison to the performance of the single stage SVM on the full dataset with *instruction labels* is that in Figure 3.23, more mix-ups of *normal* and *pressed* voice quality with *breathy* occur than for the single stage SVM results. For instance, in Figure 3.21 only 10 *breathy* samples are mistaken as *pressed* and in Figure 3.23, 51 *pressed* samples are mistaken as *breathy*.

[3]The confusion matrices of this section were also calculated using relative values, where the numbers within the charts are normalized to the underlying total number of samples, they are added in the appendix in Figure B.6 and Figure B.7

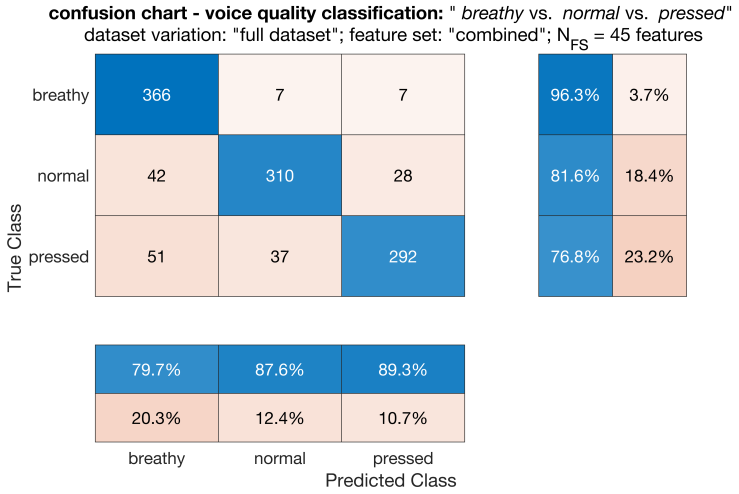


Figure 3.23 ***Confusion matrix****: absolute sample numbers distributed over 3 voice quality classes,*
SVM model configuration*: two stage SVM,*
Feature set*:* $\mathscr{X}_{combo\&\hat{f}_0}$*;* $N_{FS} = 45$*,*
Dataset*: full dataset with instruction labels.*

Performance with *experiment labels*

When *experiment labels*, in combination with the two stage SVM are used the same behaviour as for the single stage SVM is given. The performance measures also exhibit a drop, especially when looking at the *overall score*. The *overall score* in the first stage drops to $86.7\,\%$, which is approx. $4\,\%$ worse than for the usage of the *instruction labels* (see Table 3.16) and the second stage's *overall score* declines by $8\,\%$ to $80.6\,\%$.

Table 3.17 *Estimated performance measures using 100 classification iterations for the two stage SVM on a full dataset with experiment labels.*

1st SVM stage: "*breathy* vs. *rest*"		**2nd SVM stage:** "*normal* vs. *pressed*"	
training score:	$97.22\,\% \pm 0.42\,\%$	*training score*:	$92.12\,\% \pm 0.72\,\%$
test score:	$90.96\,\% \pm 2.63\,\%$	*test score*:	$74.46\,\% \pm 3.64\,\%$
misclassification rate:	$9.04\,\% \pm 1.07\,\%$	*misclassification rate*:	$26.21\,\% \pm 1.47\,\%$
overall score:	$86.7\,\% \pm 0.54\,\%$	*overall score*:	$80.06\,\% \pm 0.85\,\%$

What stands out when looking at the confusion matrix depicted in Figure 3.24 is that even more *normal* samples are misclassified as *breathy* samples. The full dataset with *experiment labels* becomes imbalanced and its biggest class is the *normal* class, which consists of 517 samples as shown in Table 3.5. The biggest class of the dataset also exhibits the most misclassifications when looking at the performance in Figure 3.24. 110 *normal* samples are mistakenly classified as *breathy* and 87 samples are classified as *pressed*, yielding only a $61.9\,\%$ accuracy for *normal* voice quality samples. The best results are again achieved for *breathy* voice quality. There $96.3\,\%$ of all *breathy* samples are correctly classified.

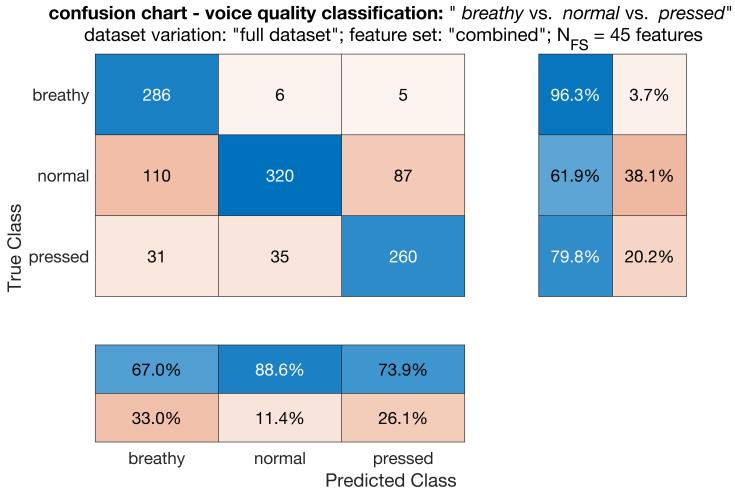


Figure 3.24 ***Confusion matrix****: absolute sample numbers distributed over 3 voice quality classes,*
SVM model configuration*: two stage SVM,*
Feature set*:* $\mathscr{X}_{combo\&\hat{f}_0}$*;* $N_{FS} = 45$*,*
Dataset*: full dataset with experiment labels.*

Additionally, confusion matrices with relative values, created by using a hold-out set, with the intention of showing the classification behaviour towards data that have not yet been presented to the model, are also appended in section B.3. Regarding the single stage SVM, the hold-out set is given by the test set, which holds $20\,\%$ of the data as illustrated in Figure 3.5. Concerning the two stage SVM, the hold-out set is stored before the datasets for the two stages are prepared. This means that $20\,\%$ of the data (randomly chosen) are used as the hold-out set and the other $80\,\%$ are then further processed and split into the two sub-datasets according to the procedure mentioned in subsection 3.2.2. It is important to note that this procedure was soley executed for the creation of the hold-out set confusion matrices appended in Figure B.7, all other analysis carried out in this thesis is done with the method presented in subsection 3.2.2.

The hold-out set's confusion matrices and the confusion charts of this section as well as the comparison with regards to the ranking behaviour of the listening experiment are carried out in the next section. In addition, the observations made in course of the present analysis are summarized.

3.6.3 Performance Summary of Single Stage SVM

Because the single stage SVM performs slightly better than the two stage SVM, in terms of an overall classification, this chapter is concluded with a compact overview on the performance measures achieved for all feature sets and dataset reduction stages in Table 3.18. As the $\hat{\boldsymbol{f}}_{\mathbf{0}}$-extension has proven itself to be beneficial for certain feature sets, each complete feature set with $\hat{\boldsymbol{f}}_{\mathbf{0}}$-extension is processed using the single stage SVM implementation presented in section 3.3 and the performance measures from subsection 3.3.1 are evaluated. The results in Table 3.18 are presented as the means of 100 classification iterations. The performance improvement, that comes with the dataset reduction is clearly visible and also the best measures are achieved using the combined feature set with $\hat{\boldsymbol{f}}_{\mathbf{0}}$-extension ($\mathscr{X}_{\text{combo\&}\hat{f}_0}$).

Table 3.18 *Summary of the performance measures, calculated for each complete feature set with $\hat{\boldsymbol{f}}_{\mathbf{0}}$-extension. Measures are displayed as means of 100 classification iterations.*

feature set	N_{FS}	**performance measure**				
		train. score	***test score***	***misclass. rate***	***overall score***	
$\mathscr{X}_{\text{MFCC\&}\hat{f}_0}$	36	93.48 %	71.57 %	29.55 %	89.1 %	*full inst.*
$\mathscr{X}_{\text{MPS\&}\hat{f}_0}$	10	87.63 %	70.91 %	29.34 %	84.29 %	
$\mathscr{X}_{\text{combo\&}\hat{f}_0}$	45	92.9 %	76.95 %	23.82 %	89.71 %	
$\mathscr{X}_{\text{MFCC\&}\hat{f}_0}$	36	94.15 %	71.57 %	29.98 %	79.67 %	*full exp.*
$\mathscr{X}_{\text{MPS\&}\hat{f}_0}$	10	89.08 %	69.37 %	32.04 %	76.01 %	
$\mathscr{X}_{\text{combo\&}\hat{f}_0}$	45	93.39 %	75.41 %	25.59 %	80.32 %	
$\mathscr{X}_{\text{MFCC\&}\hat{f}_0}$	36	94.52 %	76.93 %	24.88 %	87.37 %	*Red. I*
$\mathscr{X}_{\text{MPS\&}\hat{f}_0}$	10	91.25 %	77.6 %	22.48 %	86.92 %	
$\mathscr{X}_{\text{combo\&}\hat{f}_0}$	45	95.22 %	82.21 %	18.51 %	90.15 %	
$\mathscr{X}_{\text{MFCC\&}\hat{f}_0}$	36	96.54 %	80.23 %	21.4 %	85.62 %	*Red. II*
$\mathscr{X}_{\text{MPS\&}\hat{f}_0}$	10	94.82 %	82.23 %	18.24 %	87.98 %	
$\mathscr{X}_{\text{combo\&}\hat{f}_0}$	45	96.12 %	86.02 %	15.92 %	89.63 %	
$\mathscr{X}_{\text{MFCC\&}\hat{f}_0}$	36	96.66 %	81.6 %	19.11 %	85.61 %	*Red. III*
$\mathscr{X}_{\text{MPS\&}\hat{f}_0}$	10	95.99 %	87.95 %	13.12 %	88.98 %	
$\mathscr{X}_{\text{combo\&}\hat{f}_0}$	45	96.53 %	88.03 %	12.24 %	90.01 %	

4 Conclusion

This thesis provides proficient insight into the analysis and classification capabilities of different abstract feature sets with regards to different voice qualities in singing. The presented classification tasks are executed using a novel database created at the Institute of Electronic Music and Acoustics at the University of Music and Performing Arts Graz. The database comprises audio samples sung by 10 different professional singers. 5 vowels for 11 different pitches in 3 voice qualities (*breathy*, *normal* and *pressed*) were sung and recorded, whereby not all singers sang the same pitches. The pitch range differs for female and male singers. The resulting dataset, consists of altogether 1140 samples. The *instruction labels* reflect the voice qualities the singers were instructed to sing. A conducted listening experiment executed to rank the recorded samples with regard to the perceived voice quality and a subsequent *k-mediod cluster analysis*, yielded the *experiment labels*. Five dataset variations mentioned in section 3.2, are created, in order to obtain datasets containing samples that are more unambiguous. This is done by comparing *instruction* and *experiment labels* and discarding of samples that were not confidently rated as *breathy* or *pressed*. Another step necessary to create a dataset variation is the neglection of samples sung by male singers. For each of the dataset variations different feature sets are calculated. Two types of feature sets are dealt within this thesis. The proposed classification models, the implemented feature selection algorithm, as well as the classification performance and observations made during classification are summarized and discussed in section 4.1 and 4.2. The carried out analysis with regard to the different abstract features sets' classification capabilities, hold insight into their limitations and deficiencies, as well as their advantages.

4.1 Classification Model and Used Feature Sets

Inverse mel-filterbanks outperform the traditional mel-filterbanks

The first feature type comprises different variants of mel-frequency cepstral coefficients (MFCCs). The MFCC variants comprise the usage of 5 different filterbank types, 2 types of center frequency modification (vocal tract length normalization and vocal tract length pertubation) as well as an optional cepstral lifter. The resulting 30 different MFCC variants are narrowed down to one MFCC variant in a pre-analysis, in which 35 MFCC coefficients of each MFCC variants are used to classify the dataset variation in which *instruction* and *experiment labels* coincide. Four performance measures, namely the *training* and *test score*, the *misclassification rate*, and the *overall score* are calculated for the executed classification. The visualization of the performance measures depicted in Figure 3.8 and Figure 3.9 show that the MFCC variant where an inverse mel-filterbank, without center frequency augmentation and an activated cepstral lifter delivers the best results amongst the MFCC variants. The idea of inverting the mel-filterbank stems from [20], where it is shown that the performance of automatic speech recognition for high pitched speakers such as children can be enhanced. The filterbank inversion leads to a higher resolution towards higher frequencies, in contrast to the classic mel-filterbank approach which exhibits a denser frequency spacing at lower frequencies. The improved classification results for the inverse filterbanks confirm that a higher resolution towards higher frequencies is also beneficial with regard to the distinction of the voice quality in singing. This also coincides with the conclusions made in [45], where it is shown that the high frequency energy in speech and singing holds information on the perceived quality of voice. Hence, a better resolution

towards higher frequencies when averaging a signal's frequency spectrum using filterbanks makes sense.

Apart from the filterbank variation, the other MFCC augmentation methods do not show any improvments with regard to voice quality classification performance using MFCCs. The VTLP method, proposed in [16], visualized in the second subplot of Figure 3.8 and Figure 3.9, performed worse. Also the MFCC augmentation using the frequency warping approach VTLN from [23], stayed behind the unaugmented cepstral coefficients in terms of classification performance. Based on theoretical considerations made in subsection 2.2.2, a correlation between vowels and the estimated frequency warping factors α_{VTLN} is anticipated. However, as shown in Figure 3.7, the correlation towards the vowels is not given. α_{VTLN} seems to show a dependence towards the pitch rather than the vowel which is one reason for the missing improvement. Another source for the worse performance with VTLP and VTLN can be found in their origin. Both augmentation methods originate in automatic speech recognition, the augmentations are designed with the intention of improving the MFCC's descriptive content regarding vowels. This is done by compromising the influence of different vocal tract lengths, because vowels are shaped by the vocal tract. In contrast to vowels the phonation type is a voice characteristic, for which the voice source located with the glottis is mainly responsible [57]. Thus, an anticipated improvement with regard to voice quality classification, by diminishing the influence of the vocal tract with VTLN and VTLP, could not be detected.

MPS based features are introduced

Building on the studies mentioned in [56] which state that the human auditory system is perceptually sensitive towards joint temporal and spectral modulation, a signal representation containing either one or both types of modulation (temporal and spectral) has proven useful in the distinction of natural sounds [56], in speech detection [26] and also in the classification of voice disorders [25]. The second feature set approach is based on the modulation power spectrum (MPS). The MPS is defined as the 2D-Fourier Transform of a time-frequency signal representation (spectrogram). Based on the `Matlab` code provided in [44], the MPS of each audio sample contained in the dataset is estimated. The MPS are summed along their temporal and spectral modulation axis, resulting in the summed temporal modulation power spectrum (STMPS) and the summed spectral modulation power spectrum (SSMPS). The STMPS and SSMPS are freed of their functional trend by subtraction of a fitted polynomial, yielding the STMPS- and SSMPS-residual, whose peaks serve as a basis for the derived MPS-based features mentioned in subsection 3.4.3. In an exemplary analysis, the weakned occurence of peaks in the temporal modulations for *breathy* phonation has been shown. This is mainly caused by a diminished vibrato. Therefore, it can be argued, that the lack of vibrato for *breathy* phonation is also due to the missing tension on the vocal folds, mentioned in subsection 1.2.2. On the other hand, the summed spectal modulation residuals exhibit detectable differences in peak height and the peak height decline over the spectral modulations, which seemed to correlate with the voice quality. As shown in Figure 3.13 and in section B.2, the peaks located at $\frac{2}{\hat{f}_0}$ show the highest peak amplitude for *pressed* phonation followed by *normal* and *breathy* phonation. The spectral modulations result from a Fourier transform of the logarithmized overtone spectrum as mentioned in section 2.3, which is equal to a cepstrum [44]. The cepstrum of a spectrum with whole-number multiple harmonics, as it is present for speech signals, also exhibits peaks at whole-number multiples of the fundamental period. This means that if a peak in the spectral modulations is more prominent, it also points to more distinct overtone peaks. A possible explanation for this is the high tension on the vocal folds connected to *pressed* voice quality, which results in a very tense glottal closure process, which introduces more distinct overtones.

