Improving Supervised Music Classification by Means of Multi-Objective Evolutionary Feature Selection

Dissertation

zur Erlangung des Grades eines

D o k t o r s d e r N a t u r w i s s e n s c h a f t e n

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

Igor Vatolkin

Dortmund

2013

Tag der mündlichen Prüfung : 17.04.2013

Dekan : Prof. Dr.-Ing. Gernot Fink

Gutachter : Prof. Dr. Günter Rudolph
: Prof. Dr. Claus Weihs

Contents

Abstract

In this work, several strategies are developed to reduce the impact of the two limitations of most current studies in supervised music classification: the classification rules and music features have often a low interpretability, and the evaluation of algorithms and feature subsets is almost always done with respect to only one or a few common evaluation criteria separately.

Although music classification is in most cases user-centered and it is desired to understand well the properties of related music categories, many current approaches are based on low-level characteristics of the audio signal. We have designed a large set of more meaningful and interpretable high-level features, which may completely replace the baseline low-level feature set and are even capable to significantly outperform it for the categorisation into three music styles. These features provide a comprehensible insight into the properties of music genres and styles: instrumentation, moods, harmony, temporal, and melodic characteristics. A crucial advantage of audio high-level features is that they can be extracted from any digitally available music piece, independently of its popularity, availability of the corresponding score, or the Internet connection for the download of the metadata and community features, which are sometimes erroneous and incomplete. A part of high-level features, which are particularly successful for classification into genres and styles, has been developed based on the novel approach called sliding feature selection. Here, high-level features are estimated from low-level and other high-level ones during a sequence of supervised classification steps, and an integrated evolutionary feature selection helps to search for the most relevant features in each step of this sequence.

Another drawback of many related state-of-the-art studies is that the algorithms and feature sets are almost always compared using only one or a few evaluation criteria separately. However, different evaluation criteria are often in conflict: an algorithm optimised only with respect to classification quality may be slow, have high storage demands, perform worse on imbalanced data, or require high user efforts for labelling of songs. The simultaneous optimisation of multiple conflicting criteria remains until now almost unexplored in music information retrieval, and it was applied for feature selection in music classification for the first time in this thesis, except for several preliminary own publications. As an exemplarily multi-objective approach for optimisation of feature selection, we simultaneously minimise the classification error and the number of features used for classification. The sets with more features lead to a higher classification quality. On the other side, the sets with fewer features and a lower classification performance may help to strongly decrease the demands for storage and computing time and to reduce the risk of too complex and overfitted classification models. Further, we describe several groups of evaluation criteria and discuss other reasonable multi-objective optimisation scenarios for music data analysis.

Acknowledgements

This work would not have been possible without the support of many colleagues.

First of all, I would like to thank both supervisors of the thesis, Prof. Günter Rudolph and Prof. Claus Weihs. In my opinion, they allowed me the perfect balance for my work and helped to carry out many of my own ideas. The supervisors always had enough time for discussions, encouraged me to present my results at conferences, and gave me an opportunity to participate in the organisation and supervision of student seminars, a project group and several master theses.

A solid basis for any classification task is the feature set. I would like to thank Markus Eichhoff, Dr. Antti Eronen, Dr. Olivier Lartillot, Dr. Matthias Mauch, Dr. Ingo Mierswa, Prof. Meinard Müller, Anil Nagathil and Dr. Wolfgang Theimer for their help on the integration of features from different software toolboxes.

Our student assistants, Daniel Stoller and Clemens Wältken, contributed to the implementation of AMUSE. Thank you!

The first steps in music classification and many fruitful discussions were carried out during my cooperation with the Nokia Research Center, Bochum. During that time I had a great support by Prof. Martin Botteck (now FH Südwestfalen, Meschede) and Dr. Wolfgang Theimer (now BlackBerry, Bochum).

I am grateful to several colleagues, which supported me in certain aspects of the work. Several of discussions led to joint publications. I would like to thank Bernd Bischl for the introduction into the basics of statistical testing and proper evaluation, Prof. Dietmar Jannach for discussions on music recommendation, Markus Eichhoff for the cooperation on instrument identification, Mike Preuß for discussions on the evolutionary algorithms, and Prof. Günther Rötter for the design of many high-level features and the discussions related to music science.

Especially encouraging talks and discussions were held during the regular meetings of the "Special Interest Group on Music Analysis" (SIGMA)[1].

At the beginning of the work on this thesis, I was financially supported through a joint project with the Nokia Research Center (MusicDescriber). Most studies were carried out during the project "Multicriteria Optimization of Automatic Music Classification based on High Level Features with Methods of Computational Intelligence", supported by the Klaus Tschira Foundation.

Finally, I would like to thank God for the creation of the world with a place for both music and computer science, my wife Nadine for her love and support, and Tim and Marie for being the most important motivation of my life.

[1] http://sig-ma.de/, date of visit: 15.02.2013.

to the memory of my mother

1. Introduction

1.1. Motivation and scope

Supervised music classification is one of the most frequent applications in music information retrieval (MIR). It allows the categorisation into genres and personal music preferences, the recognition of instruments, harmony and temporal characteristics, music recommendation, management of large music collections, etc. The basic principle of this approach is to create the classification models from the ground truth: the vectors of numerical music characteristics, or features, and the corresponding labels.

For a fully automatic classification system, it is preferable to start with a large, once created feature set. Then, the definition of each new category would require only the assignment of new labels to music data. However, a large number of features, which are irrelevant for a particular classification task, often overwhelms the classification methods. Features, which are by chance recognised as significant for the training data, become part of the classification models. This leads to a suffering of model generalisation performance on the unseen data. In that case, feature selection allows the effective computation of the most relevant feature sets, which are free of noisy, irrelevant and redundant characteristics.

In general, feature selection is a very complex optimisation problem. For large feature sets, evolutionary algorithms (EA) have proven their ability to solve feature selection tasks within an acceptable number of iterations. Further, EA are even more reasonable, when feature selection and the subsequent classification should be evaluated by several conflicting criteria. One of the most straightforward approaches is the simultaneous minimisation of the number of features and the maximisation of the classification performance. Evolutionary multi-objective algorithms (EMOA) estimate feature subsets with different criterion trade-offs, from large feature sets with high classification performance, to smaller feature sets, which still provide an acceptable classification quality, but strongly reduce the computing and storage demands.

Until now, a predominant share of feature selection applications in music classification has been evaluated by a limited number of criteria, and we are not aware of any further studies (except for several own preliminary publications), which have systematically integrated a multi-objective evolutionary approach to optimise the feature selection process for MIR classification tasks.

Another relevant criterion in music classification is the interpretability of the classification models. Because music classification is a user-centered application, it is advantageous to learn comprehensible properties of the music categories. However, the most common approach is to classify music into genres or other categories from low-level audio signal characteristics. They are less understandable for music listeners and music scientists, who may be interested to identify, e.g., the most important harmonic and melodic characteristics, which define a certain genre or style.

The approach in this work integrates the automatic estimation of high-level features, which are interpretable and describe several musical characteristics: instrumentation, harmony, melody, moods, tempo, rhythm, and structure. These features are partly derived from low-level audio features, and are used themselves as input features for the identification of music genres and styles. The evolutionary multi-objective feature selection helps to identify the most relevant characteristics.

The targets of this thesis can be described as follows:

- To check the essentiality of feature selection for different supervised music classification tasks.
- To apply multi-objective feature selection, optimising two evaluation criteria at the same time.
- To examine the ability of evolutionary multi-objective algorithms to solve the feature selection task for supervised music classification.
- To construct a sufficiently large set of high-level audio features, which are interpretable and can replace baseline low-level features, with no significant decrease of the classification performance.
- To setup the labelling of ground truth closely to real-world scenario demands, so that classification training sets are built only from a small number of songs (we use 20 songs as ground truth for genre and style recognition).
- To evaluate the designed methods in an accurate way, comparing classification performance by means of statistical tests on independent holdout song sets, which neither have been involved into the model training, nor into the optimisation of feature selection.

Further, we had to limit the choice of methods and their adjustments:

- No common statistical feature processing methods, such as principal component analysis, were applied. These methods transform the original feature dimensions, so that the interpretability of high-level features is not kept anymore.
- No deterministic feature selection methods were integrated into the studies. They are less suitable for our targets than multi-objective, population-based heuristics, which may overcome local optima through integrated random components and simulated natural evolution.
- To provide a sufficient amount of experiments with respect to our main preferences, we omitted any tuning of the hyperparameters of classification methods. However, we selected four classification methods with different basic operating methods to examine the performance of feature selection for these methods.
- We restricted both low-level and high-level features to audio features only. Their major advantage is that they can be extracted from any digitally available song, and the digital signal typically has enough information to identify a genre or personal

listener preferences[1].

1.2. Main achievements and structure of the thesis

The main achievements of this thesis are:

- To our knowledge, this is the first work, which addresses feature selection for music classification in a multi-objective way, in particular, by evolutionary multi-objective algorithms (except for several own preliminary and recent studies [217, 219, 216]). The simultaneous minimisation of the balanced classification relative error and the number of selected features leads to significant performance increase with respect to both criteria.
- We introduce the novel concept called sliding feature selection, where several intermediate classification levels provide a bridge between the audio low-level features and the music categories to learn. The high-level features are estimated from other low-level and high-level features, where the optimal feature subsets are found for each subtask by means of multi-objective feature selection. The high-level features, which have been derived by the sliding feature selection, are particularly often selected as relevant for the recognition of genres and styles.
- We designed a set of 566 high-level descriptors. The first part of this set is created by the integration of already existing up-to-date algorithms and own extensions, mostly harmonic and temporal characteristics. A further part contains features estimated after the sliding feature selection: instrumentation, moods, harmony, and melody characteristics. The last part is built by structural complexity features, which measure the progress of the temporal distribution of chords, instruments, harmonic, and time characteristics.
- It is possible to completely replace the baseline low-level feature set by the high-level feature set. The significant, but relatively slight decrease of the classification performance is measured only for the Rap category. For other categories, the performance is equal or even better. For the three examined music styles, which may be treated as personal music categories, the high-level feature set leads even to a significant increase of classification performance in all cases if several classification methods were used in an experiment.
- We adapted our approach to several restrictions of the real-world situation. The training sets are limited to 20 music pieces in order to match the typically high efforts for the definition of ground truth. The features are extracted and processed from complete recordings: in our opinion, the often applied limitation to, e.g., 30 seconds excerpts from the song middle may reduce the classification performance for subgenres, which are characterised by many different and relevant segments. The two-level classification allows us to apply two different evaluation approaches. The evaluation of the classification performance on shorter classification windows

[1] Probably, our method is not well suited for some classification tasks, such as 'West Coast Rap' versus 'East Coast Rap'. However, we do not object the integration of metadata and community features in future studies. We however had to limit the research scope and avoid efforts for the identification of irrelevant or missing data.

of several seconds allows an exact evaluation of methods. For the categorisation of songs into genres and styles, the binary song relationships are estimated by major voting.

- We provide a formal categorisation of the steps, which are necessary in supervised music classification. In particular, different feature processing methods are developed and categorised. Further, we discuss several groups of evaluation metrics, which are reasonable for multi-objective evaluation and optimisation of music classification in future.
- Several recommendations for the systematic evaluation of music classification and feature selection are discussed: the choice of metrics, the choice of the evaluation method, the data set design, and the significance measurement by means of statistical tests.

The thesis is organised as follows:

- In the last introductory section, we list our own previously published contributions.
- Chapter 2 describes the backgrounds of music data analysis and introduces the methods for audio feature extraction, feature processing, and classification in detail.
- In Chapter 3, we discuss the goals of feature selection. The basics of evolutionary multi-objective optimisation and feature selection are introduced as well as the concrete algorithm adjustments for our studies. In the last sections of this chapter, we list references to studies with evolutionary feature selection and feature selection in music classification.
- Chapter 4 deals with the four essential components of reliable algorithm evaluation: evaluation metrics, evaluation methods, data set design, and statistical tests for significance measurement.
- In Chapter 5, we present the results of our studies, where the evolutionary multi-objective feature selection has been applied for several music classification tasks. The description of the studies related to high-level feature recognition (instruments, moods, harmonic and melodic characteristics) is limited to the experiment setup and study outcomes (Section 5.1). In the second part of the chapter (Section 5.2), we address genre and style recognition based on low-level and high-level features. Both sets are compared with different evaluation criteria, and the study outcomes are underlined by statistical tests. Finally, we discuss properties of different high-level feature groups and list the high-level features, which were most often selected for each combination of a classifier and a classification task.
- In Chapter 6, we summarise the results of our work and discuss several directions for further research.

1.3. Previous own publications

Some of the investigations, which were preliminary for this thesis and have influenced the design of the final study, contributed to several publications in recent years. Here, we provide a list with the corresponding references, sorted by the publication year. If the

thesis author was not the first publication author, explanations about his contribution to the joint work are provided.

- 2008: [205] is the first peer-reviewed conference publication. Here, we discussed our first steps in music classification based on audio features. Three personal music categories were predicted using the C4.5 decision tree algorithm. The thesis author's contribution was mainly the design and the development of the very first version of the Advanced MUSic Explorer (AMUSE) framework for the simple implementation of different subtasks in music classification and their evaluation.

  In [220], we applied an evolutionary strategy (ES) for the optimisation of feature selection and the length of classification windows for the first time.

- 2009: The previous work was extended in [223], where enhancements to the evolutionary algorithms were implemented: several variants of local search and a self-adaptation approach. Since this study, we focused on evolutionary feature selection.

- 2010: The first open source version of AMUSE, available on SourceForge[2], was described in [221].

  In [15], we designed several memetic and self-adaptive ES for music classification. Though these algorithms were later integrated by the thesis author into AMUSE, the implementation of ES for this study and the experimental analysis was done by the first author of the reference. This was the first work, where we switched to the recognition of AllMusicGuide genres and styles, after a thorough redesign of our song collection.

  The following three studies focused on different aspects of the music classification chain: extension of the feature set and classification methods, analysis of feature processing methods, and comparison of different evaluation metrics. In all of these publications no feature selection was applied. However, the outcomes of the studies lead to the corresponding improvements in AMUSE as a basis for further experiments.

  In [157], we extended the experiment scale: new cepstral features contributed by the first author of the work were integrated, the number of the classifiers was firstly increased to the four different methods, and 14 genre and style categories were classified. All experiments were run within AMUSE and analysed by the thesis author.

  The extensive comparison of the feature processing methods, which influenced the choice of the corresponding method in this thesis, was done in [222].

  Different confusion matrix metrics were compared for music genre and style classification in [214], and it was shown that some of them were loosely correlated. This was a first work, where we explained the reasons to integrate not only multi-objective evaluation, but also multi-objective optimisation into a music classification scenario.