Two SVM structures are implemented

The `Matlab` command `fitcecoc()` [31] builds the core of two SVM implementations presented in section 3.3. The first structure is a single stage SVM, which is able to execute multi-class classification using three binary SVM learners and the "one-versus-all " strategy. The second SVM structure is designed in two stages. The idea behind the two stage structure is the reduction of the three-class classification problem to two binary classification problems. The first stage deals with the classification of the samples into the *breathy* or *rest* class, which contains the samples of both the *normal* and *pressed* class. The goal of the second stage is the separation of *normal* and *pressed* samples. The single stage SVM allows an evaluation of each processed feature set on its own and the two stage SVM allows the categorization of the *breathy* class separately. Depending on which SVM structure and which dataset variation is used the underlying dataset is balanced using random undersampling, in order to ensure an equal number of samples per class. The resulting number of samples for the single stage SVM, after random undersampling is summarized in Table 3.10. The number of samples resulting for the usage of the two stage SVM after dataset balancing is listed in Table 3.11.

The MFCC based feature set $\mathscr{X}_{\text{MFCC}}$, the MPS-based feature set $\mathscr{X}_{\text{MPS}}$ and a combined version $\mathscr{X}_{\text{combo}}$ is analyzed with a Plus-L Minus-R feature selection algorithm (L-R selection) in combination with the implemented SVM structures. This allows an assessment on which features are more descriptive towards the phonation types. The best performing feature set is utilized for the full dataset and processed through the single and two stage SVM once with *instruction* and once with *experiment labels*, allowing an assessment on whether the ML classification procedure prefers the *instruction* or *experiment labels*. Additionally, comparisons towards the ranking behaviour observed in the listening experiment can be drawn. The performance comparison based on the feature sets L-R selection as well as results of the full dataset analysis are summarized and discussed in section 4.2.

4.2 Feature Selection Analysis and Performance Evaluation

The feature selection algorithm is carried out to exploit every possible number of reduced feature sets. Meaning the L-R selection yields a feature set for every reduced number of features, from a single feature to all but one feature of the analyzed feature set. For the single stage SVM, the MFCC feature set $\mathscr{X}_{\text{MFCC}}$, the MPS-based feature set $\mathscr{X}_{\text{MPS}}$ and the combined feature set $\mathscr{X}_{\text{combo}}$ are all subject to the L-R selection, which chooses the most descriptive features for the three-class classification, which is possible with the single stage SVM. The L-R selection chooses the features in an L-times execution of the sequential forward selection and a R-times execution of the sequential backward selection, based on the discriminant potential of each feature discussed in section 2.6. With regard to the two stage SVM the feature selection is only carried out for $\mathscr{X}_{\text{combo}}$.

Feature selection: MPS features are preferred for classification with single stage SVM

The feature selection procedure for three classes and the single stage SVM allow a calculation of the performance measures for each reduced feature set, yielding performance measure progressions across an increasing number of selected features. These progressions depicted in Figure 3.14, 3.15 and 3.16 show that the MPS-based features outperform the augmented MFCCs. The *test*, *training*, and *overall score* progressions created with the MFCC feature set all exhibit lower values than the ones for the MPS based and combined feature sets. Additionally, the MFCC feature set exhibits the largest mismatch between the test and *training score*, which indicates that the MFCC feature set also bears the most overfitting, due to the deficient or redundant information content of the MFCCs with regard to the phonation type. Nevertheless, one aspect of the MFCCs has to be emphasized, which is that the MFCCs do not show any dependence towards the fundamental frequency. This can be derived

from the performance measure progressions in Figure 3.14 (a), 3.15 (a) and 3.16 (a), because apart from minor deviations towards lower N_{FS}, the performance progressions for the feature set with and without $\hat{\boldsymbol{f}}_{\mathbf{0}}$ display identical courses. The MPS based feature set shows clear improvements when $\hat{\boldsymbol{f}}_{\mathbf{0}}$ is included. Another conclusion for the MPS-based feature set can be drawn from the *overall score* of $\mathscr{X}_{\mathrm{MPS}}$. There, the *overall score* for the feature set with $\hat{\boldsymbol{f}}_{\mathbf{0}}$-extension lies below the *overall score* for unextended feature sets. This seems to contradict the higher *test score* and lower *misclassification rate*, which are simultaneously present for $\mathscr{X}_{\mathrm{MPS}\&\hat{f}_0}$. This circumstance is even more prominent when looking at the results of the *third dataset reduction* in Figure 3.16. An explanation can be found when looking at the data used to fit the SVM models and the data underlying the *overall score's* calculation. Due to random undersampling the *third dataset reduction* is balanced in regards to the voice quality, however, imbalanced with regard to the fundamental frequency, yielding SVM models which are fitted towards certain fundamental frequencies. The imbalanced dataset behind the *overall score*'s calculation (see Figure 3.5), contains samples with fundamental frequencies which, were not included in the SVM fitting procedure. This means that the SVM models process samples with certain fundamental frequencies which they have not seen before. Thus, it can be stated that the fundamental frequency influences the classification performance especially for low dimensional MPS-based feature sets. As this dependency is also observed for low dimensional combined feature sets, it is obvious that the features selected first in the L-R selection of $\mathscr{X}_{\mathrm{combo}}$ are MPS-based features. This presumption is confirmed when looking at the feature selection tables in Appendix C, which clearly show that, if the combined feature set is used, the MPS based features are selected first, for every dataset reduction.

Analyzing the effects of an increasing number of features, it can be observed that for all analyzed feature sets the increase in dimensionality did not result in an increase of *overfitting*, indicated by the mismatch between the *training* and *test score*. With increasing number of features the deviation between *training* and *test score* either remains the same or even shrinks in all cases. The smallest deviation between *test* and *training score* is given for the combined feature set. As shown in Table 3.18, the performance measures improve with every dataset reduction stage, reaching the best performance measures for the combined feature set $\mathscr{X}_{\mathrm{combo}}$.

Feature selection: two stage SVM shows better classifiabiltiy for *breathy* phonation

The two stage SVM was created to assess which classes can be identified more easily. Figure 3.17 to 3.19, as well as the 2D-LDA projection in Figure 3.20 prominently show that the distinction of *breathy* voice quality, exhibits higher accuracy than the distinction between *normal* and *pressed* phonation. For all reduction stages *misclassification rates* of between $5\,\%$ to $10\,\%$ are achieved for the first stage responsible for the distinction of *breathy* voice quality. For the classification of *normal* and *pressed* voice quality *misclassification rates* are 3 to 4 times as high. The Plus-L Minus-R selection algorithm, for the two stage SVM, is now applied in both classification stages on the combined feature set. In section 3.6 the performance on the full dataset is analyzed when comparing the results of the single and two stage SVM in terms of the depicted confusion matrices. The single stage SVM shows fewer misclassifications for both the *instruction* and *experiment labels* than the results for the two stage SVM. A reason for this can be found in the structure of the two stage SVM, due to the fact that only the samples that are deemed to belong to the *rest* class in the first stage are further processed and then separated into *normal* and *pressed* class, which inevitably leads to an error propagation from the first to the second stage of the SVM.

SVM classification using the full dataset with *instruction labels* achieves better results

Generally, it is shown that the performance for both SVM structures increases with each dataset reduction. This holds the insight that the misclassifications within the listening experiment might not be of perceptual nature, but rather due to difficulties in the instruction execution of the singers. Additionally, the versatile evaluation possibilities that come with the novel database are undermined by the creation of the different dataset reductions. Also the feature selection reveals that the best results are achieved for all features available and the $\hat{f}_0$ extension shows benefits with regard to the MPS based feature set and the combined feature sets, for lower numbers of selected features. Subsequently, the analysis of the full dataset, which allows a comparison of the two SVM structures in terms of the overall classification, shows that the single stage SVM performs slightly better on the full dataset. Thus, the results of the single stage SVM classification on the full dataset's hold-out set (test set) with varying labels are now compared with the rating behaviour of the listening experiment. The SVM model which exhibits the highest *test score* of 100 carried out classification iterations is chosen and the respective hold-out sets are classified. Confusion matrices are created and normalized to the overall number of samples in the set, which due to RUS is given with 178 samples for the hold-out set with *experiment labels* and 228 samples for the usage of *instruction labels*. The classification on the hold-out set simulates a classification process in which the SVM-model is presented new data, this creates a situation is compareable to the results of the listening experiment. But as the results of the listening experiment comprise all samples of the dataset, confusion matrices with relative values are compared.

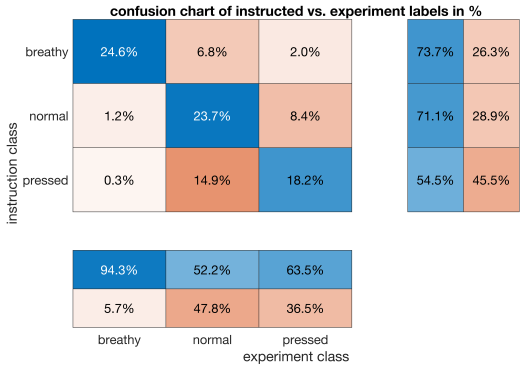
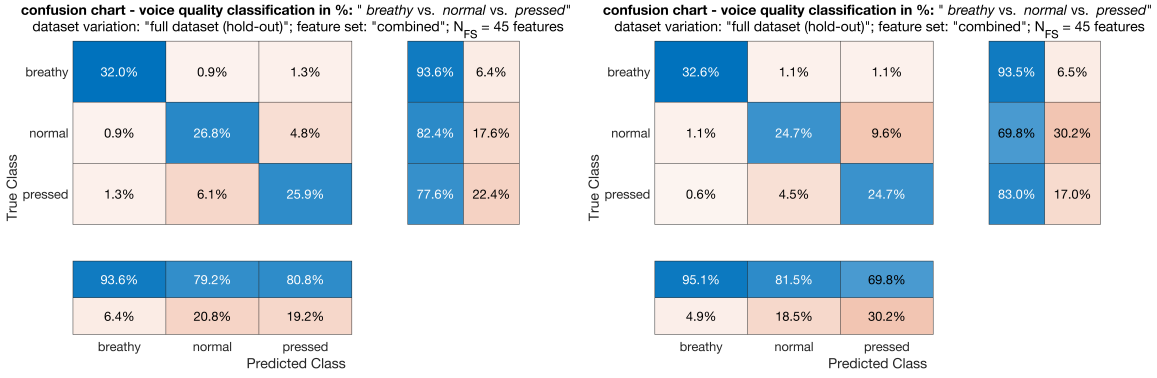


Figure 4.1 *Confusion matrix of instruction vs. experiment labels, values in %, absolute values are normalized to 1140 samples.*



(a) *instruction labels* (b) *experiment labels*

Figure 4.2 *Confusion matrices in percent, full dataset's hold-out set classification with instruction labels (a) and experiment labels(b), for the single stage SVM model.*

Figure 4.1 shows the same results as Figure 3.2, with the difference that all absolute values are normalized with the total number of 1140 samples. Thus, Figure 4.1 and Figure 4.2 are comparable. In doing so, it has to be pointed out, that even though both confusions matrices show *test scores* that are in the same vicinity percentagewise (checkable by summing the values on the main diagonal of Figure 3.21), the ML-classification model works slightly better for the full dataset with *instruction labels* as shown in Table 3.18, because the statistical variation behind the *test score* brought in by random undersampling for the usage of *experiment labels* and the random data cannot be neglected. This becomes clearer when comparing the *misclassification rate* achievable with the full dataset in Table 3.18. The SVM models behind Figure 3.21 are the ones exhibiting the highest *test score* amongst 100 classification iterations which represent best case scenario out of these 100 iterations. However, the statistically more stabilized classification properties are recogniseable in the performance measure means of Table 3.18. The improved classification results when using the *instruction labels*, are also visible in the confusion matrices for a classification of the full dataset appended in Figure B.6.

ML-based classification of full dataset with *experiment labels* is comparable to listening experiment rating

Confusion charts created on the basis of the rating behaviour of the listening experiment (see Figure 4.1) resemble the misclassifications occuring within the ML-based classification, when *experiment labels* are used. This can already be seen in Figure 3.21, although still in small scale. However, for Figure B.6 (b) and Figure B.7 (d) depicting the confusion matrices created for the full dataset classification with single and two stage SVM respectively, the resemblances between the misclassification behaviour of the listening experiment and the ML classification when using the *experiment labels* are not deniable. This indicates that a SVM model trained with the full dataset and the labels retrieved from the listening experiment, exhibit similar misclassifcation behaviour as observed for the listening experiment. For both classification procedures (the listening experiment and the ML-based classification), the class with the most correctly classified samples is the *breathy* class. Most misclassifications occur for the distinction between *pressed* and *normal* classes, whereas in the listening experiment more samples, where a *normal* phonation was instructed, were ranked with *pressed* voice quality. This is also true for the classification of the hold-out set with *experiment labels* visualized in Figure 4.2.

Under the assumption that the *instruction labels* are the ground truth and the *experiment labels* are the results of a classification task carried out by the listeners, the rating results are comparable to the classification task carried out on the hold-out set. If one compares Figure 4.1 and Figure 4.2, the ML-based classification carried out on the hold-out set shows better results than the confusion chart created from the listening experiment in Figure 4.1. The samples where *experiment* and *instruction labels* coincide amount to $66.5\,\%$. The best case hold-out classifications depicted in Figure 4.2, exhibit of $84.7\,\%$ for the usage of *instruction labels* and $82\,\%$, if *experiment labels* are employed. The statement that the ML-based classification task outperforms the human classification apprehended from the listening experiment, is only true for the underlying data, if the *instruction labels* are considered to be the ground truth. This raises the fundamental question, if the ground truth lies with the *instruction labels* or the *experiment labels*. A possible answer to this can be found in a more sophisticated analysis of the experiment's results in order to assess, where the misclassifications within the listening experiment stem from. One the one hand, the instructions given to the professional singers might have been hard to execute, because in some cases certain types of phonations are difficult to produce e.g. *breathy* for high pitches, as it is natural that singers tend to use *pressed* phonation for higher

pitches [57]. On the other hand, the misclassifications observable in the ratings might be explained by a more thorough analysis of the experiment results with a closer look at each listener, also considering the listeners' professional backgrounds or experiences with singing voices. This would result in a more sophisticated evaluation of the experiment results including e.g. possible correlations between rating behaviour, deviating *instruction* and *experiment labels* occur. Nevertheless, the analysis carried out in this thesis was not designed to answer these questions.