- 2011: The first application of the multi-objective feature selection for genre and style recognition was done in [217]. Here, we optimised in the first part of the study accuracy and selected feature rate at the same time. Afterwards, recall and specificity were maximised.

[2] http://amuse-framework.sourceforge.net/, date of visit: 15.02.2013.

In [186], the ground truth was extremely limited to only five positive songs, which should define a personal music category. The task was not to classify the songs, but to design a set of high-level characteristics, which should be useful for further classification. The contribution of the thesis author was to design and run a study, where these high-level descriptors were predicted from low-level features, and to measure the impact of high-level features on categories, compared to random feature distribution.

- 2012: [219] was the first work, in which evolutionary multi-objective feature selection was applied for instrument recognition in polyphonic mixtures.

  In [215], we applied statistical tests for the significance measurement of the results from [219].

Other peer-reviewed publications with the participation of the thesis author [226, 17, 40, 134, 93, 218, 216] had a rather marginal contribution to this thesis, but led to many worthwhile insights into other MIR related tasks.

2. Music Data Analysis

2.1. Background

Doubtlessly, music was analysed in all stages of its historical development: the construction of prehistorical instruments required the knowledge of different materials as sound sources and resonating bodies. The tuning process was and remains necessary for a large number of instruments. The composition of music pieces almost always followed rules depending on the genre. For example, the well-known Mozart's dice game [34] created music by random variations of several pre-composed parts, based on the former music rules for waltzes.

However, MUSIC DATA ANALYSIS can be rather understood as an *automatic* analysis of a *large* amount of music-related data, mostly done with the help of *computers*. The process of music rule creation or the calculation of music characteristics is not done anymore by a human, but is produced by an algorithm, which nevertheless may or may not provide interaction with its creator or a user. Such an analysis can be done in a more efficient way, compared to the engagement of music experts: modern server farms and grid systems can analyse thousands of music pieces in a few hours or even minutes. Also, the management of very large music collections demands new methods, which are capable to deal even with millions of music tracks [25]. Expert knowledge is still required at the beginning: for the choice of proper features for learning, for the assignment of the category labels to music instances, or, as an example for a more specific application, for the analysis of the pitch distribution profiles for key prediction [90]. After this information has been assembled into software or hardware, music can be processed in an accurate way without any signs of tiredness, personal preferences, or disposition.

[164], p. 2, defines music information retrieval (MIR) as a user-driven research field which is focused on listener needs on "music management, easy access, and enjoyment". It is indeed a widespread interdisciplinary research domain based on studies in music science and psychology, musicology and psychoacoustics, engineering and signal processing, computer science, data mining and statistics, neuroscience, and other fields. The combined investigations from these very different sciences provide on the first side a strong enrichment and a high potential to solve many related applications in unconventional ways. On the other side, this diversity leads to many challenges. Downie [44] discusses "a blessing and a curse" of the rich intellectual diversity of the MIR research, providing examples of the very different terms and techniques coming from different research communities, e.g., an enharmonic equivalence of the tones G♯ and A♭ for a signal processing expert, but a clear difference for a musicologist.

[44] refers to [97] as the earliest published MIR work and provides further references of the related publications. From 2000, an annual society on music information retrieval conference (ISMIR) facilitated a strong increase of the related studies, and also provided an international forum for exchange and joint research in the rapidly growing MIR community.

In recent years, several textbooks were published, which provided a structured introduction into MIR methods and applications. [120] gives an introduction into data mining in general, before specific music data analysis problems and applications are discussed in detail. [154] describes enhanced methods for chroma analysis in music audio data. [101, 181] provide collections of many current research studies, such as harmony recognition, music classification, source separation, music transcription, etc.

A promising possibility for future interdisciplinary MIR research is to integrate more results of studies from music theory scientists, which describe music structure or integrate rule-based music characteristics into automatic computer-guided approaches [204, 35].

In our work, we often use the terms music (data) analysis and music classification. The first one is intended to be a synonym of MIR. Music classification is one of the most important subtasks of MIR and is discussed among a wide range of possible MIR applications in Section 2.1.1. Section 2.1.2 describes different sources for music data analysis. Section 2.1.3 introduces the categorisation of algorithms for supervised music classification.

2.1.1. Application scenarios

Without claiming to present a complete list of all possible applications of music data analysis, we provide here an overview of the major and important listener-centered tasks and scenarios.

Probably the largest part of MIR applications is covered by CLASSIFICATION tasks (other words used as synonyms: detection, discovering, identification, recognition, tagging, etc.). Here, a given instance of music data (a song, song segment, tone phrase, melody, etc.) is categorised into one or several classes. The following applications of classification may be distinguished:

- RECOGNITION OF HIGH-LEVEL MUSIC CATEGORIES for music collections. Maybe the most studied approach is to recognise music genres [1]; [200] provides an extensive list with several hundred related publications. However, genre recognition has several serious drawbacks: no common genre taxonomy exists [165], genres may evolve during years, and the role of different subgenres may be of different importance for users – e.g., a person may organise all classical pieces in a collection as a 'classical' part or prefer to distinguish between baroque, romantic, and modern classic. Therefore, learning the listener's personal preferences is a promising approach for automatic classifier systems, as we discussed in [205].
- RECOMMENDATION SYSTEMS aim at the presentation of new music to a user, which should satisfy her or his preferences [26].
- SIMILARITY ANALYSIS enables the search for similar music tracks, e.g., for recommendation or cover song detection. Another use case is plagiarism detection through search of similar melodies [42]. A necessary design step for similarity analysis is the choice of a distance metric for the corresponding feature space, e.g., Euclidean or cosine distance. Some related measures were compared in [12].
- QUERY-BY-HUMMING and fingerprinting systems, such as Shazam [224], help to identify a song by a hummed melody or a short audio sequence.

- Identification of high-level characteristics (see later Section 2.2.1): instruments and vocals, harmonic characteristics (e.g., chords or chord progressions), tempo and rhythm, lyrics and melodic motifs, induced emotions, artist or composer, playing style (e.g., staccato for piano or tapping for guitar). A proposal to categorise these characteristics by seven facets (pitch, temporal, harmonic, timbral, editorial, textual, and bibliographic) was done in [44]. In [11], high-level features are referred to as semantic features and are used for music organisation in a so-called anchor space.

- Segmentation can be treated as a specific case: it is a classification task, where certain song parts like verse, chorus, or intro are recognised. However, it is possible to structure audio recordings into different segments based on low- or high-level characteristics, e.g., harmonic segmentation or instrumental segmentation.

Another essential method in MIR, which is required for classification, is feature extraction. Features are mostly numerical characteristics, which are integrated into the categorisation models. In some situations, feature extraction can incorporate the classification itself, if high-level features (e.g., instruments) are extracted by a classification model based on low-level features (see Section 5.1). Another possibility to extract high-level characteristics is to learn the rules, which are relevant for a certain genre, for example, the frequencies of the consecutive fifths and octaves, which are forbidden in musical counterpoint.

Further examples of music analysis tasks are:

- Score alignment maps the timeline of an audio recording to the corresponding time events in the score [58].

- Music transcription creates the score from audio [101]. This very complex application is often based on source separation, which outputs single representations of the sound sources, i.e. different playing instruments, or orchestra and a solo instrument [79]. Another related task is the correction of misplaced notes.

- Generative and synthesis tasks create new music based, e.g., on a set of rules [35] or an evolutionary process [149]. Another method, which is also closely related to similarity analysis and recommendation, is the generation of playlists [63].

- Music games provide entertainment and fun integrated into music listening, learning, or music collection organisation. Especially with the growing number of mobile devices, many new application scenarios became available, e.g., control by gyroscope or the procedural content generation based on audio music features [93].

- The enhancements of hearing aids concentrate not only on speech understanding, but also on listening to music [77].

- The studies from musicology and music psychology, such as estimation of the music influence [32], can be incorporated into algorithm-based music analysis.

- Visualisation of music songs can be a great help for the management of music collections, for example, based on self-organised maps [153] or by the creation of trajectories with slowly changing style [205]. A specific challenge is given, if music is accessed by a mobile device with a small screen [17].

All these tasks have in common that almost always feature extraction is required to represent music, and it is a part of the algorithm chain. Many further interactions between these tasks are possible: hearing aids may profit from the modelling of the psychological aspects of music hearing, music generation can be a part of a game, music transcription may improve the music alignment, and so on.

2.1.2. Music data sources

The development of music storage and recording over the past decades and centuries established several sources for different groups of features:

- The oldest possibility to capture played music is to create the SCORE. It is unclear, when exactly the very first notation systems were developed. However, the alphabet letters were used in Greece for music notation around 200 BC, and many musical activities in ancient Egypt were documented from about 2600 BC [129]. The taxonomy of symbols and notation systems for polyphonic music in the quite well investigated western notation evolved and altered strongly over centuries. For example, the transition from *black* (solid, filled with black colour) to *white* (hole notes) notation around the 15th century was motivated by the invention of thin paper, which was not well suited for filling with ink compared to earlier parchments [6]. Figure 2.1 gives examples of two older notations. In comparison to modern systems, many earlier notations did not provide very exact information. For example, the specification of pitch began around 1100 AD. The note lengths of older systems corresponded to the absolute time length, and the specification of tempo (with relative duration of note lengths to each other) began around the 16th century [6]. Another obvious limitation of the score is that it cannot restore a one-to-one copy of the once played music piece, i.e. the exact timbre of the instruments, tempo and rhythm variations by a certain interpret, or different playing techniques. On the other side, the score enables the simple estimation of many high-level features: instrumentation, tempo (if it is exactly described), harmonic structure and key, number of non-harmonic notes, melody line characteristics, etc.

- ANALOGUE RECORDING made it possible to replay a musical piece with almost complete similarity a large number of times. The techniques for analogue recording can be distinguished into *mechanical*, *magnetic*, and *optical*. The probably first invented mechanical music instrument, which could *reproduce* a music piece, was the hydraulic "Banu Musa" water organ. The interchangeable cylinders stored music by raised pins [65]. The mechanical possibility to *record* music was significantly boosted by two inventions: the phonograph by Thomas Edison in 1877, where the sound waves were converted to impressions in tinfoil, and later the gramophone by Emil Berliner in 1888, which used discs instead of cylinders[1]. This technique is still in use in LP (long play) recordings, and became even a comeback in popularity in the recent past. The magnet-based recording started in the late 1920s by the invention of the magnetic tape recorder, which saves music on tape with the help of the alternating magnet field. In optical recording devices, the tape has a varying light sensitivity

[1] As mentioned in [148], the person, who posthumously became the first inventor of mechanical recording, was Edouard-Léon Scott de Martinville, who built the phonautograph in 1857.

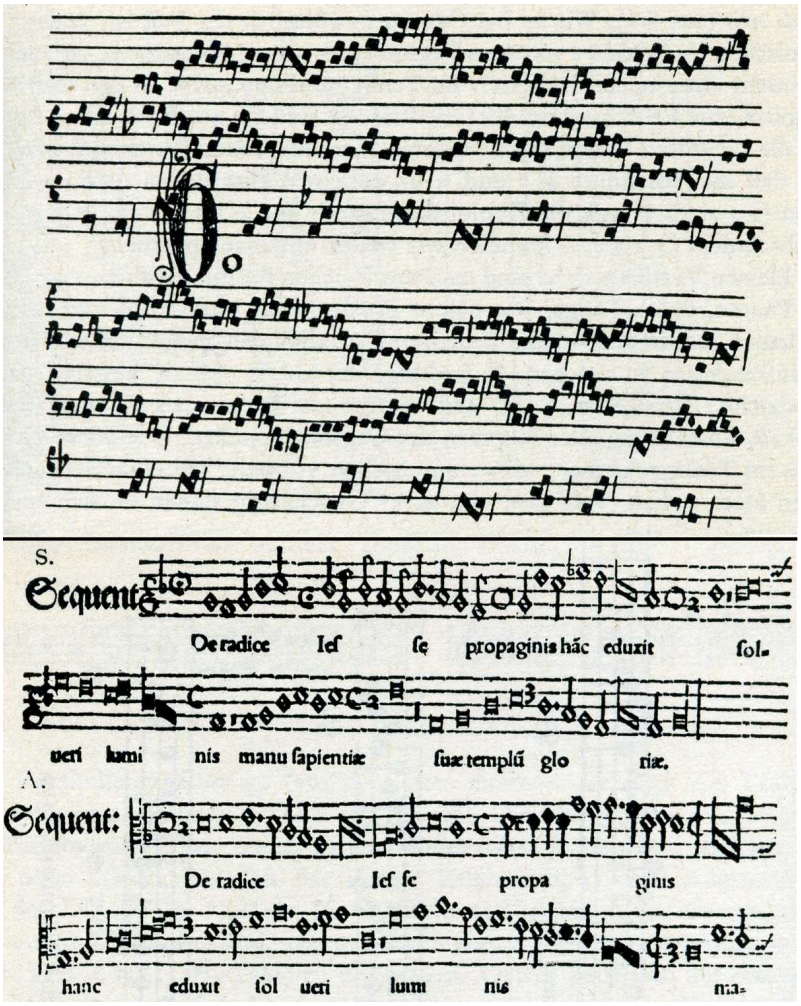

Figure 2.1.: Examples for earlier notations. Upper subfigure (black notation): Go; Flos folius est (Florence, plut. 29. I.), Bibliotheca Laurenziana, 13th century. Source: [6], p. 253. Lower subfigure (white notation): Heinrich Isaac, De radice (Choralis Constantinus), Formschneyder, Nuremberg, 1550. Source: [6], p. 188.

and music is played by a conversion of the light ray into electric waves, and then into sound waves.

- Since the late 1970s DIGITAL RECORDING became more important. Here, music is transformed to a bit sequence or digital signal. This approach originates from the Morse code, developed in the 19th century, where short and long signals were distinguished for radio transmission [197]. Digital recording enabled previously unimaginable possibilities to reproduce once played songs with a quality that sounds completely identical for a human listener. Because only two coding levels are used, the danger of different errors is strongly reduced, compared to the situation, where more different voltage levels are used for coding. On the other side, digital recording has also several drawbacks, such as the necessity for a larger bandwidth for data transmission and for complex error-detection methods [197]. Philips and Sony invented

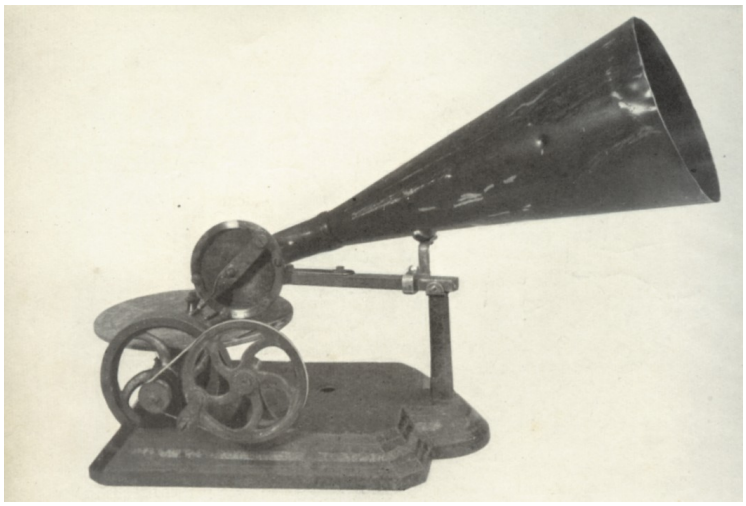

Figure 2.2.: Berliner Gramophone 1888. Source: http://www.charm.rhul.ac.uk/history/p20_4_1.html (c). Date of visit: 15.02.2013.

the CD in the early 1980s (the first commercial CD release was 1982), where the audio signal was stored by a laser beam of light. One of the oldest formats for digital storage is the pulse-code modulation (PCM), which saves a bit string representation of the previously sampled and quantised signal (see later Section 2.2.2), and was originally invented by Alec Harley Reeves in 1937. Later formats compressed the size of the bit sequence, keeping the quality of the music performance (LOSSLESS COMPRESSION), or provided stronger compression reducing the quality of the original performance (LOSSY COMPRESSION), for example removing the frequencies which are hardly perceived by human listeners or are masked by other frequencies. The currently most distributed format, the MP3, was invented by the research group around Karlheinz Brandenburg in 1992[2].

Digital recording techniques and the integration of computers into music composition also influenced the music notation by inventions of the digital score formats, such as the musical instrument digital interface (MIDI)[3] and MusicXML[4].

- Another source for music-related features are METADATA: documents, which describe music in a textual form. In [25], it is distinguished between *factual* and *cultural* metadata: the first ones are objective and describe, e.g., the circumstances of the music piece creation: composition year, age and experience of the composer, country of creation, and so on. Cultural metadata are subjective and cannot be defined very precisely. Examples of such descriptors are genres and subgenres, denoted by music critics and experts.

  With the expanding growth of the INTERNET and especially the SOCIAL EXCHANGE in the first decade of the 21th century, it became possible to save and analyse a

[2] MP3s with a bit rate of 192 kbps require approximately a seventh of audio CD storage demands and could not be significantly distinguished from CD recordings through the blind study in [142]. Further tests on comparison of the CD recordings and digital lossy formats, investigated by Communications Research Centre Canada, are discussed in [148].

[3] http://www.midi.org/, date of visit: 15.02.2013.

[4] http://www.makemusic.com/musicxml/specification, date of visit: 15.02.2013.

vast number of music metadata descriptors. The databases which are maintained by music experts contain a large number of characteristics, like genres, styles, and moods on the AllMusicGuide[5] (AMG) web site, or high-level characteristics created through the Music Genome Project for Pandora web radio[6] [94]. Another kind of data are generated by music listeners: the statistics from music playlists measure the popularity of music pieces or may help to detect similarities related to listener preferences. Tags of well-established music communities, such as Last.FM[7], provide many descriptors even for less popular songs. These personal tags can also be predicted from audio signal [13] and used as features for the detection of genre and personal preferences.

- A less studied data source, which may be promising especially for prediction of personal preferences, is the listening CONTEXT [56]. Feature estimation can be done by the analysis of environmental sounds for the identification of personal music preferences during car driving, shopping, eating, etc. An application example, which integrates context data into music classification, is the measurement of the runner step frequency for the selection of music with the appropriate tempo [159].

2.1.3. Algorithm chain

Before we provide a formal categorisation of supervised classification chain methods, the three terms, which describe the building of time windows in music classification, should be clarified (the term *window* is used as synonym to *frame* in this thesis):

- (FEATURE) EXTRACTION WINDOW is used for the estimation of an audio feature (see also the discussion of windowing in Section 2.2.2). Short extraction frames are usually of the length of several tens of milliseconds. Larger frames of several tens of seconds are necessary for the estimation of tempo and related characteristics. Some features, such as music piece duration, have extraction frames equal to the complete song length. We denote the length of the extraction window in samples by W_e, and the step size by S_e.
- (ALGORITHM) ANALYSIS WINDOW is a usually larger time window, incorporated into the estimation of a feature from many small extraction frames. As an example, low energy measures the energy of a certain number of short frames *before* the frame for which this feature is estimated. Algorithm analysis windows are also built during the estimation of structural complexity, as described in Section 2.3.3, where a given number of seconds *before* and *after* a short frame is analysed. The algorithm analysis window length in seconds is denoted by W_a, and the step size by S_a.
- (FEATURE) AGGREGATION OR CLASSIFICATION WINDOW is a time interval from a music song, which is used for the training of classification models or classification (in general data mining terms, it is also called a *classification instance*). For high-level categories, such as genres, reasonable sizes of classification windows are typically between several seconds and a complete song. In [211, 120], the windows used for classification based on timbre characteristics are also referred to as *texture* windows.

[5] http://www.allmusic.com, date of visit: 15.02.2013.
[6] http://www.pandora.com, date of visit: 15.02.2013.
[7] http://www.last.fm, date of visit: 15.02.2013.

In some of our previous works, e.g., [222], we describe them as song *partitions*. The classification window length in seconds is denoted by W_c and the step size by S_c.

Figure 2.3 provides an overview of the essential tasks and the data flow in any automatic system for supervised classification of music data. In the following, we define the corresponding methods for song-centered music classification.

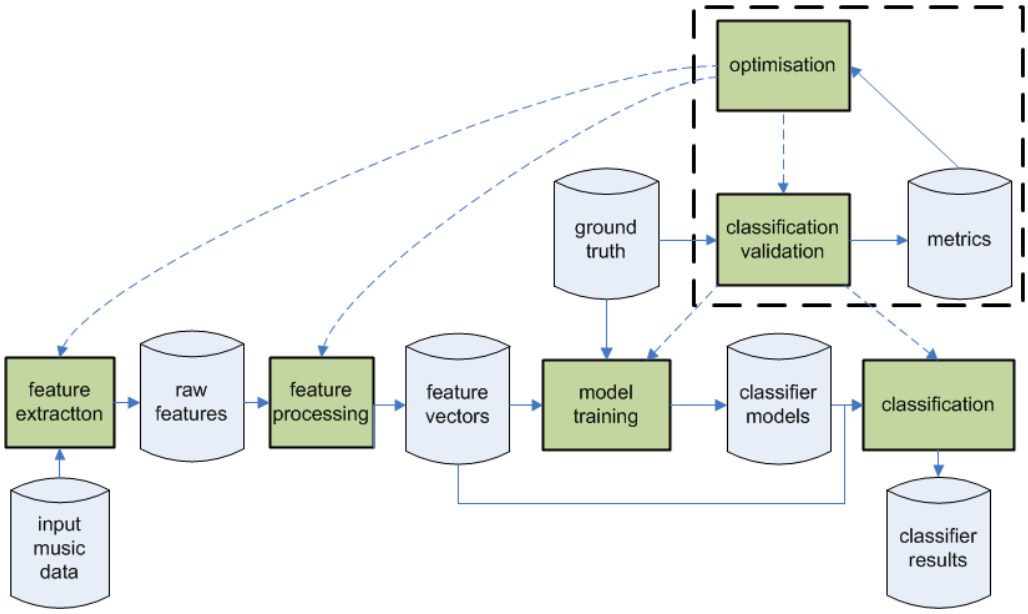

Figure 2.3.: Algorithm chain in music classification, adapted from [221]. Solid lines depict the data flow, whereas dashed lines indicate nested relationships between the methods (e.g., classification validation usually starts several model training and classification cycles for the metric estimation). It is distinguished between base classification methods (outside the rectangle with a dashed boundary) and evaluation and optimisation methods (inside the rectangle).

The *discrete audio signal* (see also Section 2.2.2) is a form of a REAL-VALUED UNIVARIATE TIME SERIES:

Definition 2.1 $\mathcal{TS} : \mathbb{N}^D \rightarrow (\mathbb{R} \times \mathbb{R})^D$, *where an index vector of length D is mapped to a D-tuple of pairs $\in \mathbb{R} \times \mathbb{R}$. The first entry of each pair denotes the real-valued time positions of equidistant time events and the second entry of each pair contains the corresponding real-valued characteristic (feature) for these time events.*

Usually, many features are extracted from the audio signal, resulting in a MULTIVARIATE SERIES. Further, the time positions of the extraction frames are not necessarily equidistant, and complex-valued transforms are possible. Therefore, the feature matrix can be treated as a product of a more general VALUE SERIES mapping (we adapt the definition from [146]):

Definition 2.2 $\mathcal{VS} : \mathbb{N}^D \rightarrow \left(\mathbb{R} \times \mathbb{C}^{F^*}\right)^D$, *where an index vector of length D is mapped to a D-tuple of pairs $\in \mathbb{R} \times \mathbb{C}^{F^*}$. The first entry of each pair denotes real-valued time positions, and the second entry of each pair contains a vector with F^* corresponding complex characteristics (features).*

The FEATURE EXTRACTION task $\mathcal{FE}$ is responsible for the storage of the numerical values, which characterise the original music data. In all studies within the scope of this thesis we saved only the real-valued feature matrices after all internal transforms, and we restrict the feature space to be real-valued. For a detailed discussion of audio feature extraction, see Section 2.2.

Because each feature may be multi-dimensional and several features may be extracted from the extraction frames of different lengths, we distinguish between the dimensionality $F^{**}(i)$ of a single raw feature i, the overall number of its extraction frames $T^{**}(i)$, the number of raw features F_N^*, and the overall sum F^* of feature dimensions after $\mathcal{FE}$[8]:

Definition 2.3 *The **real-valued song-level windowed audio feature extraction task** (referred to later as 'feature extraction') $\mathcal{FE} : \mathbf{s} \rightarrow (\mathbf{X}^*(1), ..., \mathbf{X}^*(F^*))$, where $\mathbf{s} \in \mathbb{R}^D$ is a discrete song audio signal time series of length D, $\mathbf{X}^*(i) \in \mathbb{R}^{F^{**}(i) \times T^{**}(i)}$ is the **raw song feature matrix** of $F^{**}(i)$ feature dimensions and $T^{**}(i)$ extraction windows of a single feature i, $i \in [1; F_N^*] \cap \mathbb{N}$, $F^* = \sum_{i=1}^{F_N^*} F^{**}(i)$, and $D \gg T^{**}(i)\ \forall\ i$.*

The next step, the FEATURE PROCESSING $\mathcal{FP}$, has two aims. At first, the raw features should be preprocessed for classification by normalisation, handling of non-defined values and missing data, elimination of redundant features, etc. The second objective is to create classification instances from appropriate time intervals (classification windows): the original feature extraction windows are usually too short, and on the other side it often does not make sense to classify complete songs, because they contain very different segments. Section 2.3 provides a categorisation of methods for $\mathcal{FP}$.

Definition 2.4 *The **real-valued song-level feature processing task** (referred to later as 'feature processing') $\mathcal{FP} : (\mathbf{X}^*(1), ..., \mathbf{X}^*(F^*)) \mapsto \mathbf{X}'$, where $\mathbf{X}' \in \mathbb{R}^{F \times T'}$ is the **song processed feature matrix** of F feature dimensions and T' classification windows.*

The goal of classification is to estimate class relationships for unlabelled data. The first part of this procedure, the CLASSIFICATION TRAINING task $\mathcal{CT}$, creates a classification model, which maps the preprocessed feature values to one or more class relationships (we restrict the definition to a single-class scenario, where only one category is predicted by each $\mathcal{CT}$, see also Section 2.4):

Definition 2.5 *The **real-valued single-class classification training task** (referred to later as 'classification training') $\mathcal{CT} : (\mathbf{X}, \mathbf{y}_L) \mapsto \mathcal{M}_C$, where $\mathbf{X} \in \mathbb{R}^{F \times T}$ is the **processed feature matrix** of F feature dimensions and T classification windows for **one or several songs**, $\mathbf{y}_L \in [0; 1]^T$ is the vector of labelled category relationships of the T classification windows **(ground truth)**, and $\mathcal{M}_C$ is the classification model, which maps feature values to a predicted relationship.*

Consequently, the CLASSIFICATION task $\mathcal{C}$ is defined as follows:

Definition 2.6 *The **real-valued single-class classification task** (referred to later as 'classification') $\mathcal{C} : (\mathbf{X}, \mathcal{M}_C) \mapsto \mathbf{y}_P$, where $\mathbf{X} \in \mathbb{R}^{F \times T}$ is the **processed feature matrix***

[8] For example, if a 12-dimensional mel frequency cepstral coefficients feature, and the amplitudes of 5 spectral peaks are extracted, $F_N^* = 2$, $F^{**}(1) = 12$, $F^{**}(2) = 5$, and $F^* = 17$.

*of F feature dimensions and T classification windows for **one or several songs**, $\mathcal{M}_C$ is the classification model, and $\mathbf{y}_P \in [0;1]^T$ is the predicted class relationship vector.*

Another group of classification chain methods is applied to evaluate and to tune a classification system.

The VALIDATION task $\mathcal{V}$ provides a feedback how well a classification model performs:

Definition 2.7 *The **real-valued single classification validation task** (referred to later as 'classification validation') $\mathcal{V} : (\mathbf{X}, \mathbf{y}_L, \mathcal{M}_C) \mapsto \mathbf{m}$, where $\mathbf{X} \in \mathbb{R}^{F \times T}$ is the **processed feature matrix** of F feature dimensions and T classification windows for **one or several songs**, $\mathbf{y}_L \in [0;1]^T$ is the vector of labelled category relationships of the T classification windows **(ground truth)**, $\mathcal{M}_C$ is the classification model to validate, and $\mathbf{m} \in \mathbb{R}$ is the vector of O evaluation metrics.*

It is obvious that the data for training and validation of a model $\mathcal{M}_C$ should have as small as possible overlap, at optimal equal to zero.

Usually, several validation tasks are combined, e.g., based on the n-fold cross-validation principle. Then, $\mathcal{V}$ creates n models from the given labelled instances ($\mathcal{V} : (\mathbf{X}, \mathbf{y}_L) \mapsto \mathbf{m}$) and starts both tasks $\mathcal{CT}$ and $\mathcal{C}$ as marked with the dashed lines in Fig. 2.3, see Section 4.2 for further details.