Remarks on the present overfitting

During the first overview classification analysis of the MFCCs in Figure 3.8, the analysis of the full dataset carried out in section 3.6 and also the feature selection analysis of the *first dataset reduction* in subsection 3.5.2, a mismatch between the *training* and *test score* due to overfitting is mentioned. There are two possible sources of this mismatch. Either it originates from poorly chosen SVM parameters (kernel scale γ or box constraint C) or from the feature set itself, e.g. due to the *curse of dimensionality*. However, the feature selection analysis carried out in section 3.5 showed that the dimensionality increase in the feature set does not contribute to the overfitting in a drastic manner, as the *training* and *test score* increase equally strong for higher dimensional feature spaces. In the two stage analysis of subsection 3.5.3, the large mismatch does not occur for the classification of *breathy* phonation but for the distinction of *normal* and *pressed* voice quality. This indicates that the underlying feature sets exhibit a higher variance concerning the classes *normal* and *pressed* leading to a worse separability. This is also undermined when looking at the LDA transformed features in Figure 3.20. Thus, a major contributor towards the present overfitting is found with the high variance in the feature set regarding the classes *normal* and *pressed*. However, it still begs the question on how large the influence of the SVM parameters γ and C are. As the estimation of the kernel-scale γ using the iterative procedure presented in section 3.3 already provides a strategy which adaptively selects a reasonable γ with respect to the used feature set, the influence of γ is neglectable. However this still leaves the influence of the box constraint C which was fixed with $C = 1$ for all SVM classification procedures carried out in this thesis. In order to assess this influence the initial MFCC overview analysis Figure 3.8 was carried out again, as it showed the largest mismatch between *training* and *test score*. For the additional analysis, whose results are to be found in section B.1, the box constraint was diminished by a factor of 100 to $C = 0.01$. When comparing the results of the MFCC variants with inverse filterbanks from Figure 3.8 with Figure B.1, it is detectable that the inital mistmatch in Figure 3.8 (ca. $18\,\%$) is reduced to approximately $11\,\%$ in Figure B.1, which is a $7\,\%$ drop of the mistmatch between *training* and *test score*. However, the *test score* achieved with $C = 0.01$ is $10\,\%$ lower than for $C = 1$. Thus, the decrease of the *test score* when using $C = 0.01$ is higher than the achievable drop in the deviation of *training* and *test score*. This shows that although $C = 1$ is a contributor to overfitting for certain feature sets, the benefits in regards to the *test score* still outweigh the increased mismatch. Finally, it can be argued that the mentioned variance within the data of the classes *normal* and *pressed*, is the main contributor and the box constraint $C = 1$ only plays a minor role in the observed deviation between *training* and *test score*.

In conclusion, the proposed novel abstract feature set proves itself to be very informative and outperformed the classic MFCC based approach with regard to voice quality classification. If the assumption is made that the *instruction labels* provide the ground truth behind the recorded sung vocal samples with different phonation types, the ML-based classification approach delivers better results than the equivalent classification procedure which is carried out during the listening experiment. However, it has to be kept in mind that the physiological processes of generating different phonation types are also subject to physical limitations, which become noticable especially for rising pitches. This, for example, can result in phonation types that are involuntarily changed. For instance, the usage of *pressed*

phonation is in combination with high pitches [57]. It is also possible that these involuntary changes are perceivable by humans and the *experiment labels* are closer to the ground truth, although the ML-based classification suggests that the data better corresponds to the *instruction labels* of the abstract features within the spanned feature space. However, the human auditory system and its capabilities should never be underestimated.

4.3 Outlook and Suggestions for Future Research

In order to provide a compact overview of possible improvements and future areas of research in the field of phonation type classification in singing, the following paragraphs hint at the areas exhibiting potential of improvement.

Further evaluation of the results of the listening experiment

The analysis of the results of the listening experiment, still leaves a lot of questions unanswered. The most prominent is, where the reason for the misclassifications originates and if there exist correlations between observable misclassifications and certain singers, vowels or pitches. This could provide important insight in answering the question of which labels are closer to the ground truth. Thus, a further analysis of the listeners' side of the conducted listening experiment, is recommended.

Analysis of sung vocal signals with modulation power spectrum

Using the modulation power spectrum signal representation, one assumption is made, when the temporal features are calculated. The STMPS-residual peaks for negative temporal modulations are discarded. By doing so, the MPS is assumed to be symmetrical, which in terms of sung vocal signals would mean that the signal contains equally strong up and down sweeps, as it would occur for a vibrato, where the upwards pitch movement is equally distinct as the downwards pitch movement. A way of analyzing the symmetry with a measure is proposed in [56]. If the assumption that the MPS are not symmetrical does not hold, the half containing the negative temporal modulations which include the up sweeps might also hold vibrato-related information on the voice quality. Another aspect concerning the modulation power spectrum is the possibiliy it holds with regard to sound manipulation. Certain manipulations have already been discussed in [44]. A procedure that is often used in speech signal processing in order to estimate the source signal present at the glottis is the glottal inverse filtering [7], which makes sense when considering the source filter model that is mostly presupposed in a speech context and that the origin of voice quality is found at the voice source. Nonetheless, as shown in [4], the inverse filtering algorithms based on LPC estimation of the vocal tract are strongly limited with higher fundamental frequencies. Due to the fact that within the MPS domain the source and filter are also separated, an inverse filtering procedure applied in the MPS domain could also result in a source signal estimation which could be further processed in a classification task. Another feature extraction strategy, which could yield important voice quality descriptive features is the strategy proposed in [25], which applies a dimensionality reduction scheme directly onto a modulation-based signal representation. This strategy could be applied onto the modulation power spectra, meaning that the 2D-MPS could be reduced in its dimensionality with a principle component analysis (PCA) or a linear discriminant analysis (LDA) in order to retrieve a condensed form of the MPS, whose coefficients are then usable as abstract features. Additionally, there might also still be useful information left within the STMPS- and SSMPS-residuals, which allow the derivation of additional peak based feature which might yield classification improvements.

Improvements for other feature sets and classification model

Concerning the MFCC feature set, the vocal tract length normalization based on the frequency warping approach might be improved if other reference MFCCs were chosen, because the averaged reference MFCCs did not prove to be sufficient. As shown in subsection 2.2.2, the frequency warping factor estimation works best, if samples containing similar vowels are used. If one reference singer is chosen, a sample for each vowel, ideally with the same pitch, could be used to estimate the frequency warping factor. There are also other normalization approaches, e.g. the ΔF-method proposed in [17] or the algorithm proposed in [62], which even allows the estimation of the vocal tract shape. Apart from the voice quality classification, a vowel classification/detection is also possible with the presented dataset, because not only has the voice quality been rated during the listening experiment but also were perceived vowels rated. For a vowel classification/detection a diminished influence of the vocal tract length might prove itself to be very beneficial.

Regarding the ML classification model, the analysis focusing on the model parameters has been kept to a minimum. A closer look at a potential hyperparameter optimization strategy, including different kernel functions and especially one which allows the estimation of an optimal box constraint dependent on the feature set, could still influence the classification positively. Additionally, it has been shown that if the MPS-based features exhibit a fundamental frequency dependence, especially for lower dimensional feature sets. A dataset balancing with regards to the fundamental frequency, or generally a more sophisticated dataset balancing scheme, might also present potential improvement.

Potential towards real time application analysis environment

The implemented MPS calculation in this thesis, relies on the whole signal sample. Further investigation carried out solely on the MPS in combination with sung vocal signals could provide insight into the influence of blocking parameters, such as the window length and hopsize, which are fundamental for the calculation of the modulation power spectra. However, it is important that for the MPS the period of the temporal modulations has to be taken into account and the blocking parameters have to be chosen accordingly. In contrast to the time-frequency representation of vocal signals in form of spectrograms, where the fundamental frequencies revolve around frequency ranges of $f_0 \in [70\,\mathrm{Hz}, 1480\,\mathrm{Hz}] \approx [\mathrm{D}, \mathrm{fis}^3]$, average vibrato frequencies are given with of $f_{\text{temp}} \in [4\,\mathrm{Hz}, 7\,\mathrm{Hz}]$ [14]. For instance, if a vibrato frequency of $f_{\text{temp}} = 5\,\mathrm{Hz}$ is assumed the resulting period is given with $T_{\text{temp}} = \frac{1}{f_{\text{temp}}} = \frac{1}{5} = 200\,\mathrm{ms}$, which are way larger periods, on whose basis the blocking parameters have to be chosen. Additionally to the vibrato periods, the used fundamental frequency tracker presented in section 2.4 requires a minimum block-length of $80\,\mathrm{ms}$, so if smaller block lengths are used, the f_0-tracker needs to be adapted, e.g. with a real-time capable tracker as presented in [8]. Another aspect which could improve the feature calculation of the MPS-based features proposed in this thesis, is the direct calculation of the summed spectral and temporal modulations from the spectrogram. Because the features are derived from the summed MPS, the 2D-representation is not implicitly necessary. A reduction to two 1D-Fourier transforms operations could decrease the complexity of the feature extraction procedure.

As shown, the analysis and classification of voice quality in singing still exhibits various areas that yield the potential of future research. It is also obvious that the modulation based signal representation given with the MPS holds vivid information concerning the phonation type in singing, which might also be transferable to research areas such as the classification of voice disorders or other biomedical applications. Nonetheless, it has been shown that for phonation type classification in singing, the modulation power spectrum and its derived features have proven themselves a worthy competitor to already well-established feature sets, such as the mel frequency cepstral coefficients.

Appendix A Additional Tables

A.1 Dataset Reduction Distribution of Discarded Samples

A.1.1 First Dataset Reduction

Table A.1 *Statistics on the discarded samples of the first dataset reduction: voice quality.*

#(samples)	**voice quality** (instruction)		
	breathy	***normal***	***pressed***
total number of samples	380	380	380
number of discared samples	100	110	173
percentage of discared samples	26.32 %	28.95 %	45.53 %

Table A.2 *Statistics on the discarded samples of the first dataset reduction: singers.*

#(samples)	**singers**				
	S1 (f)	**S2** (f)	**S3** (m)	**S4** (f)	**S5** (f)
total number of samples	135	135	90	135	135
number of discared samples	55	38	19	44	43
percentage of discared samples	40.74 %	28.15 %	21.11 %	32.59 %	31.85 %
#(samples)	**singers**				
	S6 (f)	**S7** (m)	**S8** (m)	**S9** (m)	**S10** (f)
total number of samples	135	90	90	60	135
number of discared samples	41	36	50	24	33
percentage of discared samples	30.37 %	40 %	55.56 %	40 %	24.44 %

Table A.3 *Statistics on the discarded samples of the first dataset reduction: vowels.*

#(samples)	**vowels**				
	/a/	***/e/***	***/i/***	***/o/***	***/u/***
total number of samples	228	228	228	228	228
number of discared samples	73	76	73	82	79
percentage of discared samples	32.02 %	33.33 %	32.02 %	35.96 %	34.65 %

Table A.4 *Statistics on the discarded samples of the first dataset reduction: pitches.*

#(samples)	pitches					
	c	**g**	**c^1**	**e^1**	**g^1**	**a^1**
total number of samples	60	60	150	150	135	135
number of discared samples	23	23	40	44	42	44
percentage of discared samples	38.33 %	38.33 %	26.67 %	29.33 %	31.11 %	32.59 %

#(samples)	pitches				
	c^2	**d^2**	**e^2**	**g^2**	**a^2**
total number of samples	90	90	90	90	90
number of discared samples	28	23	34	36	46
percentage of discared samples	31.11 %	25.56 %	37.78 %	40 %	51.11 %

A.1.2 Second Dataset Reduction

Table A.5 *Statistics on the discarded samples of the first dataset reduction: voice quality.*

#(samples)	voice quality (instruction)		
	breathy	***normal***	***pressed***
total number of samples	380	380	380
number of discared samples	219	169	280
percentage of discared samples	57.63 %	44.47 %	73.68 %

Table A.6 *Statistics on the discarded samples of the first dataset reduction: singers.*

#(samples)	singers				
	S1 (f)	**S2** (f)	**S3** (m)	**S4** (f)	**S5** (f)
total number of samples	135	135	90	135	135
number of discared samples	99	63	31	76	75
percentage of discared samples	73.33 %	46.67 %	34.44 %	56.30 %	55.56 %

#(samples)	singers				
	S6 (f)	**S7** (m)	**S8** (m)	**S9** (m)	**S10** (f)
total number of samples	135	90	90	60	135
number of discared samples	79	60	70	42	73
percentage of discared samples	58.52 %	66.67 %	77.78 %	70 %	54.07 %

Table A.7 *Statistics on the discarded samples of the first dataset reduction: vowels.*

#(samples)	vowels				
	/a/	*/e/*	*/i/*	*/o/*	*/u/*
total number of samples	228	228	228	228	228
number of discared samples	127	137	127	144	133
percentage of discared samples	55.7 %	60.09 %	55.70 %	63.16 %	58.33 %

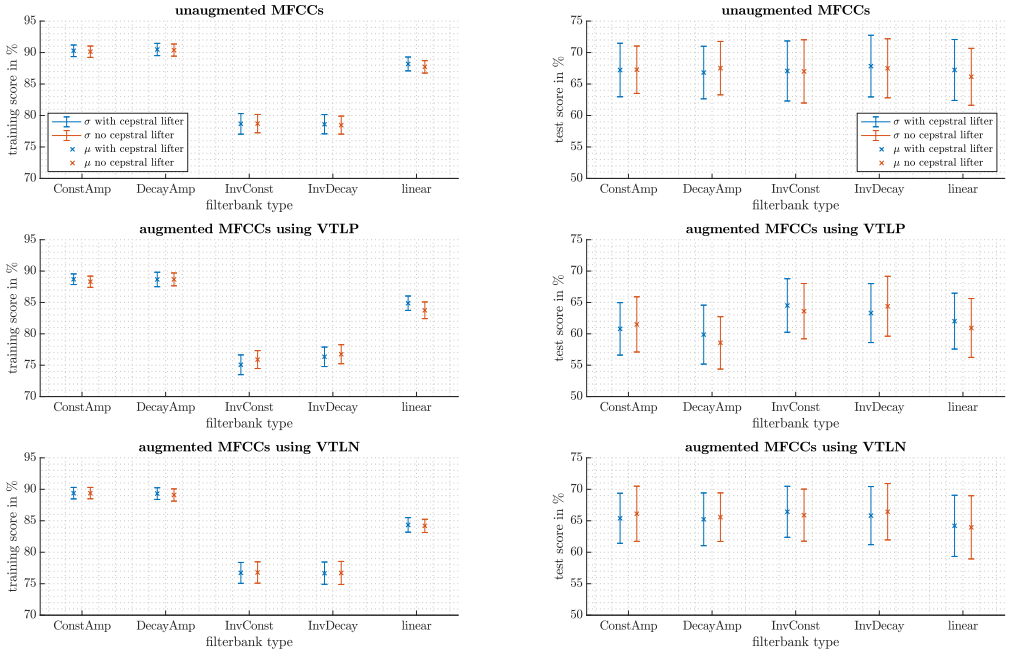
Table A.8 *Statistics on the discarded samples of the first dataset reduction: pitches.*

#(samples)	pitches					
	c	**g**	$\mathbf{c^1}$	$\mathbf{e^1}$	$\mathbf{g^1}$	$\mathbf{a^1}$
total number of samples	60	60	150	150	135	135
number of discared samples	36	37	78	81	78	81
percentage of discared samples	60.00 %	61.67 %	52.00 %	54.00 %	57.78 %	60.00 %

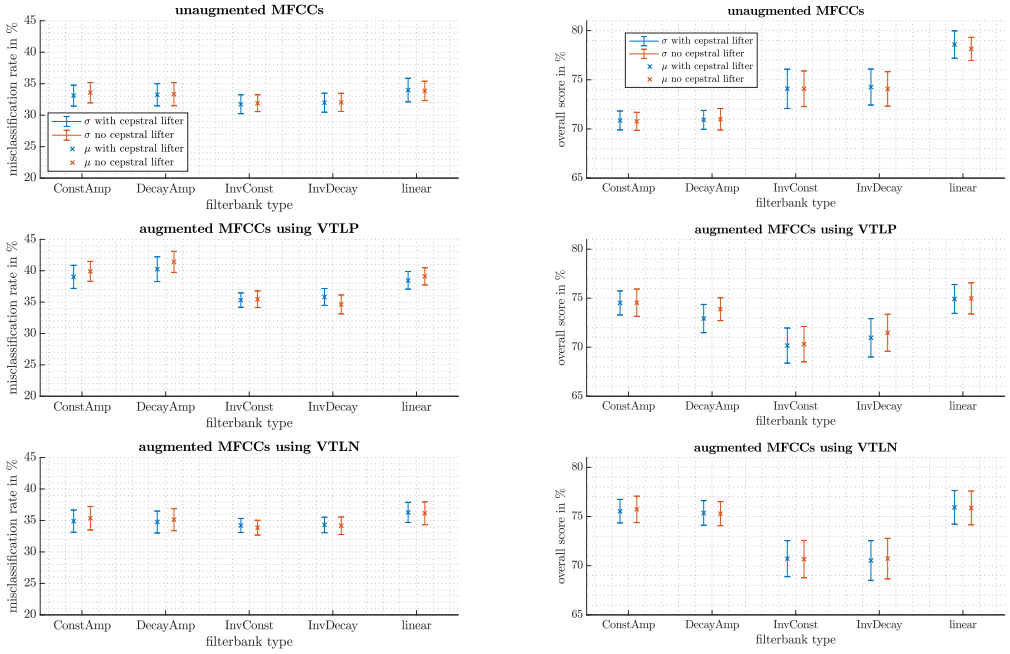
#(samples)	pitches				
	$\mathbf{c^2}$	$\mathbf{d^2}$	$\mathbf{e^2}$	$\mathbf{g^2}$	$\mathbf{a^2}$
total number of samples	90	90	90	90	90
number of discared samples	47	49	57	62	62
percentage of discared samples	52.22 %	54.44 %	63.33 %	68.89 %	68.89 %

Appendix B Additional Plots

B.1 Performance of MFCC variations with box constraint $C = 0.01$



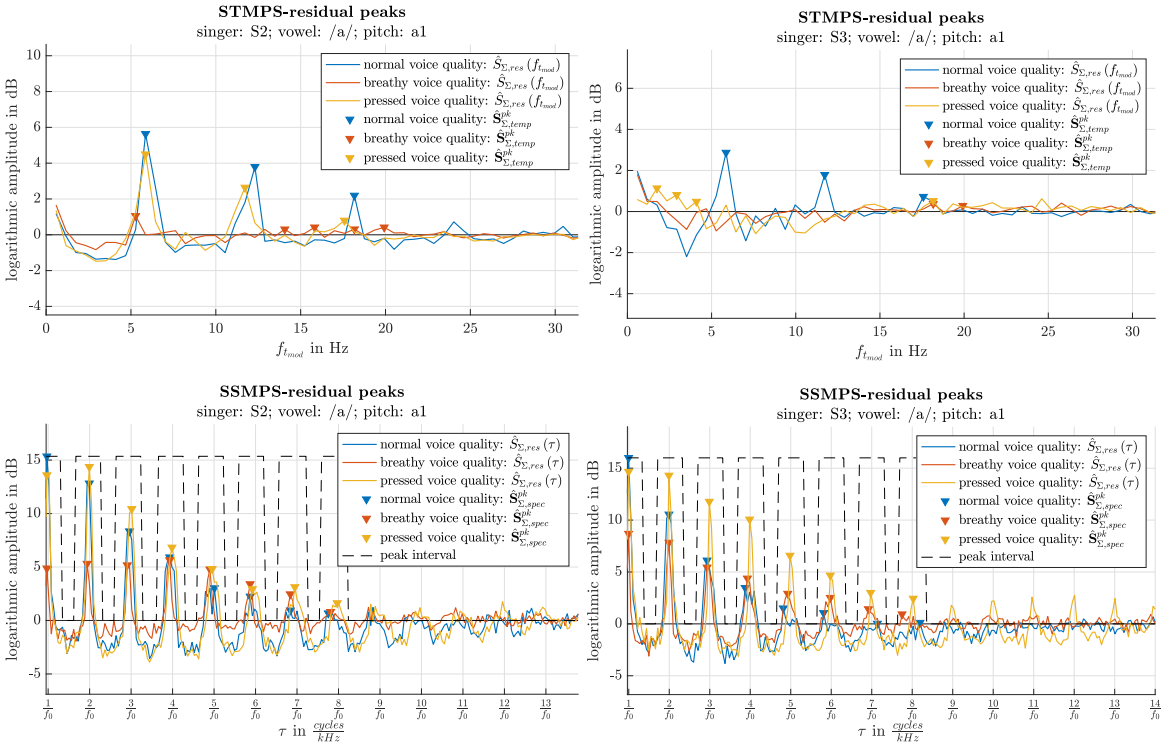
(a) *training score* (b) *test score*



(c) *misclassification rate* (d) *overall score*

Figure B.1 *Performance overview on MFCC variations for smaller box constraint $C = 0.01$*

B.2 Summed Modulation Power Spectrum-residual peaks



(a) $\hat{S}_{\Sigma,\text{res}}(\tau)$ and $\hat{S}_{\Sigma,\text{res}}(f_{t_{\text{mod}}})$; singer: S2, vowel: */a/* (b) $\hat{S}_{\Sigma,\text{res}}(\tau)$ and $\hat{S}_{\Sigma,\text{res}}(f_{t_{\text{mod}}})$; singer: S3, vowel: */a/*

Figure B.2 *Picked peaks of STMPS and SSMPS-residual for exemplary samples.*

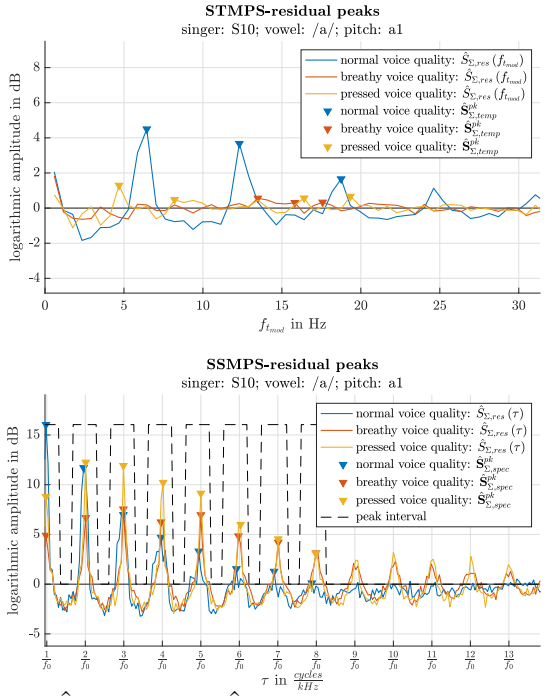
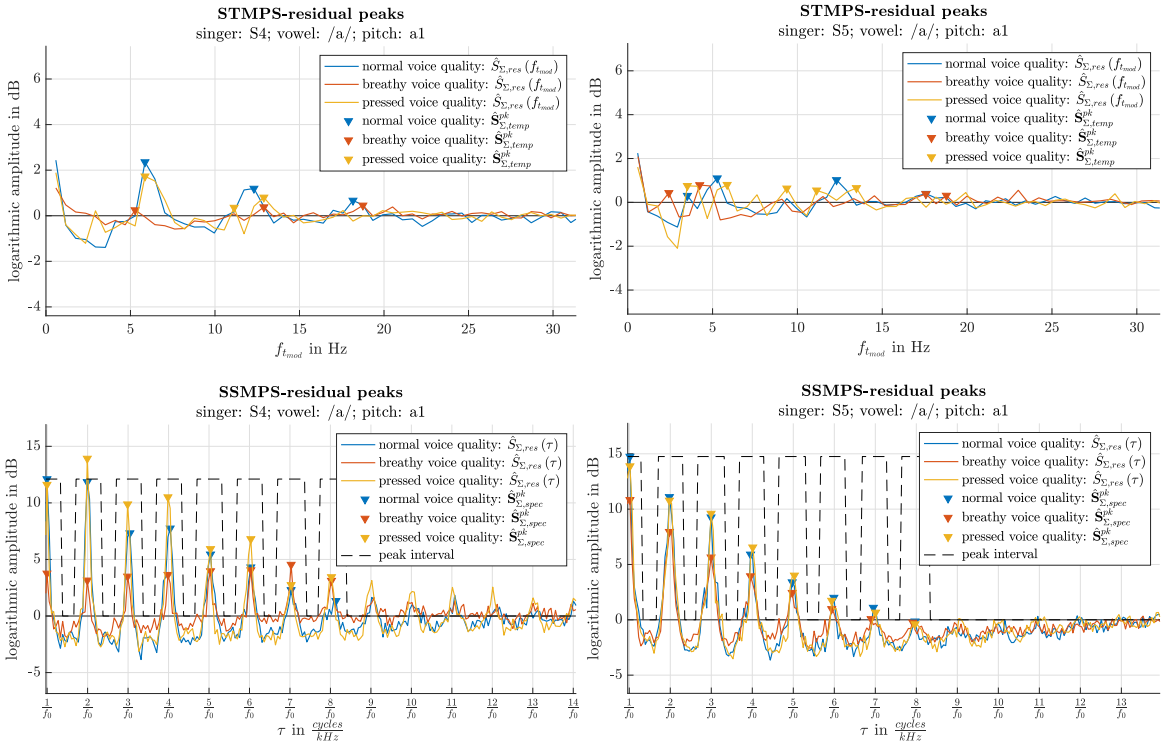
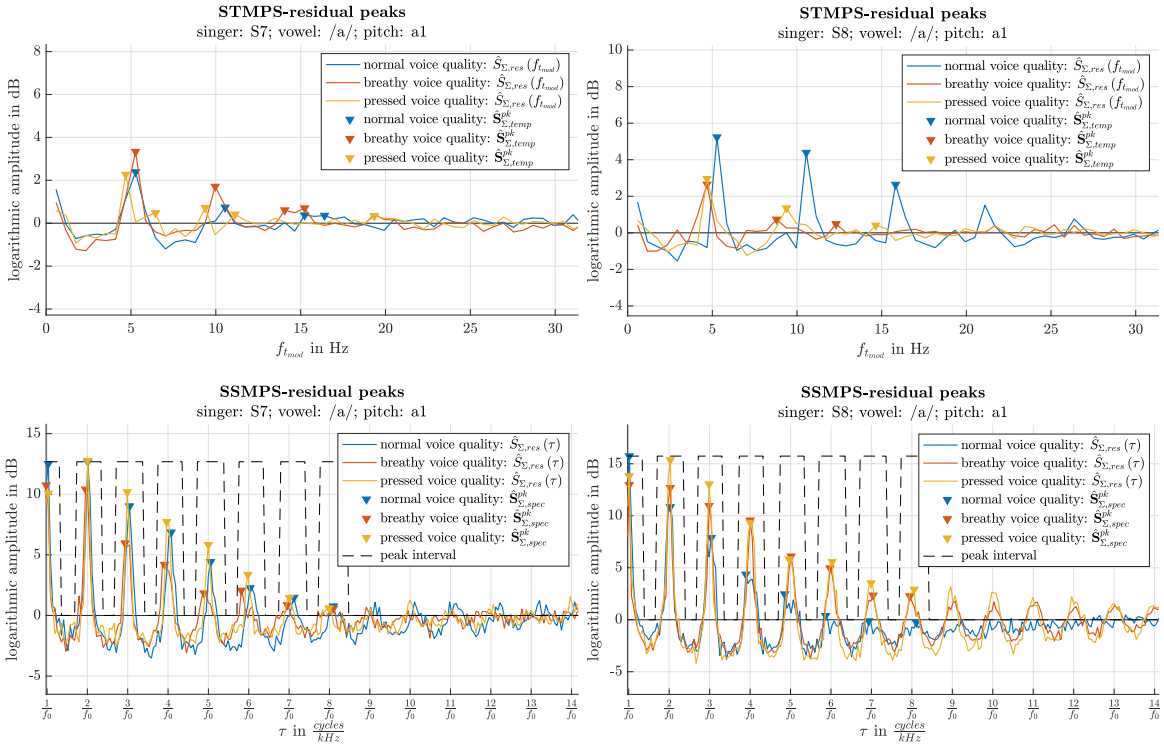


Figure B.3 $\hat{S}_{\Sigma,res}(\tau)$ *and* $\hat{S}_{\Sigma,res}(f_{t_{mod}})$*; singer: S10, vowel: /a/*



(a) $\hat{S}_{\Sigma,\text{res}}(\tau)$ and $\hat{S}_{\Sigma,\text{res}}(f_{\text{t}_{\text{mod}}})$; singer: S4, vowel: */a/* (b) $\hat{S}_{\Sigma,\text{res}}(\tau)$ and $\hat{S}_{\Sigma,\text{res}}(f_{\text{t}_{\text{mod}}})$; singer: S5, vowel: */a/*

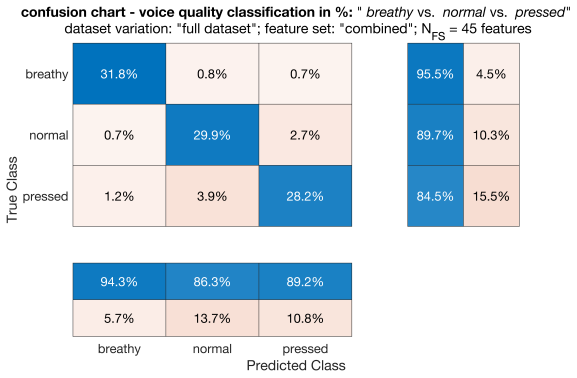
Figure B.4 *Picked peaks of STMPS and SSMPS-residual for exemplary samples.*