Finally, the OPTIMISATION task $\mathcal{O}_S$ tunes the parameters of the related methods of the classification chain with respect to some evaluation criterion, or metric, for example, classification quality. For simplicity reasons, we restrict the definition to metric minimisation: the metrics to be maximised can be easily modified for minimisation[9]. Further, we do not explicitly distinguish between *evaluation* and the *optimisation* metrics, which are both denoted by the same symbols m (for a single metric) and $\mathbf{m}$ (for several metrics). Usually, the same metric functions are used for evaluation and optimisation, however with different purposes: whereas the evaluation only measures algorithm performance, optimisation systematically tunes the algorithms for better performance measured by corresponding metrics. This achievement goes often hand in hand with a diminished performance of other evaluation metrics, which are less correlated with an optimisation metric.

Definition 2.8 *The **single-objective algorithm chain optimisation task** $\mathcal{O}_S : \mathbf{p}_\mathcal{T}^o = \arg\min_{\mathbf{p}_\mathcal{T}} m(\mathbf{p}_\mathcal{T})$, where $\mathcal{T} \subseteq (\mathcal{FE}, \mathcal{FP}, \mathcal{CT}, \mathcal{C}, \mathcal{V})$ are the tasks to optimise, $\mathbf{p}_\mathcal{T}$ the corresponding parameters to tune, m the optimisation metric, and $\mathbf{p}_\mathcal{T}^o$ is the optimal parameter vector.*

Many different and conflicting aspects of the classification chain can be evaluated, e.g., classification accuracy, runtime and storage costs, or performance on highly imbalanced sets. For example, if an algorithm provides a successful categorisation of music, but has very high demands for feature storage and works very slowly, it is completely meaningless in a real-world application. Therefore, a *multi-objective* or *multi-criteria* optimisation

[9] In general, for a function $f(x)$ it holds: $\max\{f(x)\} = -\min\{-f(x)\} \Rightarrow \arg\max\{f(x)\} = \arg\min\{-f(x)\}$. One of the simplest modifications can be applied as follows: in the first step the target set of a metric function m_j is normalised so that $m_j \in [0;1]$. Then, the optimisation goal is set to the minimisation of $1 - m_j$.

(MOO) task $\mathcal{O}_M$ becomes reasonable (the formal definitions of multi-objective optimisation terms are provided in Section 3.2.2):

Definition 2.9 *The **multi-objective algorithm chain optimisation task** $\mathcal{O}_M : \mathbf{P}^o_{\mathcal{T}} = \arg\min_{\mathbf{p}_{\mathcal{T}}} \mathbf{m}(\mathbf{p}_{\mathcal{T}})$, where $\mathcal{T} \subseteq \{\mathcal{FE}, \mathcal{FP}, \mathcal{CT}, \mathcal{C}, \mathcal{V}\}$ are the tasks to optimise, $\mathbf{p}_{\mathcal{T}}$ the corresponding parameters to tune, $\mathbf{m}$ is the vector of optimisation metrics, and $\mathbf{P^o}_{\mathcal{T}}$ is the matrix of the best non-comparable parameter vectors.*

It is obvious that the optimisation of highly correlated criteria does not make sense (minimisation of one metric leads directly to smaller values of another one). However, often two or more relevant and less correlated evaluation criteria can be selected. Then, the search of trade-off parameter sets helps to decide, which parameters are preferable for a certain situation.

2.2. Feature extraction

As discussed in the previous section, the feature extraction $\mathcal{FE}$ is the very first step for any classification task. Feature extraction must be carefully planned, since the impact of too many irrelevant or noisy features can be hardly neglected by any sophisticated classifier. It is often preferable to design a set of meaningful features, which may provide an (almost) linear separation of the data instances from different categories, instead of spending too much time on tuning the classification methods.

Each classification task may require its own features. Therefore, it is reasonable to start with a large feature set and to apply a feature selection procedure (discussed in Chapter 3) afterwards for the identification of the most representative characteristics.

Section 2.2.1 deals with the interpretability and the musical meaning of features, and the differences between low-level and high-level descriptors are discussed. Section 2.2.2 provides a short introduction into the signal processing steps required for $\mathcal{FE}$. The following sections give an overview of the audio features, which are used in this work, from timbre and energy to instrumentation and harmony characteristics.

2.2.1. Low-level and high-level descriptors

Interpretable and *music theory-related* features, which describe instrumentation, rhythmic structure, chord progression, melodies, etc., significantly improve the understandability of the classification models. When using these kind of features, the organisation of music collections or recommendation of new music becomes intuitive and helpful for music listeners.

The question is, which characteristics can be referred to as interpretable and understandable, and how can we distinguish between *low-level* and *high-level* features? In fact, no common agreement or clear definition exists. Some of the related works describe the statistics that are generated from other features already as high-level [153, 174], other limit this definition to the characteristics that may be estimated from the score [138] or name the features high-level, if a complex method sequence was developed for their extraction [24, 51, 173].

The definitions above do not suggest that high-level descriptors should be related to music theory and be understandable by music scientists and human listeners without a university degree in sound engineering and acoustics. This statement is however supported by other publications. In [230], onsets, notes, melodies, and harmonies are referred to as the high-level objects. Rhythm, melody, and harmony are mentioned in [236]. Music genres are classified by rhythm and chord characteristics in [7] and by instrument-related features in [238]. [191] includes also genre, mood, speaker characteristics, and lyrics into a list of high-level descriptors. In [27], three different abstraction layers based on interaction with a human listener are distinguished for music characteristics: low-level audio signal features describe physical and spectral characteristics, such as loudness, duration, or energy. The mid-level layer consists of descriptors closer to music listener, e.g., key and mode, rhythm, dynamics, and harmony. The high-level layer comprises the characteristics closely related to listener: emotions, opinions, memories, etc.

We provide the following definition within the scope of this thesis:

Definition 2.10 *The processed feature matrix* $\mathbf{X}'$ *is referred to as being* ***high-level****, iff it contains the values describing either:*

- *instrumentation (instruments, methods to play them, applied digital effects, etc.),*
- *harmonic descriptors (key and mode, chords and their progression, characteristics of harmonic and non-harmonic notes, etc.),*
- *melody (melodic contours, share of melodic segments, etc.),*
- *rhythm, tempo and structure characteristics,*
- *emotional and contextual impact on the listener, as well as listening habits, or*
- *metadata, which describe the source of a music piece (place of composition, composer's age, etc.) and lyrics.*

Furthermore, we list the preconditions, which are **not** assumed to hold for all high-level features:

- Dependency on the data source: High-level features may be estimated either from the score, audio, or any further sources, sometimes exclusively. For example, the application of a digital effect, like hall, may be identified only from audio, if it is not mentioned in the score. On the other side, the tempo is often indicated in the score, but also can be calculated from audio.
- Characterisation by a precise numerical value: For example, the parallel keys, e.g., C-major and a-minor, are built from the same halftones, and in some cases the key relationship is not crisp.
- Invariance for different recordings of the same piece: Even if many harmonic characteristics are usually kept, the same piece may be played by different instruments and with different tempi.
- A clear definition boundary to low-level features is not always possible and is vague in certain cases: e.g., a chroma vector corresponds to a wrapped semitone spectrum and is rather low-level, whereas the strongest chroma component may correspond to high-level harmonic characteristics of a piece.

- A clear advantage against low-level features w.r.t. classification quality: Some recent works stated that very short audio intervals are sufficient to classify songs into genres. 250 ms were already enough in many cases for the identification of 10 genres in [70], and segments of 400 ms were used for the identification of artist and title in [109]. Many high-level characteristics cannot be properly extracted from such a short frame. However, it does not change the fact that only the classification models built from high-level features are really interpretable and helpful for listeners, if the properties of certain genres should be studied. Also, both above mentioned studies do not cover all possible enhanced music categorisation tasks: the identification of specific music subgenres or personal categories was beyond their scope, so that further investigations are required[10]. Furthermore, several studies exist, which claim the improvement of classification performance by the combination of different groups of features (including high-level descriptors): low-level and instrumental features in [238], audio and symbolic features in [121], low-level and high-level audio features in [7], and a combination of audio, cultural and symbolic features in [138]. Another issue is that if the high-level features are estimated from low-level features (which holds in many cases), they do not provide *additional* knowledge for supervised classification, but aggregate the low-level characteristics into a less dimensional and comprehensible representation.
- Music community tags [47] can be treated as high-level features in many cases, since they may well describe the characteristics from Def. 2.10. However, this is not always the case, and such tags may contain false, contradictory, or irrelevant information.

Table 2.1 provides several examples of low-level and high-level features (for definitions and discussion of these features see Section 2.2.3).

Table 2.1.: Examples of low-level and high-level features.

Low-level		**High-level**	
Group	**Example**	**Group**	**Example**
Timbre	Tristimulus	Instruments	Piano share
Energy	Low energy ratio	Playing style	Staccato
Chroma	Bass chroma vector	Harmony	Relative strengths of fifths
Autocorrelation	Strongest autocorrelation peak	Tempo and rhythm	Beats per minute, waltz
Envelope distribution	Linear prediction coefficients	Structure	Number of different segments

Some publications provide extensive lists of descriptors, which can be referred to as high-

[10] It is easy to define counterexample categories, where the classification from short intervals would achieve its limit, and the music preferences of a listener depend clearly on long-framed high-level features: consider a categorisation of 'progressive rock' versus 'folk rock'. The identification of progressive rock songs may perform well only by detection of complex rhythmic patterns and the share of longer instrumental segments with orchestra. Other progressive rock song segments may be similar to folk rock songs. Therefore, we propose to interpret with caution the statement that very short intervals are enough to identify a music category: it may indeed perform well for the recognition of popular and distinctive genres, but not for the more challenging identification of personal categories.

level within the scope of Def. 2.10. [201] provides a list of parameters of musical expression for music analysis, comprising time, melodic, orchestration, tonality, dynamic, acoustical, and mechanical aspects. [138] contains descriptions of 153 symbolic features (instrumentation, musical texture, rhythm, dynamics, pitch statistics, melody characteristics, and chords). In our previous study [186], 61 high-level features (instruments, singing and speech characteristics, melody, harmony, rhythm, dynamics, effects, structure, and level of activation) were predicted from low-level audio signal features.

2.2.2. Music and signal processing

SOUND is an energy form, which is represented by periodic vibrations of environment molecules (SOUND WAVES). The waves are created by a vibrating sound source. The sound is described as MUSIC, if its sources are musical instruments. Their exhaustive classification was introduced in [213], and this work is still the most common after almost 100 years (only the last group of electrophones was added lately):

- IDIOPHONES produce sound by their natural vibration and are composed of a hard body. The examples are xylophone and triangle.
- MEMBRANOPHONES use a vibrating membrane as a sound source and contain different groups of drums (kettel drums, cylindrical drums, barrel drums, etc.)
- CHORDOPHONES are the instruments with vibrating strings, such as violins and pianos.
- AEROPHONES create sound by air vibrations, e.g., in the pipes of organ and wind instruments.
- ELECTROPHONES build sound waves with the help of loudspeakers and can be distinguished into the two groups: conventional instruments with electric amplification (e-guitar) and newer instruments, which often have keys, such as synthesisers.

Each sine wave is characterised by its AMPLITUDE w_a and LENGTH, or PERIOD w_l. w_a measures the strength of the wave and relates to the environment pressure and sound loudness. PRESSURE is a physics term, which is defined as force divided by the area of its application. LOUDNESS corresponds to perceived pressure, or volume. w_l describes the periodicity, i.e. the time length of a single vibration. Wave FREQUENCY $w_f = \frac{1}{w_l}$ is the inverse of the length and describes how often the wave achieves the same stage of its period in a second (measured in Hz). w_f is closely related to PITCH, which enables the ordering of sounds from low to high. Pitch can be defined as "the frequency of a sine wave that is matched to the target sound by human listeners" ([81] in [100], p. 8), or in terms of music theory as an "attribute of sensation whose variation is associated with musical melodies" ([172], p. 2).

Another meaning of sound is "an auditory sensation in the ear" [185], p. 3. The way how we react to music originates from the interaction of several complex systems: from auditory periphery to intermediate auditory stage and the central nervous system [156]. The sound waves arrive at the outer ear, pass on through the tympanic membrane to the three bones of the middle ear, and are then transmitted to the cochlea in the inner ear, which activates the auditory nerve system. During this process, some frequencies are enhanced and other attenuated. As a consequence, some frequency ranges are perceived

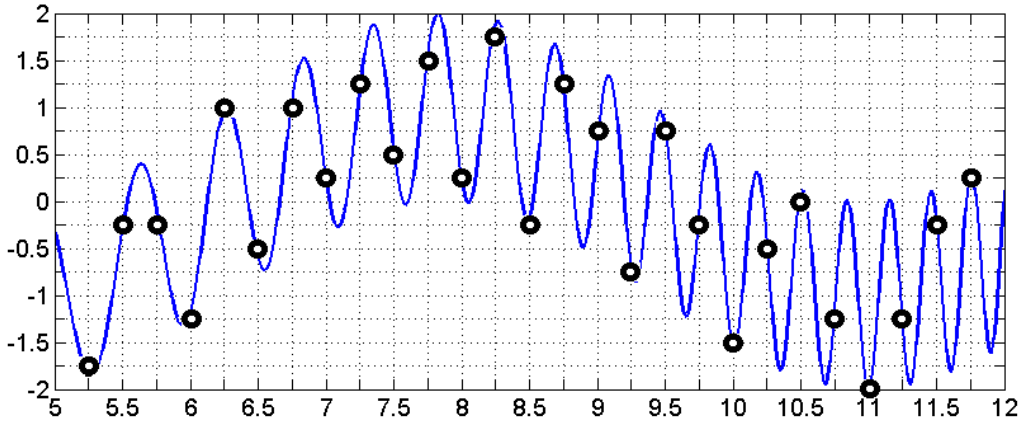

Figure 2.4.: Discretisation of the function $y = \sin x + \sin\left(x^2 - 2x\right)$ with the bit range of 4 bit. The original function is marked with a line, the saved values after discretisation with thick circles.

louder than other in spite of the same wave amplitudes. The best perceived range is between 2,000 Hz and 5,000 Hz [62]. An interesting fact is that the range of frequencies recognisable by a human ear is nine times greater than the range of frequencies perceived by an eye [185].

The recording techniques discussed in Section 2.1.2 store the sound as a time series (see Def. 2.1), or AUDIO SIGNAL, which maps equidistant time events to the corresponding environment pressure. Then, this pressure can be reproduced in almost the same quality as the original one. For the digital storage of audio two discretisation levels are necessary:

- SAMPLING defines the time points, at which the audio wave amplitudes are measured, for example each 0.25 points in Fig. 2.4. The sampling frequency f_s is measured in Hz. A standard audio CD uses $f_s = 44{,}100$ Hz.
- BIT RANGE, or DEPTH b_r, is the number of bits required for storing the wave amplitude levels. For example, 16 different values $\{-2, -1.75, ..., 1.75\}$ in Fig. 2.4 can be saved using 4 bits. A CD has a bit range of 16 bits, so that $2^{16} = 65{,}536$ discrete values are used for the measurement of amplitudes.