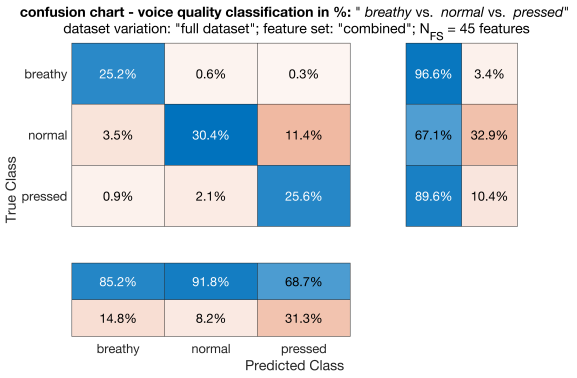
(a) $\hat{S}_{\Sigma,\text{res}}(\tau)$ and $\hat{S}_{\Sigma,\text{res}}(f_{\text{t}_{\text{mod}}})$; singer: S7, vowel: */a/* (b) $\hat{S}_{\Sigma,\text{res}}(\tau)$ and $\hat{S}_{\Sigma,\text{res}}(f_{\text{t}_{\text{mod}}})$; singer: S8, vowel: */a/*

Figure B.5 *Picked peaks of STMPS and SSMPS-residual for exemplary samples.*

B.3 Relative Confusion Matrices for Single and Two Stage SVM Classification

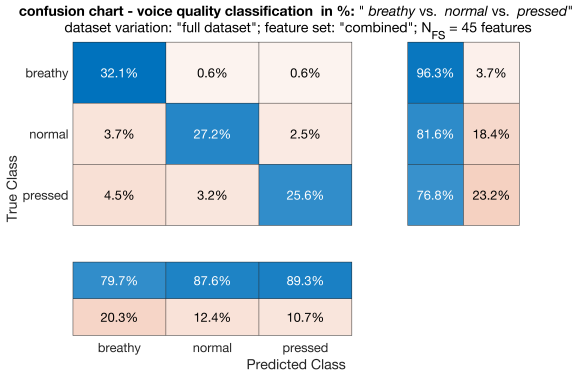


(a) full dataset with *instruction lables*; single stage SVM

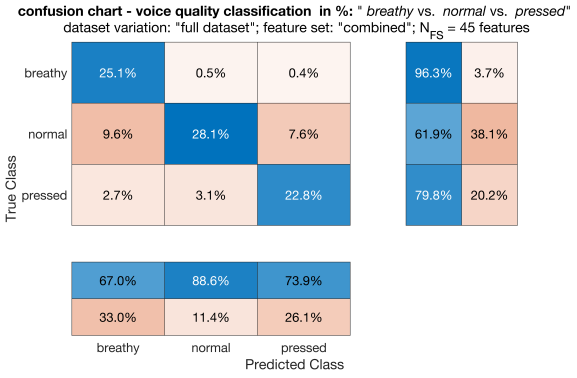


(b) full dataset with *experiment lables*; single stage SVM

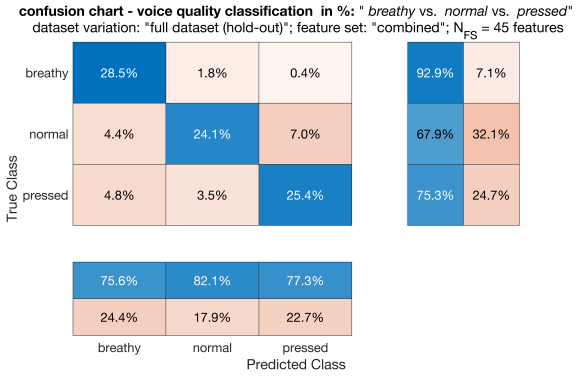
Figure B.6 *Classification results in percent of single stage SVM for (a) full dataset with instruction labels and (b) full dataset with experiment labels.*



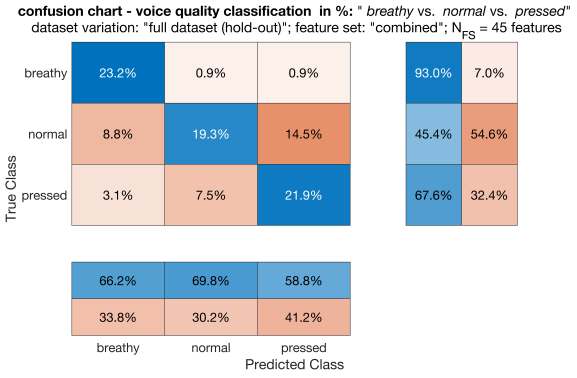
(a) full dataset with *instruction lables*; two stage SVM



(b) full dataset with *experiment lables*; two stage SVM



(c) full dataset's hold-out set with *instruction lables*; two stage SVM



(d) full dataset's hold-out set with *experiment lables*; two stage SVM

Figure B.7 *Classification results in percent of two stage SVM for (a) full dataset with inst. labels and (b) full dataset with exp. labels, (c) hold-out set classification with inst. labels in and (d) exp. labels.*

Appendix C Feature Selection Order

C.1 L-R Selection Order Tables for the Single Stage SVM

C.1.1 L-R Selection of MPS-based Feature Set

Table C.1 *L-R selection;*
***Feature set**: MPS-based*
Dataset reduction: 1^{st}

$\mathscr{X}_{\text{MPS}}$	N_{FS}							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
$\hat{\boldsymbol{S}}^{\text{pk},1}_{\Sigma,\text{temp}}$	-	-	-	-	✔	✔	✔	✔
$\boldsymbol{\Delta}_{\text{temp}}$	-	-	-	-	-	-	✔	✔
$\boldsymbol{\Sigma}^{\text{pk}}_{\text{temp}}$	-	-	-	-	-	-	-	✔
$\hat{\boldsymbol{S}}^{\text{pk},2}_{\Sigma,\text{spec}}$	✔	✔	✔	✔	✔	✔	✔	✔
$\bar{\boldsymbol{\Delta}}_{\text{spec},1}$	-	-	✔	✔	✔	✔	✔	✔
$\bar{\boldsymbol{\Delta}}_{\text{spec},2}$	-	-	-	✔	✔	✔	✔	✔
$\bar{\boldsymbol{\Delta}}_{\text{ratio}}$	-	-	-	-	-	-	-	-
$\bar{\boldsymbol{\Delta}}_{\text{overall}}$	-	✔	✔	✔	✔	✔	✔	✔
$\boldsymbol{\Sigma}^{pk}_{\text{spec}}$	-	-	-	-	-	✔	✔	✔

Table C.2 *L-R selection;*
***Feature set**: MPS-based*
Dataset reduction: 2^{nd}

$\mathscr{X}_{\text{MPS}}$	N_{FS}							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
$\hat{\boldsymbol{S}}^{\text{pk},1}_{\Sigma,\text{temp}}$	-	-	-	-	✔	✔	✔	✔
$\boldsymbol{\Delta}_{\text{temp}}$	-	-	-	-	-	-	✔	✔
$\boldsymbol{\Sigma}^{\text{pk}}_{\text{temp}}$	-	-	-	-	-	-	-	✔
$\hat{\boldsymbol{S}}^{\text{pk},2}_{\Sigma,\text{spec}}$	✔	✔	✔	✔	✔	✔	✔	✔
$\bar{\boldsymbol{\Delta}}_{\text{spec},1}$	-	-	✔	✔	✔	✔	✔	✔
$\bar{\boldsymbol{\Delta}}_{\text{spec},2}$	-	-	-	✔	✔	✔	✔	✔
$\bar{\boldsymbol{\Delta}}_{\text{ratio}}$	-	-	-	-	-	-	-	-
$\bar{\boldsymbol{\Delta}}_{\text{overall}}$	-	✔	✔	✔	✔	✔	✔	✔
$\boldsymbol{\Sigma}^{pk}_{\text{spec}}$	-	-	-	-	-	✔	✔	✔

Table C.3 *L-R selection;*
***Feature set**: MPS-based*
Dataset reduction: 3^{rd}

$\mathscr{X}_{\text{MPS}}$	N_{FS}							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
$\hat{\boldsymbol{S}}^{\text{pk},1}_{\Sigma,\text{temp}}$	-	-	-	-	✔	✔	✔	✔
$\boldsymbol{\Delta}_{\text{temp}}$	-	-	-	-	-	-	✔	✔
$\boldsymbol{\Sigma}^{\text{pk}}_{\text{temp}}$	-	-	-	-	-	-	-	✔
$\hat{\boldsymbol{S}}^{\text{pk},2}_{\Sigma,\text{spec}}$	✔	✔	✔	✔	✔	✔	✔	✔
$\bar{\boldsymbol{\Delta}}_{\text{spec},1}$	-	-	-	✔	✔	✔	✔	✔
$\bar{\boldsymbol{\Delta}}_{\text{spec},2}$	-	-	✔	✔	✔	✔	✔	✔
$\bar{\boldsymbol{\Delta}}_{\text{ratio}}$	-	-	-	-	-	-	-	-
$\bar{\boldsymbol{\Delta}}_{\text{overall}}$	-	✔	✔	✔	✔	✔	✔	✔
$\boldsymbol{\Sigma}^{pk}_{\text{spec}}$	-	-	-	-	-	✔	✔	✔

C.1.2 L-R Selection of MFCC Feature Set

Table C.4 *L-R selection;* ***Feature set****: MFCCs;* ***Dataset reduction:*** 1^{st}

$\mathscr{X}_{\text{MFCC}}$	N_{FS}																																
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**
$\tilde{c}_1$	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_2$	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_3$	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_4$	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_5$	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_6$	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_7$	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_8$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_9$	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{10}$	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{11}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{12}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{13}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{14}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{15}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔
$\tilde{c}_{16}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{17}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{18}$	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{19}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{20}$	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{21}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{22}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{23}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{24}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔
$\tilde{c}_{25}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{26}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔
$\tilde{c}_{27}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{28}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔
$\tilde{c}_{29}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔
$\tilde{c}_{30}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
$\tilde{c}_{31}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔
$\tilde{c}_{32}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{33}$	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{34}$	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{35}$	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

Table C.5 *L-R selection;* ***Feature set****: MFCCs;* ***Dataset reduction:*** 2^{nd}

$\mathscr{X}_{\text{MFCC}}$	N_{FS}																																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
$\tilde{c}_1$	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_2$	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_3$	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_4$	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_5$	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_6$	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_7$	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_8$	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_9$	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{10}$	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{11}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{12}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{13}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{14}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{15}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔
$\tilde{c}_{16}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{17}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{18}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔
$\tilde{c}_{19}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{20}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{21}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔
$\tilde{c}_{22}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{23}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{24}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{25}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{26}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔
$\tilde{c}_{27}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔
$\tilde{c}_{28}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{29}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔
$\tilde{c}_{30}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
$\tilde{c}_{31}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{32}$	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{33}$	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{34}$	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{35}$	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

Table C.6 *L-R selection;* ***Feature set****: MFCCs;* ***Dataset reduction:*** 3^{rd}

$\mathscr{X}_{\text{MFCC}}$	N_{FS}																																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
$\tilde{c}_1$	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_2$	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_3$	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_4$	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_5$	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_6$	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_7$	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_8$	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_9$	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{10}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{11}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔
$\tilde{c}_{12}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{13}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔
$\tilde{c}_{14}$	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{15}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{16}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔
$\tilde{c}_{17}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{18}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{19}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{20}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{21}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{22}$	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{23}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{24}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔
$\tilde{c}_{25}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{26}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{27}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔
$\tilde{c}_{28}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔
$\tilde{c}_{29}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{30}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
$\tilde{c}_{31}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔
$\tilde{c}_{32}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{33}$	-	-	-	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{34}$	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
$\tilde{c}_{35}$	-	-	-	-	-	-	-	-	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

C.1.3 L-R Selection of Combined Feature Set

Table C.7 *L-R selection;* ***Feature set****: combined;* ***Dataset reduction:*** 1^{st}

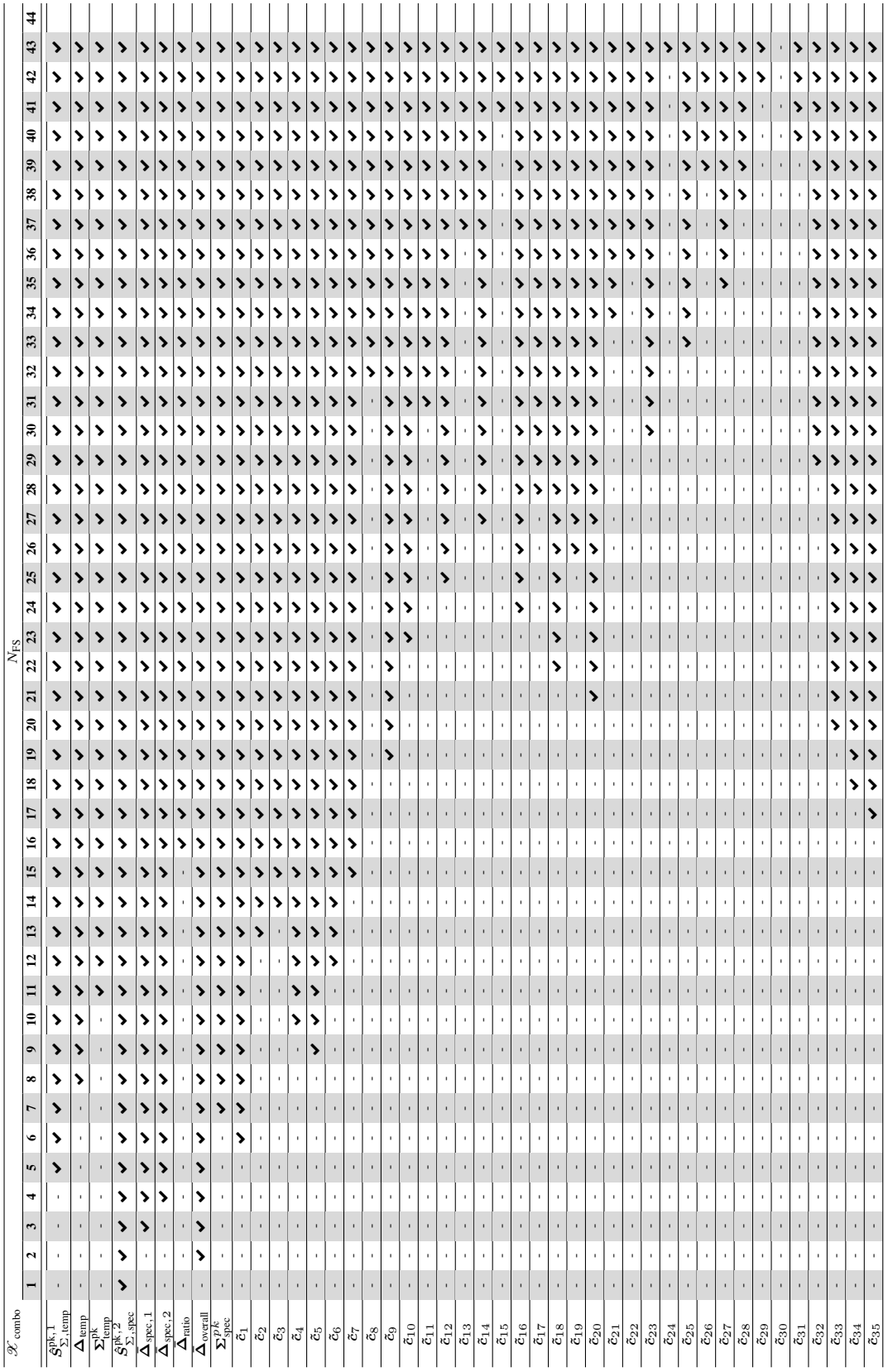


Table C.8 *L-R selection;* ***Feature set****: combined;* ***Dataset reduction:*** 2^{nd}

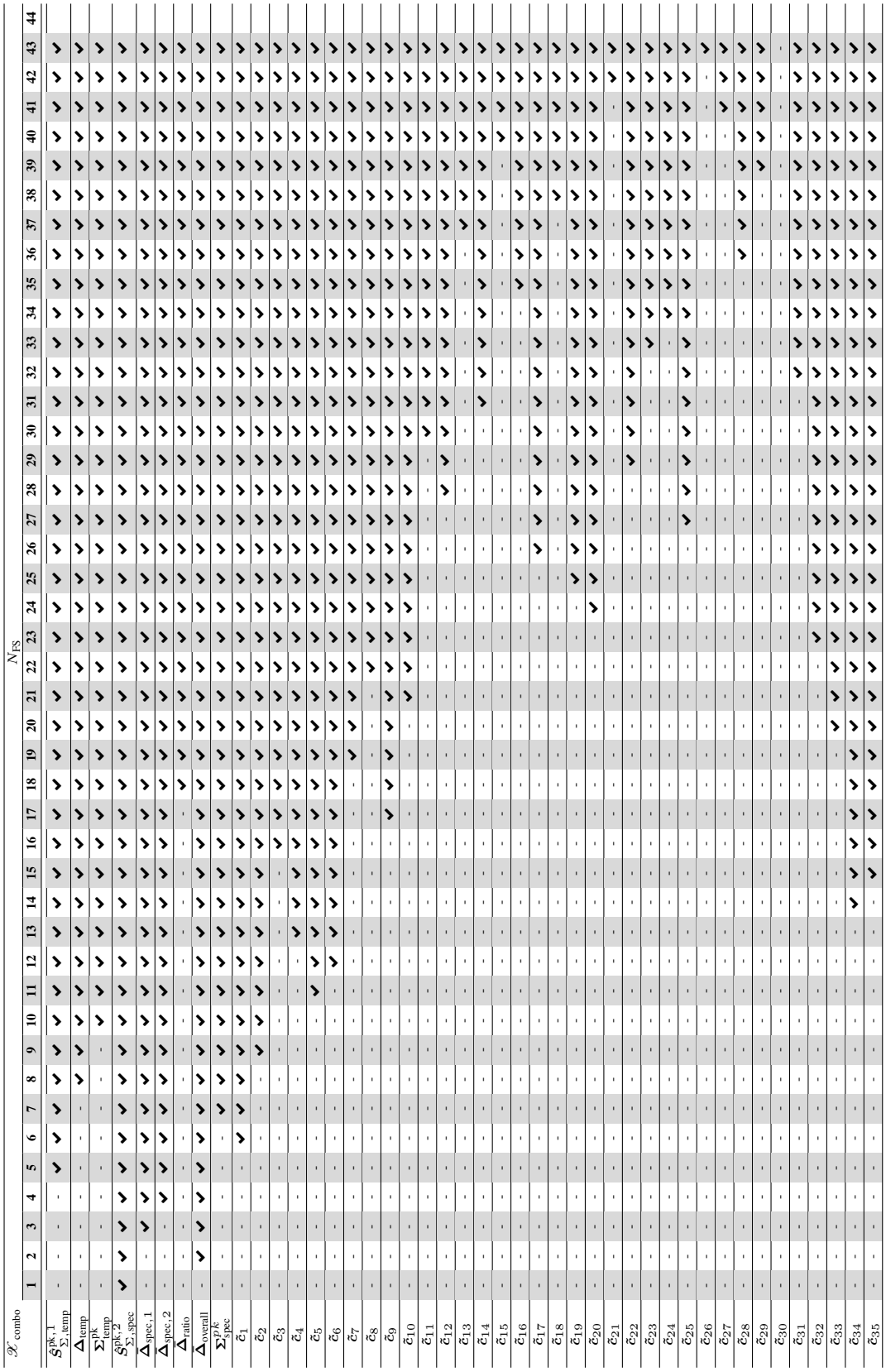
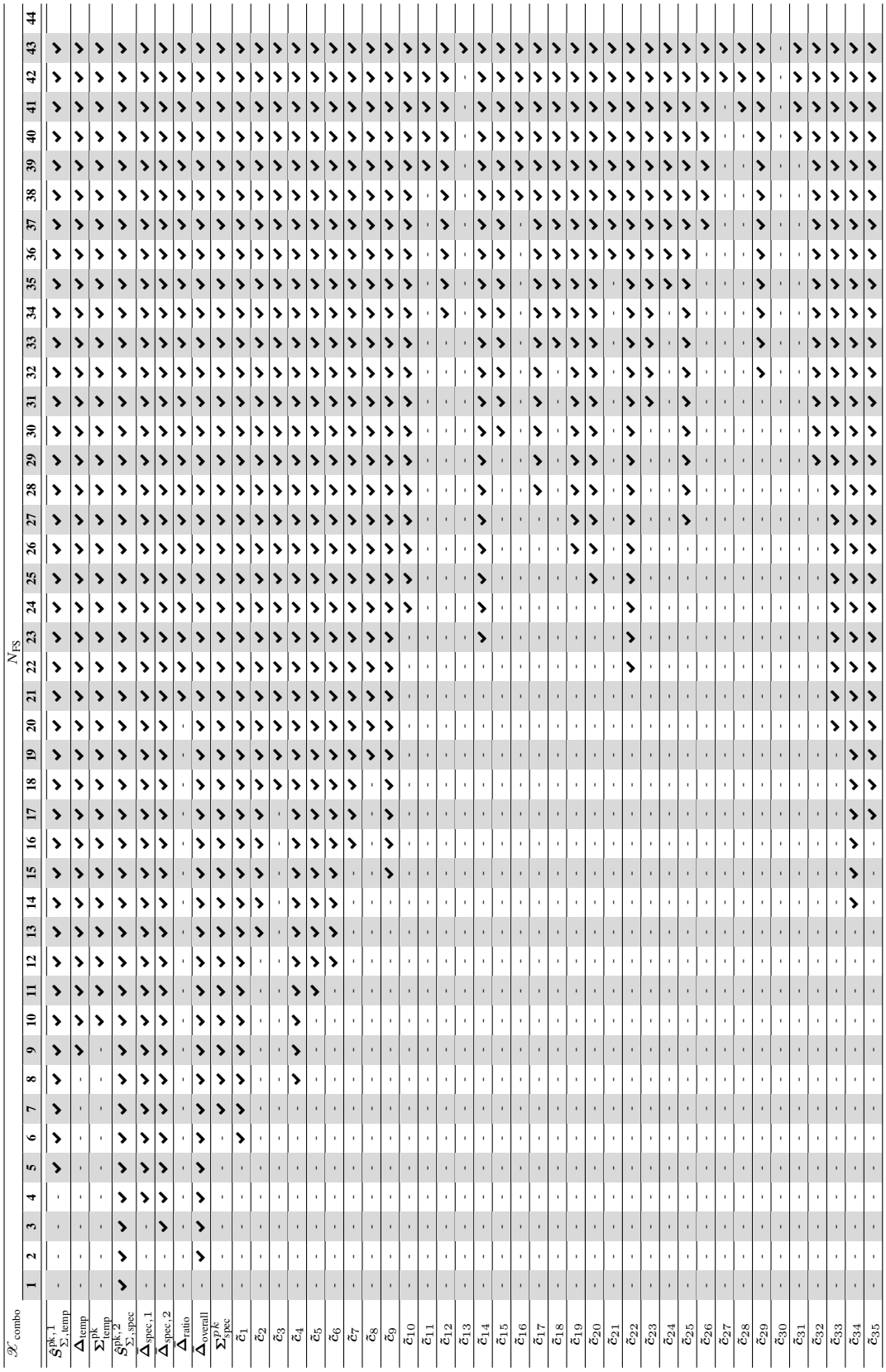


Table C.9 *L-R selection;* ***Feature set****: combined;* ***Dataset reduction:*** 3^{rd}



C.2 L-R Selection Order Tables for the Two Stage SVM

Table C.10 *L-R selection;* ***Feature set****: combined;* ***Dataset reduction:*** 1^{st}*;* ***SVM stage:*** 1^{st}

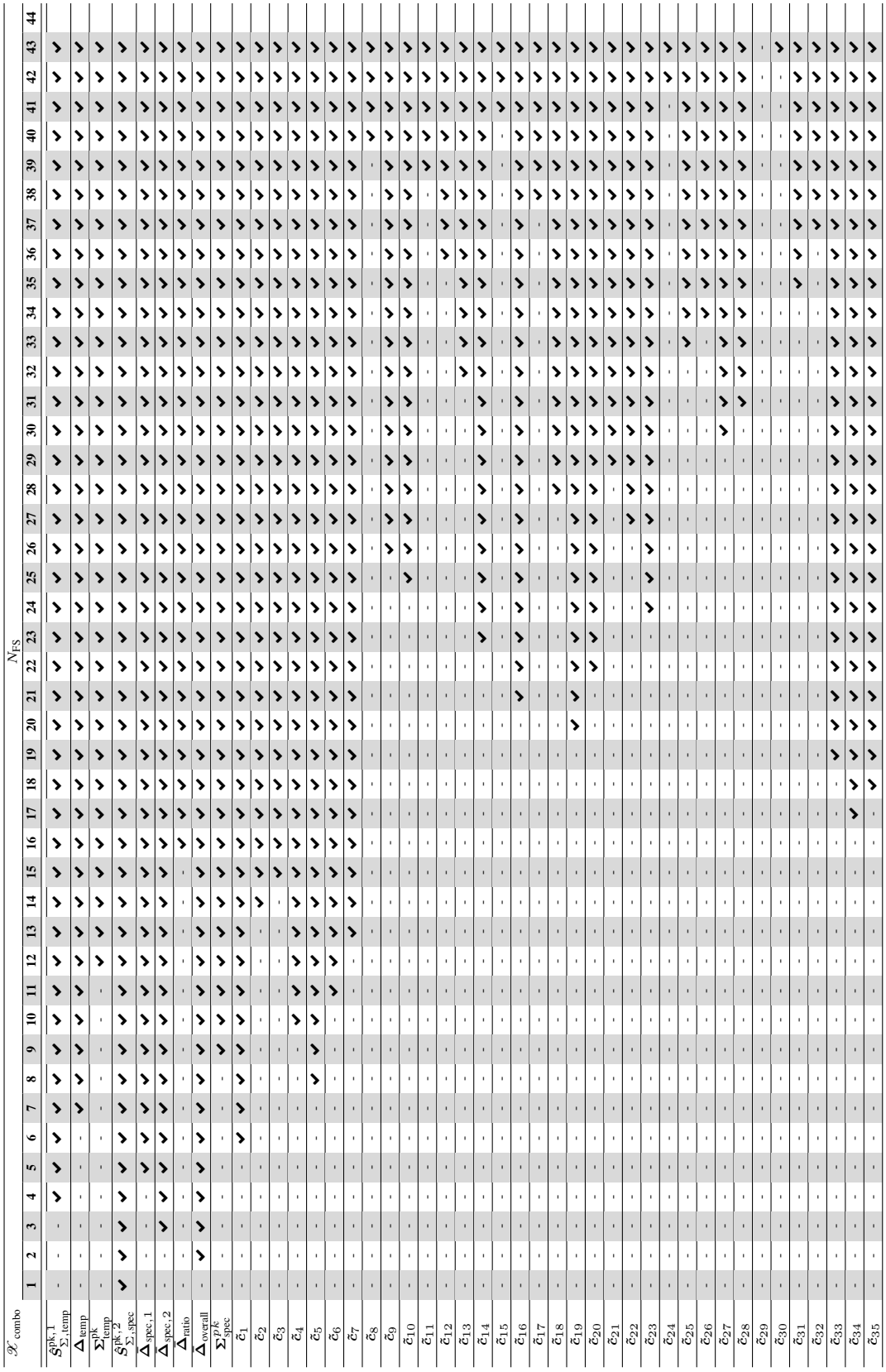


Table C.11 *L-R selection;* ***Feature set****: combined;* ***Dataset reduction:*** *1st;* ***SVM stage:*** *2nd*

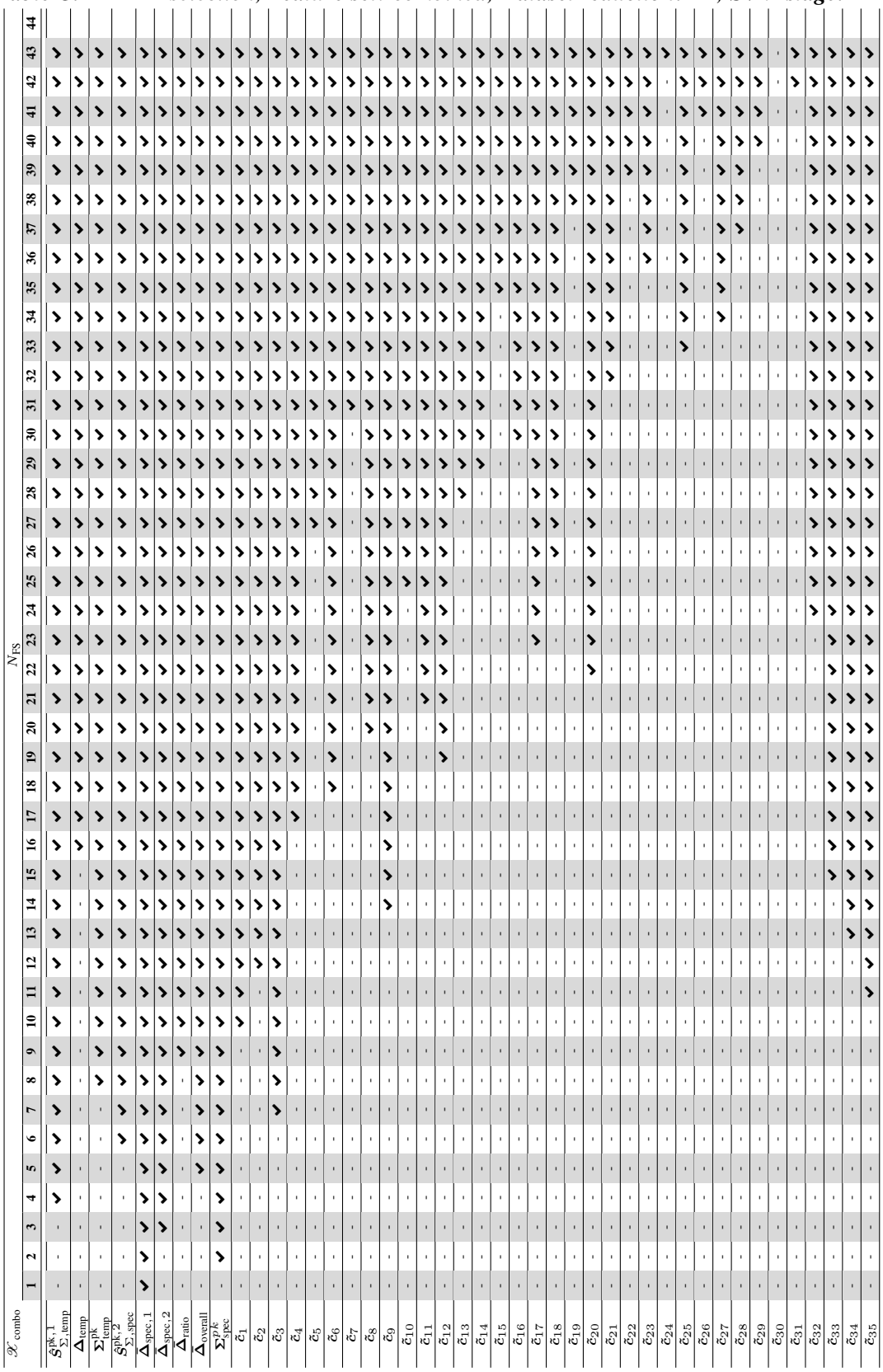


Table C.12 *L-R selection;* ***Feature set****: combined;* ***Dataset reduction:*** 2^{nd}*;* ***SVM stage:*** 1^{st}