Larger f_s and b_r provide an exacter reproduction of sound, but have higher storage demands on the other side. Because human sound perception has natural limits between approximately 20 Hz and 20,000 Hz, it does not make sense to increase both parameters above a certain value.

Improper sound discretisation may lead to undesirable effects:

- The SHANNON THEOREM [212] claims that if the sound is digitised using a sampling rate f_s, only the waves with $w_f < \frac{f_s}{2}$ (NYQUIST FREQUENCY) can be recognised. An example of a wave with the higher frequency ($w_l = 2\pi$), which cannot be properly identified using $f_s = 2.5\pi$, is illustrated in Fig. 2.5. Here, this wave has the same discretisation values as a wave with $w_l = 5\pi$. This effect is called ALIASING. Then, it is necessary to apply an analogue LOW-PASS FILTER before digitalisation, which keeps only the waves with $w_f < \frac{f_s}{2}$ and removes the waves with $w_f \geq \frac{f_s}{2}$. For an

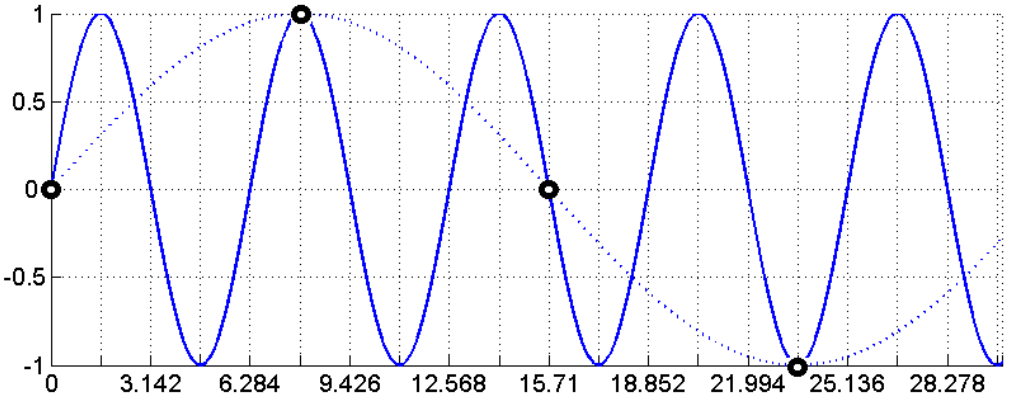

Figure 2.5.: The sine wave $y = \sin x$ (solid line) is wrongly recognised as $y = \sin(x/5)$ (dotted line), if the sampling is done each 2.5π points.

audio CD with $f_s = 44{,}100$ Hz, the frequencies up to 22,050 Hz can be saved. Higher frequencies are barely perceived by human ear[11].

- Another problem, the QUANTISATION ERROR, can occur by using of a too small bit range, such that the discretised signal does not properly represent the original one. The impact of this effect is particularly high for signals with very small amplitudes, so that the same discrete values are saved for different amplitudes.

The frequency representation of a sound, its SPECTRUM, can be estimated by the application of the Fourier transform, which maps any continuous periodic function to a sum of sine and cosine waves. The CONTINUOUS FOURIER TRANSFORM (CFT) is defined as follows:

$$CFT_{\mathbf{s}}(f^*) = \int_{-\infty}^{\infty} \mathbf{s}(t^*) \cdot (\cos 2\pi f^* t^* - j \sin 2\pi f^* t^*)\, dt^* = \int_{-\infty}^{\infty} \mathbf{s}(t^*) \cdot e^{-j2\pi f^* t^*}\, dt^*, \quad (2.1)$$

where t^* denotes the continuous time, $\mathbf{s}(t^*)$ the continuous time signal, f^* the continuous frequency, and $j = \sqrt{-1}$ is the imaginary unit.

Because the sound waves are discretised for storage as described above, the DISCRETE FOURIER TRANSFORM (DFT) is applied in praxis:

$$DFT_{\mathbf{s}}(f) = \sum_{t=0}^{B_f - 1} \mathbf{s}(t) \cdot e^{-j2\pi f t}. \quad (2.2)$$

Here, B_f corresponds to the number of spectrum frequency bins, t is the discrete time, $\mathbf{s}(t)$ is the discrete time signal, and f is the discrete frequency. $DFT_{\mathbf{s}}(f)$ is then the amplitude of a frequency bin $t \in \{0, ..., B_f - 1\}$. The Fourier transform can be estimated

[11] According to [87], p. 3, the sampling rate of 44,100 Hz was selected "simply for the reason that it was easier to remember" instead of 44,056 Hz – which was an alternative required for the compatibility with NTSC and PAL standards at that time.

significantly faster if B_f is a power of 2, for example using the Cooley-Tukey algorithm [33].

In modern western music, the sounds generated by musical instruments are organised in halftones. Each halftone has a certain perceived pitch. However, the halftone spectrum is more complex than a single wave: it is built from HARMONIC and NON-HARMONIC frequency components. The first group consists of a FUNDAMENTAL FREQUENCY, which is closely related to the pitch, and several nearly whole number multiples of the fundamental frequency, or OVERTONES. The non-harmonic frequencies are formed by the interaction of an instrument with its environment, e.g., strikes of violin bow or piano key. FORMANTS are the frequencies, which are especially amplified through an instrument body.

An OCTAVE is represented by 12 subsequent halftones (C, C♯ or D♭, D, D♯ or E♭, E, F, F♯ or G♭, G, G♯ or A♭, A, A♯ or B♭, and B), whose fundamental frequencies have a logarithmic distribution in the frequency domain. For simplicity reasons, we refer to the halftone fundamental frequencies as halftone pitches, or tone pitches. The whole number multiples of the original pitch are perceived as similar, and the same symbols are assigned to similar halftones of different octaves in the notation system. Most of modern pianos have the lowest halftone A0 and the highest C8. Whereas the tone pitches can be directly obtained from the score, they cannot be directly identified in a digitally saved audio signal. For this purpose, a frequency f can be mapped to the corresponding halftone in the SEMITONE SPECTRUM, where $\mathbf{p}$ denotes the bins [171]:

$$p(f) = \left\lceil 12 \log_2 \frac{f}{440} \right\rceil + 69. \tag{2.3}$$

Many different algorithms exist for the estimation of semitone spectrum amplitudes, or CHROMA. One of the common established is the pitch class profile (PCP) [68]:

$$PCP(p_w) = \sum_{\forall l: M(l)=p} ||X_l||^2, \text{ where} \tag{2.4}$$

$p_w \in \{0, 1, .., 11\}$ is a halftone pitch class (for building the *wrapped* semitone spectrum), and $||X_l||$ are the amplitudes of the spectrum bins, which correspond to the halftones of the class and are defined by:

$$M(l) = \begin{cases} -1 & \text{for } l = 0 \\ \left\lceil 12 \log_2 \left((f_s \cdot \frac{l}{B_f}) / f_{ref} \right) \right\rceil \bmod 12, & \text{for } l = 1, 2, ..., B_f/2 - 1 \end{cases}. \tag{2.5}$$

(f_{ref} is the reference frequency for $PCP(0)$).

The CEPSTRUM domain is defined by an inversed FFT (ICFT) logarithm [101]:

$$ICFT_{log(|CFT_{\mathbf{s}}(f^*)|)}(\tau) = \int_{-\infty}^{\infty} log\left(|CFT_{\mathbf{s}}(f^*)|\right) \cdot e^{j2\pi f \tau} df^*. \tag{2.6}$$

The cepstrum domain is used for the estimation of mel frequency cepstral coefficients (MFCCs), which were successfully applied for speech recognition [178], and are also used

in many music classification tasks.

The PHASE TRANSFORM for audio classification was proposed in [146], where the phase domain features provided a successful discrimination between music with a higher share of percussive impulses (rock and pop) and music with a lower share of percussive impulses (classical). A phase domain representation of the signal $\mathbf{d}$ with length D is defined as:

$$PD(e^{PD}, m^{PD}, \mathbf{d}) = \left\{ \mathbf{d}_i^{PD} | i = 1, ..., D - (m^{PD} - 1)e^{PD} \right\}, \text{ where} \tag{2.7}$$

$$\mathbf{d}_i^{PD} = \left(d(i), d(i + e^{PD}), d(i + 2e^{PD}), ..., d(i + (m^{PD} - 1)e^{PD}) \right), \text{ and} \tag{2.8}$$

e^{PD} is the delay in the phase space, m^{PD} is the dimensionality of the phase space, and $i \in \{1, ..., D\}$.

A common way to estimate audio features for a music piece is to apply WINDOWING in the corresponding domain, where each feature value is calculated from a window of a certain length, e.g., 512 samples. Some features (such as time structure and tempo characteristics) require large extraction frames. On the other side, many timbre and harmonic characteristics are calculated from small extraction frames not longer than the shortest note. The lowest frame size boundary for spectrum-related features is restricted by the Shannon theorem: for example, a frame of 512 samples from the signal with f_s = 44,100 Hz permits after DFT the estimation of amplitudes of 512 equally distributed frequency bins between 0 Hz and 22,050 Hz, so that the distance between the bins is equal to 43.07 Hz. This bin resolution produces larger errors especially for halftones of the lower octaves: e.g., C1 corresponds to 32.70 Hz and the next halftone C♯1 to 34.65 Hz).

The extraction frame sizes often correspond to the powers of 2, for reasons of efficient computation. For chroma and harmonic analysis, a frame size of 4,096 provides a good compromise [71].

Figure 2.6 provides examples of different feature domains. The upper subfigure shows the score of the first bars of Beethoven's "Für Elise". The second subfigure from the top depicts the corresponding time signal. The subfigure below illustrates the spectrum with frequency amplitudes up to 1,000 Hz. The bottom left subfigure shows the chroma discrete cosine transform-reduced log pitch for the first 9 notes, where the melody line E-D♯-E-D♯-E-B-D-C-A is indicated with the squares with darker red colours (high chroma amplitude) and the low chroma amplitudes are marked with darker blue colours. The bottom right subfigure plots the phase domain representation of the first bars.

2.2.3. Audio features

This section describes the features used for experimental studies of this thesis. Because of the major advantage of audio features that they can be always extracted from the corresponding MP3 song, we restricted the feature set used in this work to audio features only. The score is not always available for popular music pieces, and metadata are often incomplete or subjective. The exact feature lists with literature references are provided in Appendix A. In the following sections we briefly discuss characteristics of the feature groups and provide definitions for several features, which were implemented for this thesis.

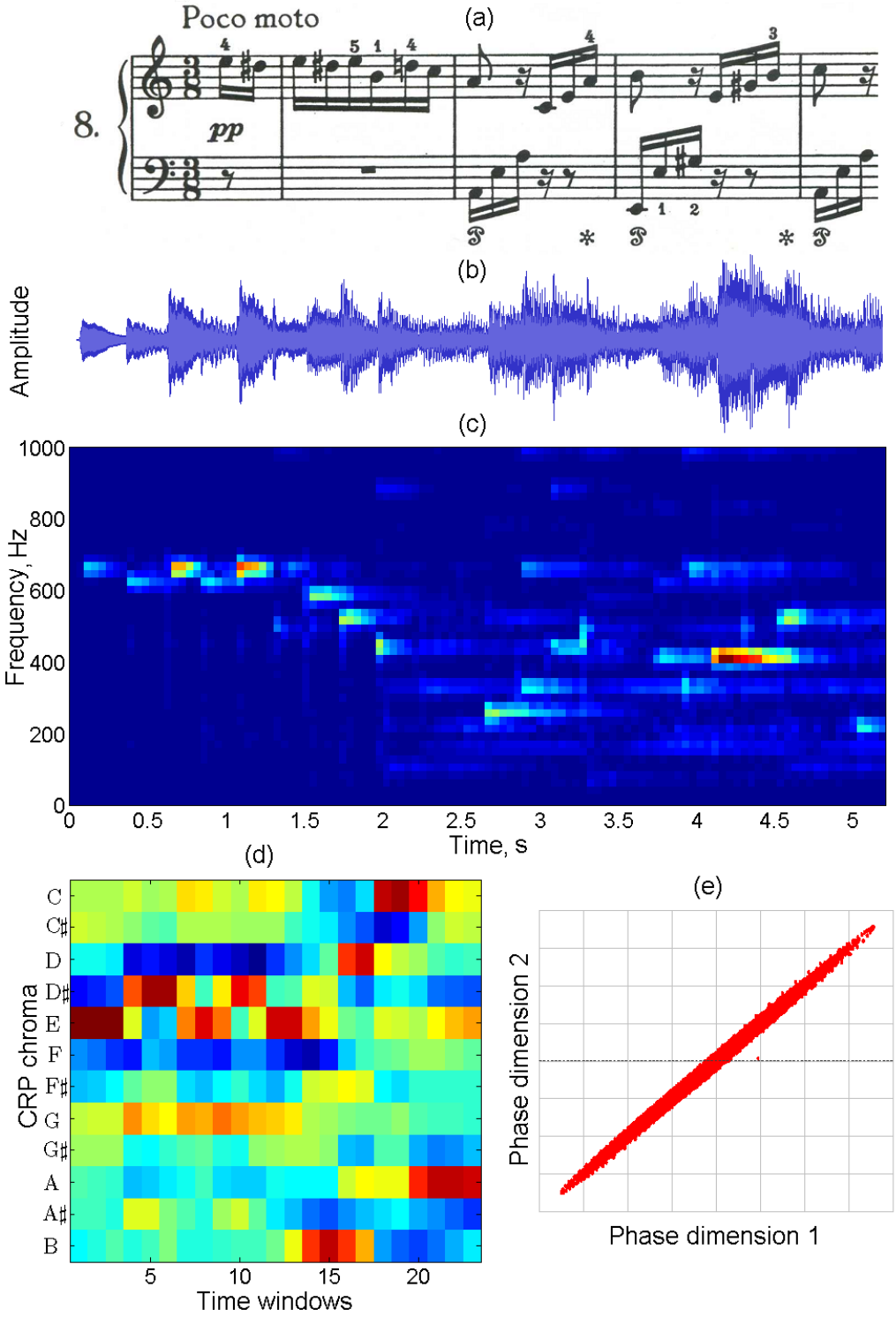

Figure 2.6.: Several representations and audio feature domains for Beethoven's - "Für Elise": (a) the score; (b) time domain; (c) spectrum domain; (d) chroma discrete cosine transform-reduced log pitch (CRP) [155]; (e) phase domain [146].