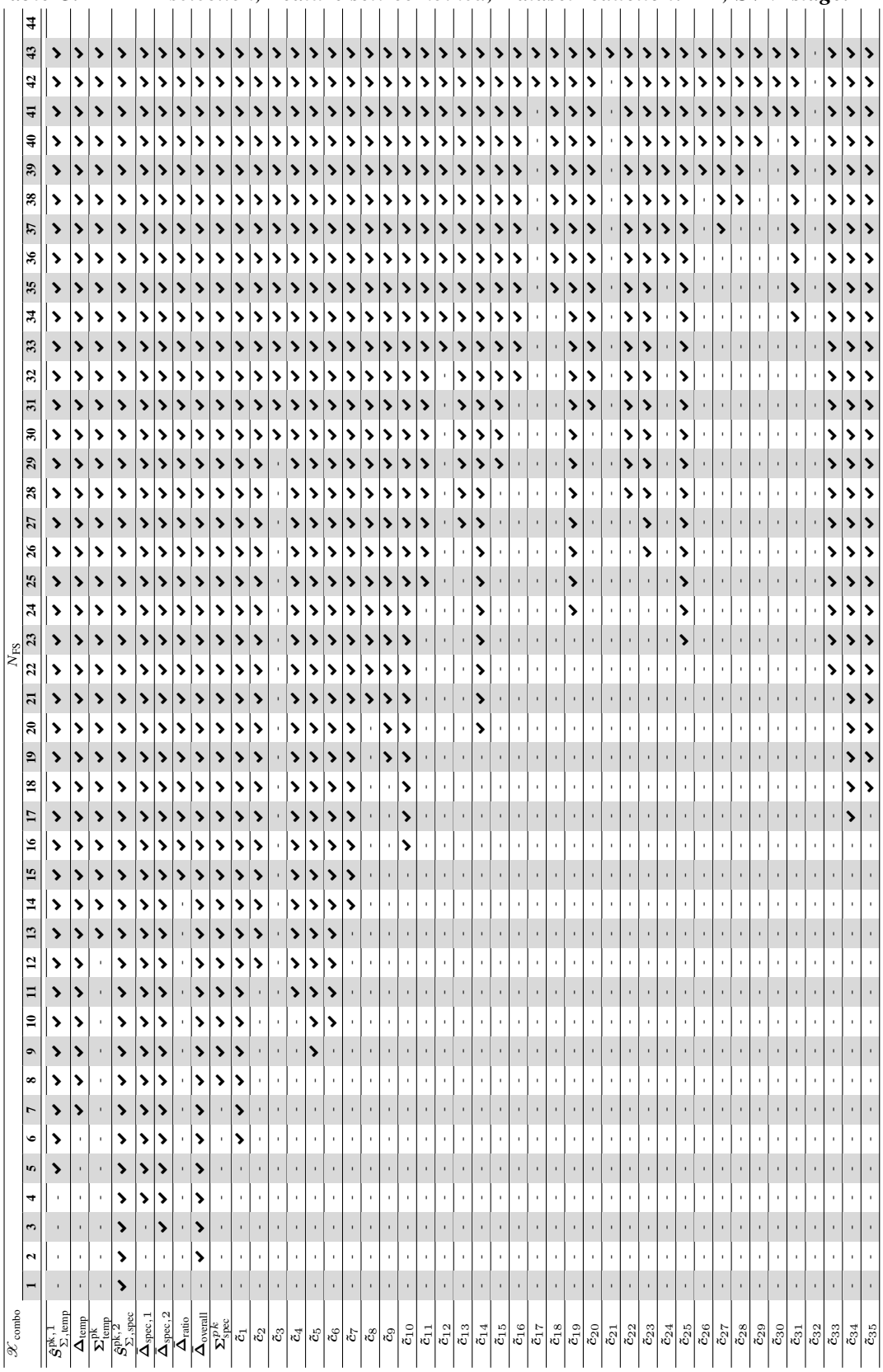


Table C.13 *L-R selection;* ***Feature set****: combined;* ***Dataset reduction:*** 2^{nd}*;* ***SVM stage:*** 2^{nd}

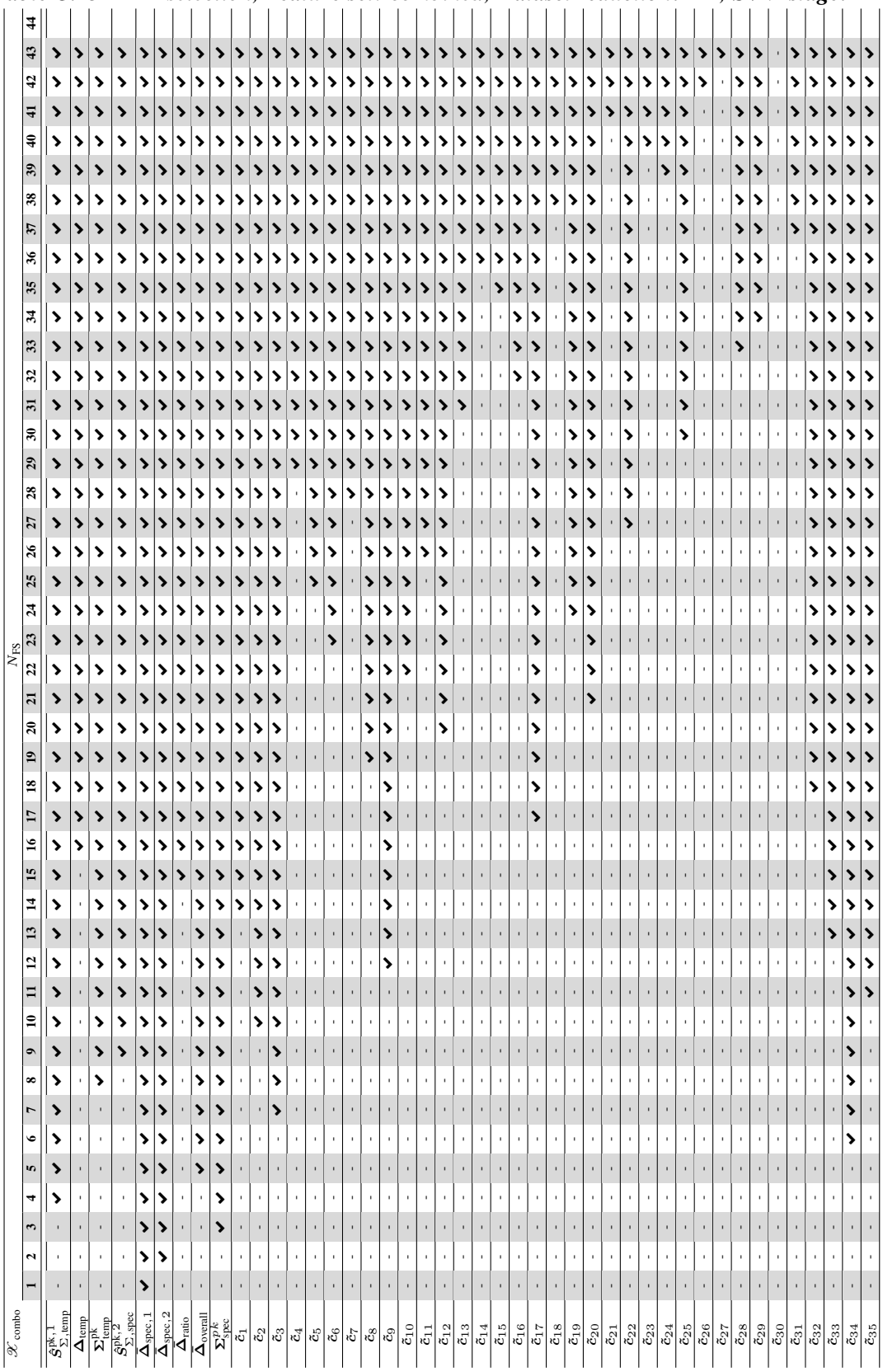


Table C.14 *L-R selection;* ***Feature set****: combined;* ***Dataset reduction:*** 3^{rd}*;* ***SVM stage:*** 1^{st}

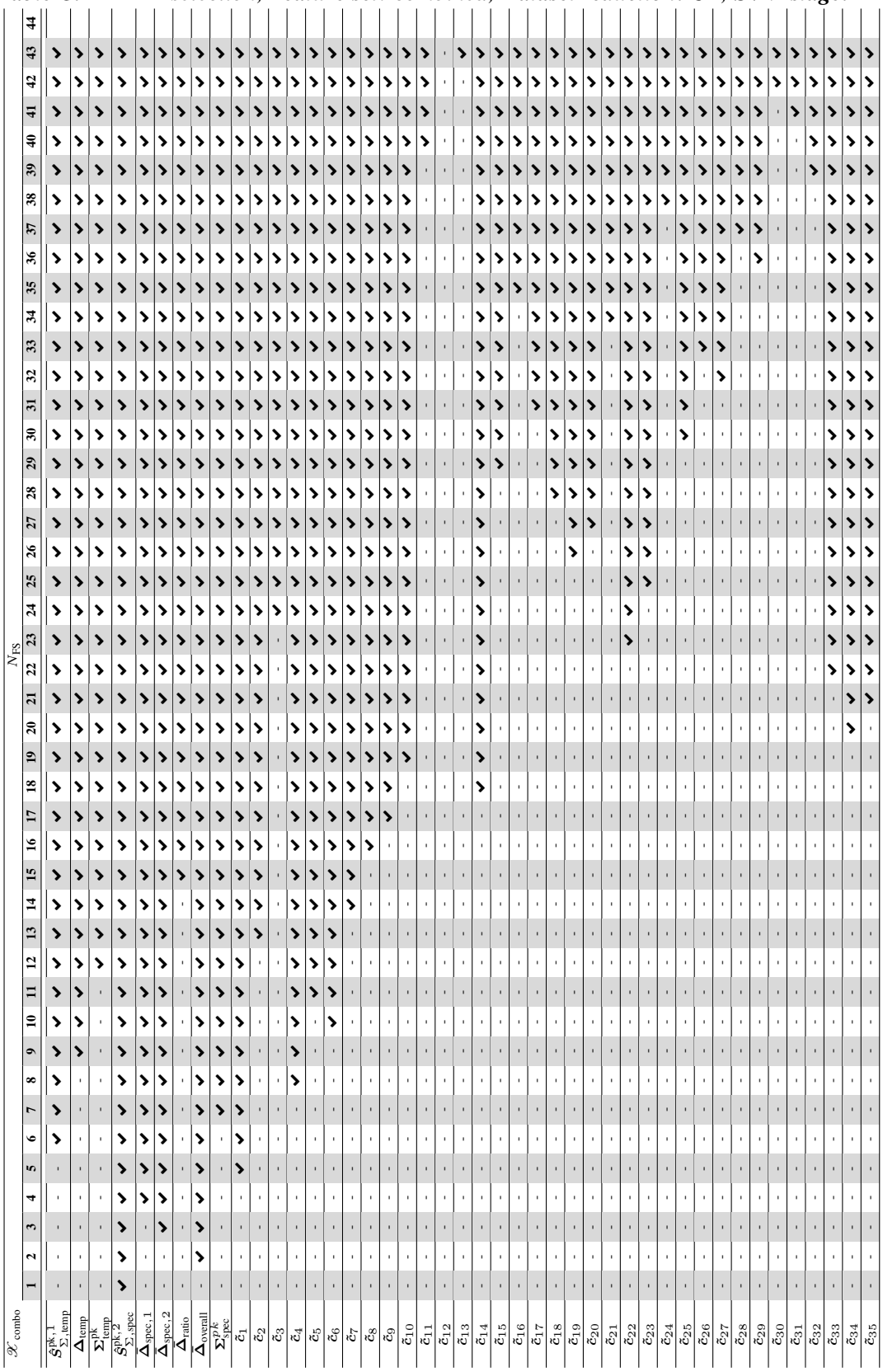
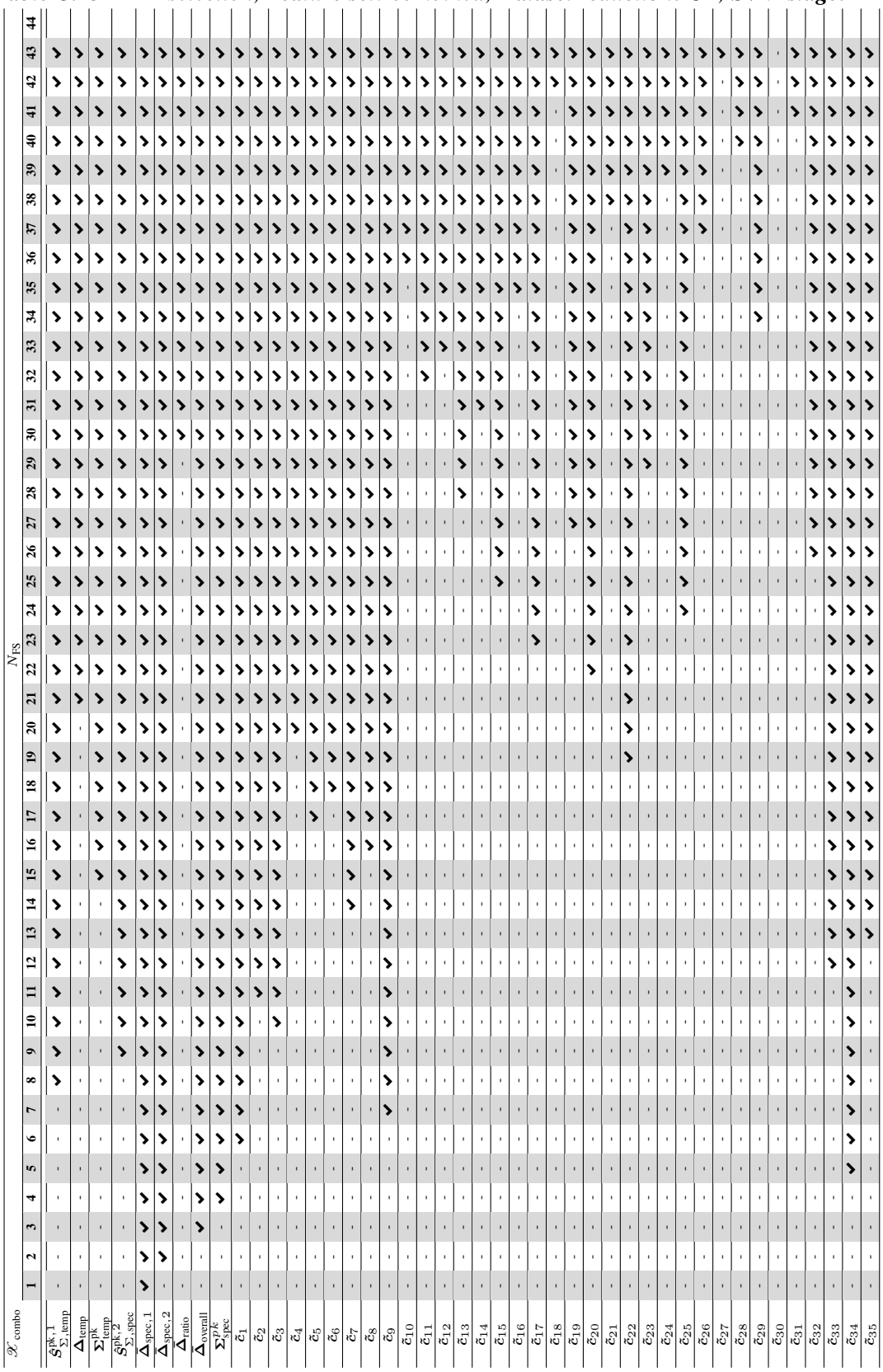


Table C.15 *L-R selection;* ***Feature set****: combined;* ***Dataset reduction:*** 3^{rd}*;* ***SVM stage:*** 2^{nd}



Bibliography

[1] Paavo Alku. Glottal wave analysis with pitch synchronous iterative adaptive inverse filtering. *Speech Communication*, 11(2):109–118, 1992. Eurospeech '91.