The Sections 2.2.3.1 to 2.2.3.3 adapt the categorisation of feature groups from [211], where three types of audio features were described: timbral, rhythmic, and pitch characteristics. We extend these groups to 'timbre and energy' (features estimated mostly from time, spectral, and cepstrum domains), 'chroma and harmony' (pitch-related, short-frame, and low-level characteristics, as well as high-level descriptors, e.g., a number of different chords), and 'tempo, rhythm, and structure' (long-frame, mostly high-level features, which describe the time structure of music). In Chapter 2.2.3.4 three high-level feature groups are mentioned, which are derived from the audio signal by application of the sliding feature selection (which is introduced in Section 3.3). The experimental studies for the creation of these high-level features are described in detail in Section 5.1.

Table 2.2 lists the software tools used for $\mathcal{FE}$. All of them are integrated as libraries or plugins into the Advanced MUSic Explorer (AMUSE) [221]. We developed this Java framework with the target to provide interfaces for various music classification tasks, as categorised in Section 2.1.3.

Table 2.2.: Software tools for $\mathcal{FE}$.

Name	**Reference**
AMUSE	[221]
Chroma Toolbox	[155]
jAudio	[138]
MIR Toolbox	[117]
NNLS Chroma and Chordino Vamp plugins	[135]
Yale	[147]

2.2.3.1. Timbre and energy

Timbre and energy features can be considered as low-level (see Section 2.2.1), and most of them are estimated from short extraction frames. TIMBRE is a characteristic, which makes the halftones of the same pitch and loudness sound differently, depending on the source instrument and the playing style. ENERGY features relate to the noisiness and loudness of an audio signal.

Table A.1 in Appendix A lists the feature names, literature references, extraction frame sizes in samples W_e (for mono signals with $f_s = 22,050$ Hz), numbers of feature dimensions, the software used for feature estimation, and the unique AMUSE feature IDs. Most of these features are described in our technical report [206] and the manual of the MIR Toolbox [115], in which references to further works are given.

It is possible to group these features by their extraction domain:

- TIME DOMAIN characteristics describe the audio signal time series, e.g., by its approximation with linear prediction coefficients or energy distribution. For example, 'low energy' compares the energy of a frame to the energy of the previous larger *analysis* window. Another commonly used and simple feature is the zero-crossing rate. It correlates with the noisiness of the signal, which in turn describes the timbre [211].

- SPECTRAL DOMAIN features correspond to the numerous statistics of the distribution of the frequency bin amplitudes: spectral centroid, crest factor, slope, kurtosis, flux, skewness, distances between spectral peaks, etc.
- CEPSTRAL DOMAIN descriptors consist of the several implementations of the mel frequency cepstral coefficients (MFCCs) and the cepstral modulation ratio regression (CMRARE) features [133], which describe the temporal cepstrum progress using a polynomial approximation.
- PHASE DOMAIN features are the average distance and the average angle in the phase domain. These features are well suited for the separation of classical music and popular genres with a higher percussion share [146].
- Finally, ERB AND BARK SCALE DOMAINS are motivated by the characteristics of human perception, where different frequency bands are sensed differently [151].

2.2.3.2. Chroma and harmony

HARMONY describes the relationship between simultaneously played tones (and is often described as the 'vertical' music component). If exactly two tones are played at the same time, they build an INTERVAL; three and more tones are characterised as CHORD. One of the central terms in music harmony is the CONSONANCE: consonant intervals are perceived as more complete and pleasing, whereas dissonant intervals are perceived as rough. The differences between consonant and dissonant sounds can be measured by mathematical, physical, physiological, and psychoacoustical aspects. However, it is difficult to provide an exact definition, in particular, because the comprehension of consonance altered over centuries. References to older and newer theories are provided in [144, 185].

Because the exact notes cannot be perfectly extracted from audio, the first step in the estimation of almost all audio harmonic characteristics is the transformation into the chroma domain. One of the simplest possibilities is to estimate the PCP, as defined in Equ. 2.4. The chroma-related harmonic characteristics are often not so precise as the score features. However, they build a bridge between signal processing methods and music theory and are essential when no score is available.

Chroma and harmony features listed in Table A.2, Appendix A, comprise low-level spectral characteristics as well as high-level music theory related harmonic descriptors. It can be roughly distinguished between chroma-based features, harmonic characteristics, and chord statistics. A semitone spectrum, which is estimated from the frequency bin amplitudes aggregated around the corresponding pitches, can be considered as low-level. On the other side, the characteristics of chords and musical keys can be referred to as high-level.

Several features were implemented for this study directly in AMUSE and are defined as follows:

- INTERVAL STRENGTHS FROM THE 10 HIGHEST SEMITONE VALUES: First, a semitone spectrum is estimated with NNLS Chroma [135], saving the amplitudes $SC(\mathbf{p})$ for the 85 different pitch levels. Then, the indices of the 10 highest values are sorted and saved in $\mathbf{p}_{10}$. The interval strengths $IS(k)$ $(k \in \{1, 2, ..., 12\})$ are calculated as

follows:

$$IS(k) = \sum_{\substack{i,j \in \mathbf{p}_{10} \\ |i-j|=k}} min\left(SC(i), SC(j)\right). \tag{2.9}$$

- Interval strengths from the semitone spectrum above 3/4 of its maximum value: If a part of simultaneously played tones is significantly louder than another part, the 10 strongest SC values may describe the fundamental frequencies, overtones and noisy components only from the louder tones. Therefore, another possibility to measure the interval strengths is to allow all values above a certain threshold to contribute to the interval estimation. Here, all semitone spectrum values above 3/4 of the maximum are used:

$$IST(k) = \sum_{\substack{SC(i),SC(j) > \frac{3}{4} \cdot SC(\mathbf{p}_{10}(1)) \\ |i-j|=k}} min\left(SC(i), SC(j)\right). \tag{2.10}$$

- Strengths of the CRP cooccurrences: Chroma discrete cosine transform-reduced log pitch (CRP) [155] is an enhanced chroma variation. It was developed especially for filtering out timbre sound characteristics, which are mostly captured by lower MFCCs. The strength of two cooccurrent values $CRP(i)$ and $CRP(j)$, $i, j \in \{1, 2, ..., 12\}$ is defined as:

$$CRP^S(i,j) = \frac{CRP(i) + CRP(j)}{2}. \tag{2.11}$$

  The estimation of all strengths between CRP values provides a raw description of interval strengths, and the overall number of dimensions is equal to $\frac{12 \cdot 11}{2} = 66$.

- Number of different chords and chord changes in 10 s: This feature is estimated from the chords, which were previously extracted by the Chordino Vamp plugin [135]. A frequent chord change does not necessarily correspond to a rich harmonic progression, since only a few different chords may be a part of the chord sequence.

- Shares of the most frequent 20, 40, and 60 per cent of chords with regard to their duration: Initially, the durations of each chord are summed up for each chord type, and the most frequent chords for the complete music piece are estimated. $\mathbf{c}_{20}$, $\mathbf{c}_{40}$ and $\mathbf{c}_{60}$ save the indices of the most frequent chords, which cover more than 20%, 40% and 60% of the song. Afterwards, the time shares of these most frequent chords are estimated for each extraction window:

$$CS(k) = \sum_{i \in \mathbf{c}_k} \frac{Ch_i}{W_e}, \tag{2.12}$$

  where $k \in \{20; 40; 60\}$, and Ch_i is the overall duration of the chord i in the extraction frame.

2.2.3.3. Tempo, rhythm, and structure

Table A.3 lists mostly high-level features, which describe the temporal music structure: tempo, rhythmic patterns, and segmentation characteristics. The extraction frame size $W_e = -1$ means that the feature was estimated from the complete song.

- TEMPO features consist of signal autocorrelation statistics, as well as perceived periodicities in music: the tatum is the shortest, and the beat is the strongest entity of the perceived repetitions [193]. The onset events mark the beginnings of the new notes and are often detected by changes in the energy distribution.
- RHYTHM describes groups of periodic entities, e.g., the repetitions of strong and weak accents in subsequent measures. Songs with the same tempo may have different rhythmic patterns, and the same rhythmic pattern may occur for music pieces with different tempi. An extended discussion of tempo and rhythmic properties is given in [190]. One possibility to measure rhythmic properties of the audio signal is to calculate the progress of loudness for different subbands (fluctuation patterns) [167]. Rhythmic clarity describes how easily music listeners may perceive the periodic impulses [116].
- The 3-dimensional SEGMENTATION characteristics feature is based on the method from [169], which outputs a segment sequence with start and end times of each segment. The segment descriptions (such as 'bridge' or 'chorus') are predicted by hidden Markov models from the training set. The number of segment changes, the number of different segments, and the relative share of different segments are stored.

2.2.3.4. High-level descriptors from classification models

A large part of the high-level audio features (Tables A.4, A.5 and A.6) was built by the application of classification models trained from other, mostly low-level, characteristics, where the target label was equal to the high-level descriptor. Multi-objective feature selection was applied for model optimisation (see Section 3.2.2), and some of the high-level features were used as input features for the prediction of other high-level characteristics, as proposed in the concept of the sliding feature selection (see Section 3.3). These features describe instrumentation, vocal characteristics, harmony and melody, moods, etc. For the details about these features, please see below (Sections 5.1.1 to 5.1.3).

2.3. Feature processing

The feature processing task $\mathcal{FP}$ is defined in Def. 2.4. Its target is to convert original feature matrices to classification instances for training or classification. It should be mentioned that in many actual music data analysis studies more attention is paid to the development of new features or classification methods, rather than to feature processing. However, this intermediate step also plays a relevant role. Improper feature aggregation may lead to a significant decrease of the classification quality. Furthermore, a strong reduction of the feature matrix provides faster training and categorisation, and also saves vast amounts of disc space and reduces memory demands.

Let $(\mathbf{X}^*(1), ..., \mathbf{X}^*(F^*))$ be the original feature matrices for all audio features $1, ..., F^*$. Different features are extracted from frames of different sizes: e.g., the spectral centroid can be estimated from the non-overlapping frames of 23 ms, tempo from frames of 10 s and the song duration from the complete audio recording. In that case it is not possible to create a proper feature matrix for a complete song: for example, if the song duration is equal to 45 s, the tempo is not estimated for the last 5 s. A more serious problem is that many $\mathcal{FP}$ methods cannot be applied directly on vectors of different dimensions.

Before any further processing methods are started, we apply the following simple method for matrix harmonisation, which is illustrated in Figure 2.7 for four example features. First of all, the smallest extraction frame size W_{min} is estimated across all raw features (feature X_1). The time dimension of the harmonised matrix $\mathbf{X}^H$ is set to the whole number of the extraction windows of this length, which fit into the complete song[12]. For an example from the last paragraph, it would be equal to $\left\lfloor \frac{45s}{23ms} \right\rfloor = 1,956$. For all features with larger extraction windows, their values are assigned to several frames of length W_{min}, which are contained in the original larger extraction window. For each larger extraction window j of the feature i, the latest corresponding new small frame is calculated as $\left\lceil j \cdot \frac{W_e(i)}{W_{min}} \right\rceil$. If the last small frames do not correspond to any large frame (it holds for X_3 in Fig. 2.7), the values are set to NaN (not a number).

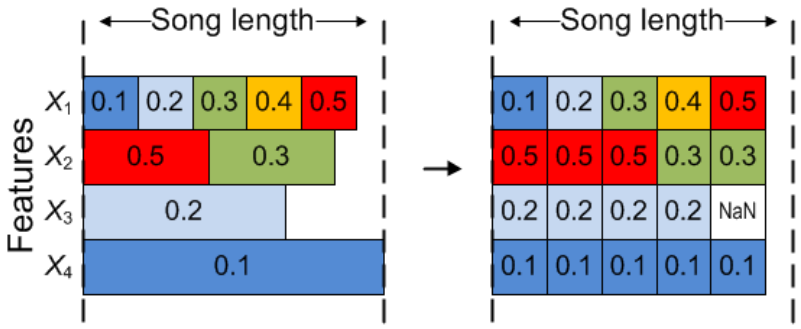

Figure 2.7.: Creation of the harmonised raw feature matrix $\mathbf{X}^H$.

After $\mathbf{X}^H$ is created, several possibilities to operate on the matrix dimensions exist, as sketched in Fig. 2.8:

- PREPROCESSING methods, such as normalisation, prepare the data for further algorithms and do not change the matrix dimensionality (a). We describe the most relevant methods in Section 2.3.1.
- Methods for PROCESSING OF FEATURE DIMENSION mainly operate on rows in the feature matrix. STATISTICAL PROCESSING OF FEATURES does not change the number of features and transforms the feature domains, e.g., by principal component analysis (PCA) (a). FEATURE CONSTRUCTION techniques create new features from existing ones and increase the feature dimensionality, for example by an estimation of a linear combination of several features (b). FEATURE SELECTION, e.g., by retaining the most relevant principal components after PCA or the least correlated features in

[12] Because the first step of our experiments is always the harmonisation of the feature matrix, we denote the output matrix as $\mathbf{X}^H$. For simplicity reasons, we denote as $\mathbf{X}'$ *all* matrices, which are output by any further $\mathcal{FP}$ step.

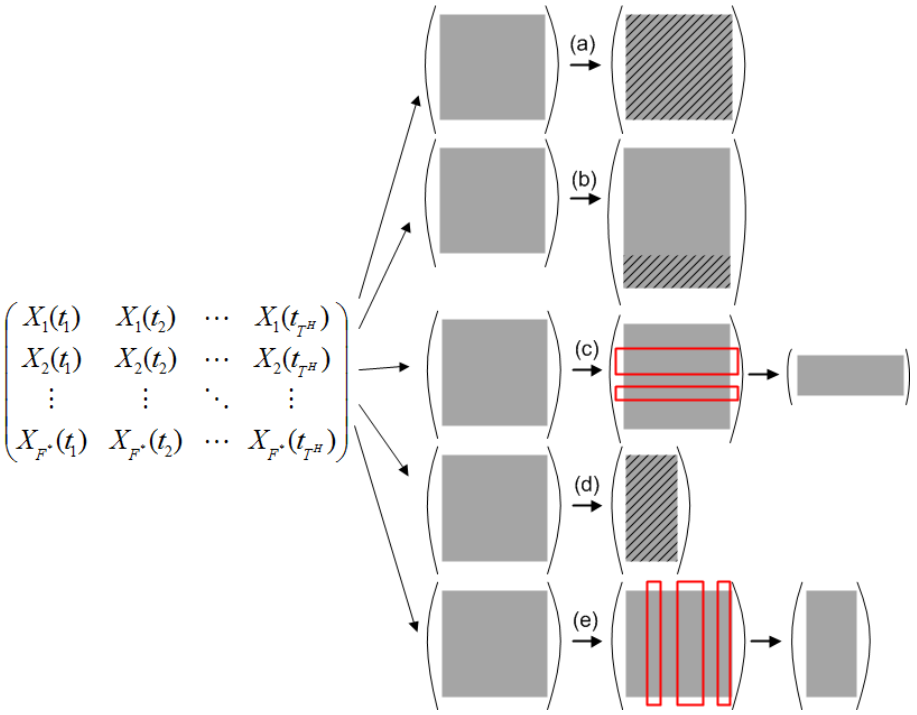

Figure 2.8.: Impact of different $\mathcal{FP}$ methods on the dimensionality of the feature matrix. Changed feature values are marked with dashed matrix regions. Methods, which select certain feature dimensions (c) or time intervals (e) are marked with red rectangles.