[2] Paavo Alku, Tiina Murtola, Jarmo Malinen, Juha Kuortti, Brad Story, Manu Airaksinen, Mika Salmi, Erkki Vilkman, and Ahmed Geneid. OPENGLOT - an open environment for the evaluation of glottal inverse filtering. *Speech Communication*, 107:38 – 47, 2019.

[3] UCLA Phonetics Lab Archive. International phonetic alpha, 2003. [accessed 28.11.21], `https://www.ipachart.com`.

[4] Paul A. Bereuter and Florian Kraxberger. Synthesis and linear prediction analysis of sung vocal signals. Audio engineering project, 2019.

[5] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, Berlin, Heidelberg, 2006.

[6] Manuel Brandner, Matthias Frank, and Daniel Rudrich. Dirpat—database and viewer of 2d/3d directivity patterns of sound sources and receivers. In *Audio Engineering Society Convention 144*, May 2018.

[7] Yu-Ren Chien, Daryush D. Mehta, Jón Guðnason, Matías Zañartu, and Thomas F. Quatieri. EGIFA - Evaluation of Glottal Inverse Filtering Algorithms Using a Physiologically Based Articulatory Speech Synthesizer. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(8):1718–1730, Aug 2017.

[8] Orchisama Das, Julius Orion Smith, and Chris Chafe. Real-time pitch tracking in audio signals with the extended complex kalman filter. 2017.

[9] Thomas Drugman and Abeer Alwan. Joint robust voicing detection and pitch estimation based on residual harmonics. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 2011*, pages 1973–1976, Florence, Italy, January 2011.

[10] Michael Döllinger and Manfred Kaltenbacher. Preface: Recent Advances in Understanding the Human Phonatory Process. *Acta Acustica united with Acustica*, 102(14):195–208, February 2016.

[11] Daniel P. W. Ellis. PLP and RASTA (and MFCC, and inversion) in Matlab, 2005. [accessed 06.12.21] , `http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/`.

[12] Gunnar Fant, Johan Liljencrants, and Qiguang Lin. A Four-Parameter Model of Glottal Flow. In *Quarterly Progress and Status Report*, volume 26(4) of *STL-QPSR*, pages 1–13. KTH School of Computer Science and Communication, Dept. for Speech, Music and Hearing, 1985.

[13] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, B. Krawczyk, and Francisco Herrera. Learning from imbalanced data sets. In *Springer International Publishing*, 2018.

[14] Peter-Micheal Fischer. *Stimmgattungen und Stimmtypen*, pages 117–137. J.B. Metzler, Stuttgart, 1998.

[15] Christer Gobl. A preliminary study of acoustic voice quality correlates. In *Quarterly Progress and Status Report*, volume 30(4) of *STL-QPSR*, pages 9–22. KTH School of Computer Science and Communication, Dept. for Speech, Music and Hearing, 1989.

[16] Navdeep Jaitly and Geoffrey E. Hinton. Vocal tract length perturbation (vtlp) improves speech recognition. In *In International Conference on Machine Learning (ICML)*, page 2013.

[17] Keith Johnson. The δf method of vocal tract length normalization for vowels. 2020.

[18] Sudarsana Reddy Kadiri and Paavo Alku. Mel-frequency cepstral coefficients derived using the zero-time windowing spectrum for classification of phonation types in singing. *Journal of the Acoustical Society of America*, Volume 146, issue 5:EL418–EL423, 2019.

[19] Sudarsana Reddy Kadiri and Bayya Yegnanarayana. Analysis and Detection of Phonation Modes in Singing Voice using Excitation Source Features and Single Frequency Filtering Cepstral Coefficients (SFFCC). In *Proc. Interspeech 2018*, pages 441–445, 2018.

[20] Hemant Kumar Kathania, S. Shahnawazuddin, Waquar Ahmad, and Nagaraj Adiga. Role of linear, mel and inverse-mel filterbanks in automatic recognition of speech from high-pitched speakers. *Circuits, Systems, and Signal Processing*, 38(10):4667–4682, 2019.

[21] L. Kocher. Evaluation of a proper measurement environment to determine directivity characteristics of the singing voice. Audio engineering project, 2019.

[22] Branko Kovacevic, Milan Milosavljević, Mladen Veinović, and Milan Markovic. *Robust Digital Processing of Speech Signals*. Springer International Publishing, June 2017.

[23] L. Lee and R. Rose. A frequency warping approach to speaker normalization. *IEEE Transactions on Speech and Audio Processing*, 6(1):49–60, 1998.

[24] Hui-Ling Lu and Julius Orion Smith. Glottal source modeling for singing voice synthesis. CCRMA, Stanford University, 01 2000.

[25] Maria Markaki and Yannis Stylianou. Using modulation spectra for voice pathology detection and classification. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, 2009:2514–7, 2009.

[26] Maria Markaki and Yannis Stylianou. Discrimination of speech from nonspeeech in broadcast news based on modulation frequency features. *Speech Communication*, 53(5):726–735, 2011. Perceptual and Statistical Audition.

[27] MathWorks. Matlab documentation: `confusionchart()`. Software documentation, The Mathworks Inc., 2021. [accessed 28.11.21] , `https://de.mathworks.com/help/stats/confusionchart.html`.

[28] MathWorks. Matlab documentation: `crossval()`. Software documentation, The Mathworks Inc., 2021. [accessed 05.12.21] , `https://de.mathworks.com/help/stats/classificationsvm.crossval.html?searchHighlight=crossval&s_tid=srchtitle_crossval_2`.

[29] MathWorks. Matlab documentation: `eig()`. Software documentation, The Mathworks Inc., 2021. [accessed 19.12.21] , `https://de.mathworks.com/help/matlab/ref/eig.html`.

[30] MathWorks. Matlab documentation: `findpeaks()`. Software documentation, The Mathworks Inc., 2021. [accessed 09.12.21] , `https://de.mathworks.com/help/signal/ref/findpeaks.html`.

[31] MathWorks. Matlab documentation: `fitcecoc()`. Software documentation, The Mathworks Inc., 2021. [accessed 01.12.21] , `https://de.mathworks.com/help/stats/fitcecoc.html`.

[32] MathWorks. Matlab documentation: `interp()`. Software documentation, The Mathworks Inc., 2021. [accessed 07.12.21] , `https://de.mathworks.com/help/signal/ref/interp.html?searchHighlight=interp&s_tid=srchtitle_interp_1`.

[33] MathWorks. Matlab documentation: `kfoldLoss()`. Software documentation, The Mathworks Inc., 2021. [accessed 05.12.21] , `https://de.mathworks.com/help/stats/classreg.learning.partition.classificationpartitionedmodel.kfoldloss.html#mw_9b5b558b-092c-4d1e-8eea-47a3390503a8`.

[34] MathWorks. Matlab documentation: `kmediods()`. Software documentation, The Mathworks Inc., 2021. [accessed 28.11.21] , `https://de.mathworks.com/help/stats/kmedoids.html`.

[35] MathWorks. Matlab documentation: `mean()`. Software documentation, The Mathworks Inc., 2021. [accessed 07.12.21] , `https://de.mathworks.com/help/matlab/ref/mean.html`.

[36] MathWorks. Matlab documentation: `mldivide`. Software documentation, The Mathworks Inc., 2021. [accessed 11.1.21] , `https://de.mathworks.com/help/matlab/ref/mldivide.html?searchHighlight=backslash&s_tid=srchtitle_backslash_1`.

[37] MathWorks. Matlab documentation: `polyfit()`. Software documentation, The Mathworks Inc., 2021. [accessed 09.12.21] , `https://de.mathworks.com/help/matlab/ref/polyfit.html`.

[38] MathWorks. Matlab documentation: `predict()`. Software documentation, The Mathworks Inc., 2021. [accessed 05.12.21] , `https://de.mathworks.com/help/stats/classreg.learning.classif.compactclassificationsvm.predict.html?searchHighlight=predict&s_tid=srchtitle_predict_4`.

[39] MathWorks. Matlab documentation: `rand()`. Software documentation, The Mathworks Inc., 2021. [accessed 07.12.21] , `https://de.mathworks.com/help/matlab/ref/rand.html`.

[40] MathWorks. Matlab documentation: `sort()`. Software documentation, The Mathworks Inc., 2021. [accessed 11.1.21] , `https://de.mathworks.com/help/matlab/ref/sort.html`.

[41] MathWorks. Matlab documentation: `std()`. Software documentation, The Mathworks Inc., 2021. [accessed 07.12.21] , `https://de.mathworks.com/help/matlab/ref/std.html?searchHighlight=std&s_tid=srchtitle_std_1`.

[42] MathWorks. Matlab documentation: `templateSVM()`. Software documentation, The Mathworks Inc., 2021. [accessed 02.12.21] , `https://www.mathworks.com/help/stats/templatesvm.html`.

[43] MathWorks. Matlab documentation: `zscore()`. Software documentation, The Mathworks Inc., 2021. [accessed 16.12.21] , `https://de.mathworks.com/help/stats/zscore.html`.

[44] Thomas Mayr. Klangtransformationen auf basis des modulation power spectrums. Master's thesis, 2017. [acessed on 23.12.21], `http://phaidra.kug.ac.at/o:43181`.

[45] Brian B. Monson, Eric J. Hunter, Andrew J. Lotto, and Brad H. Story. The perceptual significance of high-frequency energy in the human voice. *Frontiers in Psychology*, 5:587, 2014.

[46] Feiping Nie, Shiming Xiang, Yangqing Jia, Changshui Zhang, and Shuicheng Yan. Trace ratio criterion for feature selection. volume 2, pages 671–676, 01 2008.

[47] Douglas O'Shaughnessy. *Automatic Speech Recognition*, pages 367–435. 2000.

[48] Ranjit Panigrahi and Samarjeet Borah. 1 - classification and analysis of facebook metrics dataset using supervised classifiers. In Nilanjan Dey, Samarjeet Borah, Rosalina Babo, and Amira S. Ashour, editors, *Social Network Analytics*, pages 1–19. Academic Press, 2019.

[49] Kaare B. Petersen and Michael S. Pedersen. The matrix cookbook, nov 2012. Version 20121115.

[50] Ville Pulkki and matti karjalainen. *Communication Acoustics: An Introduction to Speech, Audio and Psychoacoustics*. 01 2015.

[51] M. Shahidur Rahman and Tetsuya Shimamura. Linear prediction using refined autocorrelation function. *EURASIP Journal on Audio, Speech, and Music Processing*, 2007(1):045962, 2007.

[52] K. Sreenivasa Rao and K. E. Manjunath. *Speech Recognition Using Articulatory and Excitation Source Features*. Springer Publishing Company, Incorporated, 1st edition, 2017.

[53] Jean-Luc Rouas and Leonidas Ioannidis. Automatic Classification of Phonation Modes in Singing Voice: Towards Singing Style Characterisation and Application to Ethnomusicological Recordings. In *Proc. Interspeech 2016*, pages 150–154, 2016.

[54] Peter Sciri. Singing Voice Vibrato: Measurement and Modification. Master's thesis, Institute of Electronic Music and Acoustics, University of Music and Performing Arts, Graz, June 2011.

[55] Eberhard Sengpiel. Klaviatur und frequenzen, 2021. [accessed 09.12.21], `http://www.sengpielaudio.com/Rechner-notennamen.htm`.

[56] Nandini C. Singh and Frédéric E. Theunissen. Modulation spectra of natural sounds and ethological theories of auditory processing. *The Journal of the Acoustical Society of America*, 114(6):3394–3411, 2003.

[57] Johan Sundberg. *The Science of the Singing Voice*. Northern Illinois University Press, 1987.

[58] Johan Sundberg. 6 - the perception of singing. In Diana Deutsch, editor, *The Psychology of Music (Second Edition)*, Cognition and Perception, pages 171–214. Academic Press, San Diego, second edition edition, 1999.

[59] Johan Sundberg. 3 - perception of singing. In Diana Deutsch, editor, *The Psychology of Music (Third*

Edition), pages 69–105. Academic Press, third edition edition, 2013.

[60] Johan Sundberg. Objective characterization of phonation type using amplitude of flow glottogram pulse and of voice source fundamental. *Journal of Voice*, 2020.

[61] B Venkatesh and J. Anuradha. A review of feature selection and its methods. *Cybernetics and Information Technologies*, 19:3, 03 2019.

[62] Rebecca Vos, Jamie A. S. Angus, and Brad H. Story. A new algorithm for vocal tract shape extraction from singer's waveforms. In *Audio Engineering Society Convention 136*, Apr 2014.

[63] Andrew R Webb. *Statistical Pattern Recognition*. Wiley, New York, 2. edition edition, 2003.

[64] Kamil Wojcicki. Htk mfcc matlab. MATLAB Central File Exchange, 2021. [accessed, 06.12.21], `https://www.mathworks.com/matlabcentral/fileexchange/32849-htk-mfcc-matlab`.

[65] Richard Wright, Courtney Mansfield, and Laura Panfili. Voice quality types and uses in north american english. *Anglophonia*, 27, 11 2019.

[66] Steve J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book Version 3.4*. Cambridge University Press, 2006.

[67] Julia Ziegerhofer. Excitation Signal Analysis - Gender Differences. Master's thesis, Signal Processing and Speech Communication Lab, Graz University of Technology, Graz, March 2018.