$\mathbf{X}'$, reduces the feature dimensionality without feature domain transforms (c). Several algorithms for processing of feature dimension are briefly mentioned in Section 2.3.2. Since multi-objective feature selection was an essential part of this thesis, it is described in detail in Chapter 3.

- Methods for PROCESSING OF TIME DIMENSION focus on columns in the feature matrix. STATISTICAL AGGREGATION OF FEATURES handles music as time series and calculates, e.g., the autoregression coefficients or estimates Gaussian models of the feature distribution (d). TIME REDUCTION BASED ON MUSICAL EVENTS selects features from certain frames based on music temporal and structural characteristics, for example, saving only features from extraction frames between beat events or from representative music segments, such as the chorus (e)[13]. One focus of our work was to integrate the knowledge of musical events into $\mathcal{FP}$; the corresponding methods for processing of time dimension are discussed in Section 2.3.3.

Although it is possible to significantly increase the dimensionality of $\mathbf{X}'$ (for example by estimating several derivations for each feature and keeping all extraction frames), it is indeed desirable to keep the matrix dimensions as small as possible. This holds not only for the number of features (too many noisy and irrelevant features may overwhelm the classification algorithms, as discussed later), but also for the time dimension: music

[13]Some of the methods correspond to a combination of (e) and (b): e.g., if the feature values related to different time events are used to build several new feature dimensions.

pieces usually consist of several similar repetitive parts, so that it is not required to train categorisation models from the complete songs.

Finally, the last $\mathcal{FP}$ task, BUILDING OF CLASSIFICATION FRAMES, is to identify concrete classification instances, which should be categorised or used for training – for example by averaging the feature values for time intervals of 5 s. This procedure is addressed in Section 2.3.4.

2.3.1. Preprocessing

The target of preprocessing is to prepare the numerical characteristics for successful classification, which may be limited by some constraints: for example, a classifier may not handle missing feature values or only process categorical features. Some works also refer to the more complex data reduction methods as preprocessing [22]. According to the categorisation of $\mathcal{FP}$ methods in the previous section, we limit the definition of preprocessing within the scope of this thesis to those methods, which do not change the dimensionality of the feature matrix and have one or several of the following objectives:

- STANDARDISATION is necessary, if the related features are extracted from the same domain, but are differently scaled. For example, time intervals can be measured in minutes, seconds, milliseconds, but also in samples.
- HANDLING OF MISSING AND NON-DEFINED VALUES: For data mining in general, handling of the missing features, which could not be properly extracted by an experiment or are simply not available, is a well-known problem. It is addressed in the corresponding literature, see, e.g., [131]. In our work, we use only audio features, which are previously extracted from each music piece. Therefore, there are *no* missing values (this situation may change quickly, when metadata features are integrated). However, there are several characteristics, which are not defined for certain extraction frames. As an example, the feature 'low energy' estimates the share of the root mean square (RMS) energy amplitudes, which are below the average RMS from a larger analysis window *before* the extraction frame. The jAudio default implementation uses an analysis frame of 100 windows, so that it is not possible to extract the low energy for the first 100 frames. Other examples are the amplitudes of the 2nd to 5th spectral peaks, which cannot be estimated, if only a single spectral peak exists. Further, some extraction frames with 'not a number' values are artificially created through the raw feature matrix harmonisation, as discussed in the previous section and illustrated in Fig. 2.7. Many methods are available to handle the missing values – for example, elimination of all instances with such characteristics or substitution by another value, e.g., zero, the mean, or the median across all values of this feature. In this work, the median is used. We use an abbreviation 'NaN' (not a number) for both missing and non-defined values.
- NORMALISATION makes compatible the differences between features, if they have very different definition areas. It is necessary especially for classification methods, which handle all feature dimensions in a similar way, e.g., by estimating the Euclidean distance between feature vectors. For example, zero crossing rate values are limited to $[0; 1]$ by their definition, whereas spectral peak positions belong to $[0; 22,050]$ for $f_s = 44,100$ Hz. In our work, we apply a so-called 0-1 normalisation [120]. Here, the

experimental minimum X_i^{min} and maximum X_i^{max} are estimated for each feature X_i from a large number of songs. Then, each value $X_i(t_j)$ (j is the extraction frame number) is replaced by $X_i'(t_j)$:

$$X_i'(t_j) = \frac{X_i(t_j) - X_i^{min}}{X_i^{max} - X_i^{min}}. \tag{2.13}$$

- Domain transforms change the feature domains according to the requirements of classification algorithms, for example by a mapping of nominal characteristics to real values. If a classifier processes only nominal characteristics, the features with continuous real values have to be discretised and mapped to nominal values.

2.3.2. Processing of feature dimension

As introduced in Section 2.3, a method operating on the feature dimension may either reduce, increase, or leave the number of features unchanged.

Statistical processing of features transforms the feature dimensions with the aim to enhance the classification quality or to reduce the number of features in the next step, retaining the most relevant characteristics. Principal component analysis (PCA) [92] is one of the well-established algorithms. It creates new orthogonal feature dimensions, which maximise the variance of the feature values, so that some of the new dimensions with smaller variances can be later discarded. Another method is the linear discriminant analysis (LDA) [4], which also takes into account classification labels and transforms feature dimensions in a way that the separability between different classes is maximised.

Feature selection aims at the decrease of the number of features by a feature subset evaluation according to one or several criteria. Feature selection is discussed in detail in Chapter 3. It can be applied together with other methods – for example generating new features as derivations of others and then selecting a small part of the old and new characteristics, which are less correlated. Another commonly used approach is the selection of a certain number of principal components after PCA.

Feature construction, or generation, adds new feature dimensions. One of the common methods is the estimation of one or several derivations of a single feature, for example for delta MFCCs. This algorithm describes the feature time series and also belongs to the methods for processing of time dimension. A rather generic approach is to create new features by the application of some mathematical operators on the feature vectors, e.g., a product or a sum of the two feature vectors. This approach was applied in [146, 127].

The choice of the processing methods has a strong influence on the classification performance. Many statistical methods have the following disadvantages [57], so that we did not integrate them into our processing chain:

- The *interpretability* of feature sets (see Section 4.1.2) is completely lost, if the original high-level feature domains are transformed through statistical feature processing. Feature selection does not change the feature interpretability.
- The $\mathcal{FE}$ efforts are not decreased in many cases: even if a limited number of feature dimensions after PCA is selected, it is still necessary to extract *all* original features

for each new song added to the music collection.

2.3.3. Processing of time dimension

We can distinguish between general time or value series analysis methods, and methods, which incorporate knowledge about the temporal structure of music.

Time series mining comprises many different statistical algorithms, which describe and predict the time series. Because an exhaustive method overview is beyond the scope of this thesis (please refer to [20, 152]), we provide only several examples of these methods, which were already applied for music classification. See also [227] for a summary of literature related to *long-term* features and [1], section 'Temporal feature aggregation'.

- The simplest possibilities are to sample the data or to process the features from a certain interval: for example, the selection of 30 s from each song is applied in many related publications (often from the beginning or the middle of a song) [211, 2, 231, 141].
- Simple feature vector statistics can be described by the estimation of the first four moments (mean, standard deviation, skewness, and curtosis, applied in [112, 153]), or quantiles [41].
- Gaussian mixture models (GMMs) estimate several Gaussian distributions for time series characterisation, and also can be treated as a classification method [16]. They were applied for music classification in [21, 8, 211, 128].
- Autoregressive moving average models (ARMA models) describe the time series by the parameters of the linear model, in which each signal observation is estimated from its predecessors and successors [126]. Enhanced autoregressive statistics performed well for genre recognition in [141, 188].

The specific characteristic of music time series is that they are highly structured on several levels, and this knowledge can be explicitly integrated into temporal aggregation of features. Recalling the terms shortly introduced in Section 2.2.3.3, we may distinguish between the following levels of musical structure:

- Tatum is the shortest perceived entity of periodicity in a song (i.e. a shortest note).
- Beat is the strongest perceived entity of periodicity in a song. The distance between the beat events corresponds to the approximately whole number multiples of the distances between the tatum events.
- Onset marks the beginning of a new note. Onsets must not be coincident with beat and tatum events for several reasons: breaks in melodies, varying shares of fast notes, or because the beat and tatum grids are estimated from time windows, which are significantly larger than the bar length.
- Bar corresponds to the shortest grouping of notes with some similar periodic characteristics, in particular, the number of beats and the distribution of accents. The latter describes the rhythm, e.g., a 3/4 bar is typical for waltz and consists of three beats with the strongest accent at the first beat.

- SEGMENT is a larger music interval, which contains the bars grouped by some comprehensible high-level characteristic(s): the same tempo, rhythm, key and mode, instrumentation, etc. Especially for popular music, vocals and lyrics play a significant role in segment detection: verse and chorus are both segments with vocals, which are repeated several times. Whereas the verse text often varies, the chorus text remains the same and represents a highlight of a song, usually by a well memorable melody.

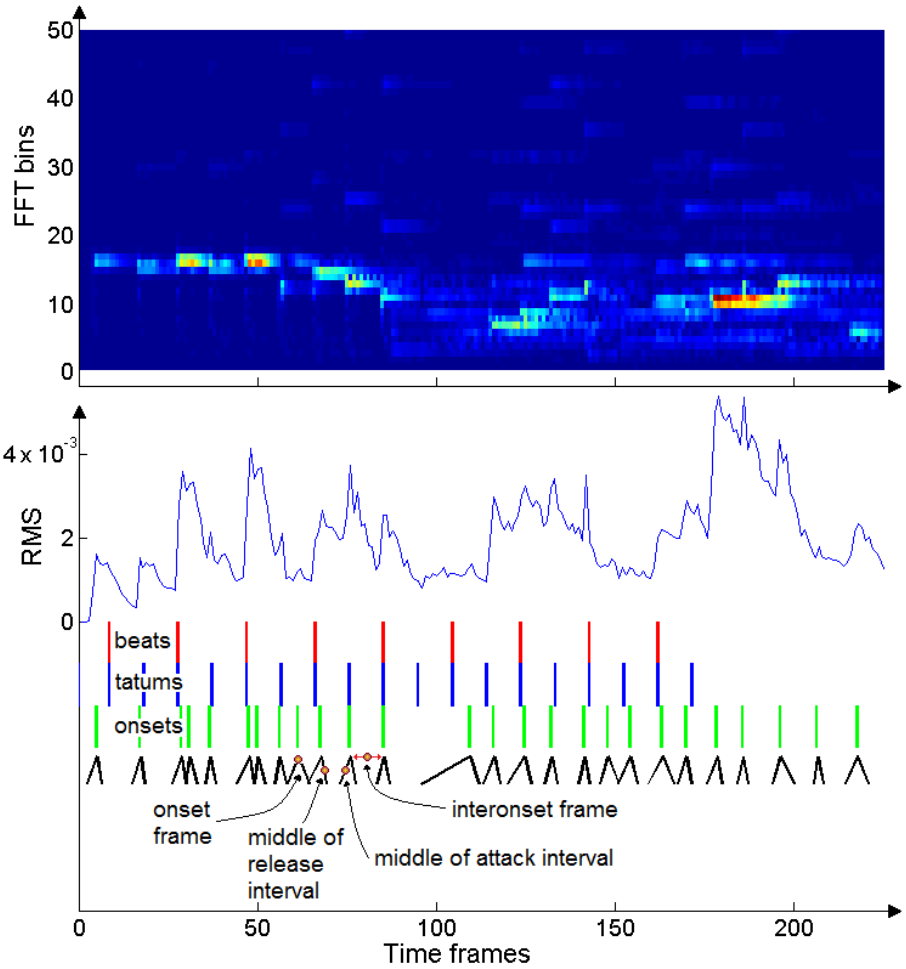

Figure 2.9.: Examples for beat, tatum and onset events, and attack and release intervals, which are extracted from Beethoven's - "Für Elise". Upper subfigure: amplitudes of the lowest 50 frequency bins. Subfigure below: the time signal RMS, which measures its energy.

Figure 2.9 provides examples for time events, which are extracted from the beginning of Beethoven's "Für Elise". The upper subfigure shows the amplitudes of the first 50 frequency bins, and the beginning melody line (E-D♯-E-D♯-E-B-D-C-A) is clearly seen (cf. also the discussion of Fig. 2.6). The plot below illustrates the RMS energy of the corresponding time signal, where the peaks in most cases correspond to the beginnings of the notes. The beat and tatum events are extracted with the algorithm from [56], and the onsets with the MIR Toolbox [117]. The slanted lines below the onset grid are the *attack*

and *release* intervals. They build a simplified model of the ATTACK-DECAY-SUSTAIN-RELEASE ENVELOPE (ADSR envelope), which describes the timbral characteristics of a musical tone [168]:

- ATTACK interval starts with the first occurrence of any tone-related frequencies. It is characterised by a strong energy increase and a high share of the non-harmonic frequency components, which are produced by the interaction between a music instrument and a musician, or another corpus (the strike of piano key, the noise of violin bow, etc.).
- DECAY is the subsequent short phase with decreasing energy.
- SUSTAIN is typically the longest interval with constant energy and a high share of the harmonic frequencies.
- RELEASE is characterised by decreasing energy of the vanishing sound.

Related to the simplified model of the ADSR envelope, the ATTACK-ONSET-RELEASE (AOR) envelope, the following short frames can be estimated as illustrated in Fig. 2.9:

- ONSET FRAMES: Feature extraction frames which contain an onset event.
- INTERONSET FRAMES: Feature extraction frames which are positioned exactly between two succeeding onsets).
- Frames which are positioned at the BEGINNINGS AND THE MIDDLES OF ATTACK INTERVALS.
- Frames which are positioned at the MIDDLES AND THE ENDS OF RELEASE INTERVALS.

The following methods for processing of time dimension were implemented and investigated in our studies, and most of these methods are compared in [222]. AOR-related features were integrated in the studies [219, 216], and structural complexity characteristics were used for the recognition of high-level features and genres (Sections 5.1.2 to 5.2):

- INTERVAL SELECTION with the length set to 30 s as in related publications (mentioned at the beginning of this section)[14]. Because popular songs usually consist of many different segments, which may be all important for genre and category prediction, the interval selection was not only done from the first 30 s of a song, but also from the middle and after the 1st minute, in the hope to capture at least the chorus as the most representative section.
- BEAT AND TATUM RELATED SELECTION: These short-frame features are saved only from the extraction windows positioned either at beat and tatum events, or in between these events. Aggregation around beats was proposed in [55].
- AOR-RELATED SELECTION works similarly, selecting the extraction frames from AOR-related events: for our studies we implemented the frame selection from the beginnings of attack intervals, the middles of attack intervals, onset events, the middles of release intervals, and the ends of release intervals.

[14] We are not aware of any study, which confirms the choice of 30 s by statistical means. In 2008, there was a discussion about "30 seconds" on the MUSIC-IR list. One opinion was that in some countries it was legal to distribute 30 s song excerpts. In another one it was suggested that the selection of 30 s from the middle was a good compromise to skip less representative segments, such as the intro.

- SAMPLING RELATED TO THE NUMBER OF EVENTS: The number of selected equidistant extraction frames is set as a factor of the corresponding number of time events in a song.
- STRUCTURE-RELATED SELECTION: At the beginning, the automatic song segmentation detects song segments with different low-level or high-level characteristics. In our studies, we use the method from [169]; for a general overview of automatic music structuring methods see [170]. Then, a limited number of larger *classification* frames (see the next section) is selected from each segment.
- STRUCTURAL COMPLEXITY is a temporal feature aggregation method, which is introduced in [136]. First, a set of features is selected, which describe some high-level characteristic, such as harmony or instrumentation. Let F^* be the number of all considered features, and $X_k(t_j)$ denote the value of feature k in frame j. For each feature extraction frame i, a number of N_f^{SC} preceding and N_f^{SC} succeeding frames is taken into account to measure the differences between the summary feature vector before (**wp**) and after (**ws**) the frame i by the Jenson-Shannon divergence $d_{JS}(\mathbf{wp}, \mathbf{ws})$:

$$d_{JS}(\mathbf{wp}, \mathbf{ws}) = \frac{d_{KL}(\mathbf{wp}, \frac{\mathbf{wp}+\mathbf{ws}}{2}) + d_{KL}(\mathbf{ws}, \frac{\mathbf{wp}+\mathbf{ws}}{2})}{2}, \tag{2.14}$$

where the Kullback-Leibler divergence is defined as follows:

$$d_{KL}(\mathbf{wp}, \mathbf{ws}) = \sum_{k=1}^{F^*} \mathbf{wp}_k \cdot \log\left(\frac{\mathbf{wp}_k}{\mathbf{ws}_k}\right), \text{ and} \tag{2.15}$$

$$\mathbf{wp}_k = \frac{1}{N_f^{SC}} \sum_{j=i-N_f^{SC}}^{i} X_k(t_j),\ \mathbf{ws}_k = \frac{1}{N_f^{SC}} \sum_{j=i+1}^{i+N_f^{SC}} X_k(t_j), k \in \{1, .., F^*\}. \tag{2.16}$$

In [136], the structural complexity was calculated for chroma, rhythm, and timbre features. We applied it for the 7 feature groups (chords, chroma, chroma related, harmony, instrumentation, tempo/rhythm, and timbre). Table A.7 provides the exact lists of the features, which are involved in the structural complexity estimation for each group. W_e and S_e describe the large extraction frames for the estimation of complexity (length and step size). The frame size in seconds was set to an integral multiple of 4, because we use classification frames with length $W_c = 4$ s for genre and style recognition. Several W_a values describe the lengths of the music interval, which are summarised before and after each structural complexity short extraction frame.

We compared a set of different time processing methods in [222]. Figure 2.10 presents the results, which are averaged for 28 music categories, 2 feature sets and two different classification frame sizes. The horizontal axis corresponds to a logarithm of the pruning rate (share of the extraction frames remained after the time processing, related to the original extraction frame number). A smaller pruning rate means a stronger data reduction. The vertical axis plots the average accuracy ranks across all categories. Smaller

ranks correspond to higher classification performance.

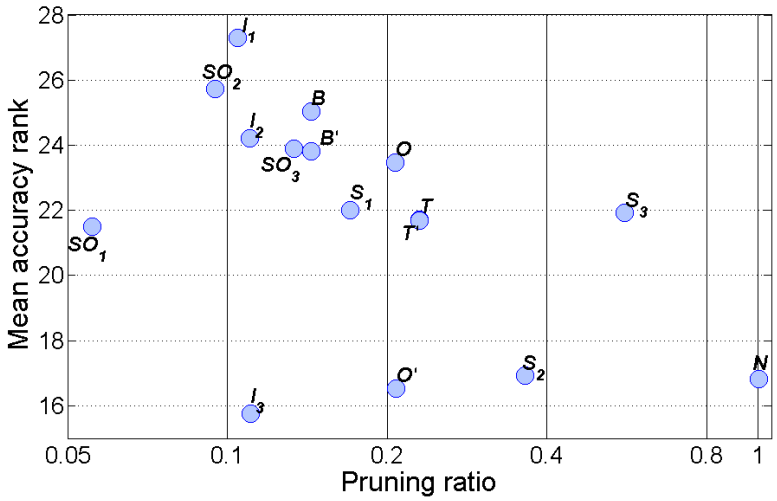

Figure 2.10.: Comparison of different time processing methods, adapted from [222]. N: baseline method without any time processing; T, B, O: selection of extraction frames only with tatum, beat and onset events; T', B', O': selection of extraction frames exactly in between tatum, beat and onset events; I_1: interval from the first 30 s of the song; I_2: interval from the middle of the song; I_3: 30 s interval after the 1st minute; S_1-S_3: selection of 1, 2 and 4 classification frames from previously extracted segments after [169]; SO_1-SO_3: combination of S_1-S_3 methods with interonset selection.

An interesting outcome of this study is that a rather simple interval selection method was the best one, namely the selection of 30 s after the 1st minute of a song. The two related methods, the selection of 30 s from the middle and from the beginning of the songs led to a strong decrease of the classification quality. The second best method was the selection of interonset frames. In general, methods, which selected frames between time events (T', B', O'), performed better than methods, which selected frames positioned at the corresponding events (T, B, O).

However, the results of the study must be treated with caution. Only the C4.5 classification method was tested, and the variance of the accuracy ranks was rather strong (the best rank was close to 16). No preceding feature selection was applied, so that the classification with other feature sets could provide other accuracies. Since the interonset frame selection performed second best and this method is motivated by music theory (the sound between the two notes tends to be stable and free of the noisy non-harmonic attack components), we integrated this method in the most following studies [15, 217, 218]. For instrument recognition, the ADSR envelope characteristics, among others also the non-harmonic components, are also relevant [123]. Therefore, we used a method, which selects the extraction frames from the middle of the attack intervals, onset frames and the middle of the release intervals [219, 216].

2.3.4. Building of classification frames

The final $\mathcal{FP}$ step is to prepare classification inputs by feature aggregation. For the recognition of high-level categories, the simplest possibility is to save some feature statistics for complete songs. This method may indeed perform well for very easily discriminable genres, such as classical music and pop music. In other cases, it is more promising to classify the feature vectors from shorter audio signal intervals. These intervals should be large enough to characterise the relevant musical events, and on the other side small enough not to mix the characteristics of different larger song segments, e.g., an intro, verse, or bridge.

The method to aggregate features for a classification frame W_c with a step size S_c can be selected from the algorithms discussed in Section 2.3.3. In our experiments, we calculate the mean and the standard deviation of each feature vector in the classification frame.

W_c could be itself a subject for optimisation: in one of our previous studies [223], we applied an evolutionary strategy for the simultaneous optimisation of feature selection and the length of classification frames. For the easiest of the three personal music categories to recognise, large frames around 24 s provided the smallest classification errors. For the two more complex tasks, the optimal frame size was below 5 s, and the classification performance decreased (approximately linearly) with an increasing frame size. The classification frames with $W_c < 4$ s also led to a rapid decrease of performance.

Another interesting related experimental result was outlined in studies [109, 70], as already mentioned in Section 2.2.1. Here, classification frames of only 400 ms and 250 ms were enough to provide reliable categorisations into artists and genres. However, it is not clear, if these small frames are indeed the best. Especially for more complex genres and styles, which depend on long-term high-level musical characteristics, such small frames may not perform well anymore. Further investigations in the future may help to provide clearer recommendations.

Based on the observations from above, it is hard to decide, which size of classification windows is optimal, when no extended knowledge about the categorisation task is available. Because

- we aimed at the aggregation of feature statistics from a large enough number of short feature extraction frames,
- too large classification frames had a tendency to increase the classification error[15], and
- we decided to concentrate rather on the optimisation of feature selection, and not on the optimisation of classification frame size and other parameters,

W_c was set by default to 4 s for the experiments of this thesis and in [217, 218]. 5 s frames were used in [15].

[15] Although the error for the easiest category in [223] was indeed slightly larger for frames around 4-6 s rather than for the optimal frames of approximately 24 s, the difference was not very high. Also, this category had strong similarities with the 'classic vs. pop' scenario, and in that case even feature aggregation over complete songs may provide acceptable results.

2.4. Classification

The main purpose of automatic classification is to organise data instances into classes (or categories) and to do it with an acceptable quality using an acceptable amount of resources. Most of classification scenarios can be handled by methods based on the three following concepts:

- Supervised classification learns from ground truth: previously labelled data, i.e. feature vectors mapped to categories. During the training of models, the characteristics of features are analysed to predict the categories, for example, by a linear separation in the high-dimensional feature domain. The definitions of supervised classification training tasks $\mathcal{CT}$ and the classification $\mathcal{C}$ were introduced in Section 2.1.3.
- Unsupervised classification does not start with any labelled data. The categories are created from scratch. The desired number of categories can be set before the classification or be identified by an algorithm itself.
- Semi-supervised classification builds models from both labelled and unlabelled data: this situation is closest to real-world scenarios, where a large amount of data is available, but only a small part of them are labelled. Especially, if new categories are defined over and over again (consider the prediction of personal music preferences), data labelling is a very cost-intensive procedure. However, the benefit of the integration of unlabelled data into the building of classification models depends strongly on the classification task. The labelled ground truth is still required, if the target categories should match the preferences of a music listener.

Regarding theoretical and practical issues of classification in data mining, exhaustive overviews with a focus at supervised and unsupervised approaches are provided, e.g., in [150, 45, 16, 4]. An overview of semi-supervised methods is given in [29]. A study, which examined the impact of the balance between labelled and unlabelled data, was investigated in [37].

In many music classification tasks, the target categories are well defined: genres, emotions, instrumentation, harmonic characteristics, etc. (see the discussion in Section 2.1.1). Therefore, supervised methods gained a widely accepted support. Many references to related studies (also considering unsupervised and semi-supervised methods) are provided for example in [1, 227, 120]. Unsupervised classification is suitable for the organisation of large music collections by self-organised maps [153, 102], but can be also successfully integrated into the recognition of high-level features, which were discussed in Section 2.2.1. For example, unsupervised methods were integrated into the recognition of the temporal structure [228] and the harmonic-related structure [95]. Semi-supervised music classification remains less investigated. However, some promising approaches were described in [119, 202], and it can be expected that the number of corresponding works will grow in future.

Another approach worth to mention, which is very closely related to classification, is similarity analysis. Here, no ground truth with several categories is given in the beginning. The goal is to measure the similarities of some songs with a given music piece. Different measures in the feature space can be taken into account for similarity estimation

[12]. One of the challenges is that the evaluation of algorithms is not so straightforward as for the direct classification with assigned labels [229].

The experiments in this thesis were restricted to supervised music classification scenarios. If we use the term classification within the scope of this thesis, we mean *supervised* classification. However, feature selection also makes sense for other situations, since too large feature sets with noisy, redundant, and irrelevant characteristics may diminish the quality of unsupervised approaches and similarity analysis.

A characterisation of classification methods can be done according to their inputs (classification instances) and outputs (labels). It can be distinguished between the following outputs:

- BINARY classification algorithms predict exactly two labels: an instance may completely belong to one class (positive instance) or not (negative instance), so that the classification target for an instance i can be mapped to zero or one: $y_P(i) \in \{0; 1\}$, recall Def. 2.6. The results of binary classifications for the individual song parts may be averaged for a complete song j, providing continuous class indicators as a result: $y_P(j) \in [0; 1]$.
- MULTI-CLASS methods classify instances into more than the two categories, where each classification instance belongs exclusively to exactly one of the C categories. The classification target can be then mapped to a discrete value between zero and one: $y_P(i) \in \left\{\frac{1}{C}, \frac{2}{C}, ..., 1\right\}$.
- MULTI-LABEL classification enables the assignment of several different labels to the same instance: $\mathbf{y}_P(i) \in \{0; 1\}^C$. This method is reasonable, if the data can be described by several independent categories, such as moods [208].
- STATISTICAL APPROACHES do not only output the labels directly, but also estimate the probabilities that an instance belongs to a category: $y_P(i) \in [0; 1]$.

These ways to produce different outputs can be transformed into each other: a multi-label problem may be converted into several single-label tasks by "problem transformation methods" [209]. Another example is the combination of several classifiers by AdaBoost for the probabilistic prediction of instance categories [82].

The satisfactory share of correctly classified instances depends on the classification approach: for example, a classification error of 45% means a very low performance for a binary classifier, if the instances are distributed equally across both classes. A *random guess* would have an expected error of 50%. For a multi-class task with 10 different categories, a random guess will have an expected error of 90% (under the assumption of an equal category distribution), so that an error of 45% corresponds to a higher classification performance.

Some classification algorithms do not accept all possible *inputs*. These restrictions should be addressed by proper feature (pre)processing (see Section 2.3). The most common issues are:

- HANDLING OF MISSING OR NON-DEFINED VALUES: Some classifiers expect only numerical feature values, so that 'not a number' entries must be substituted.

- Normalisation is necessary, if a distance measure, such as the Euclidean distance, is estimated for class separation, and all feature dimensions are similarly treated. This holds for example for the k-nearest neighbours classifier.

- Some methods process only features from categorical, discrete or continuous domains.

- Learning only by positives is a challenging task, if the ground truth only enlists examples, which belong to a given category [203]. Negative examples must be detected by the algorithm itself.

Sections 2.4.1 to 2.4.3 briefly describe classification methods, which have been used in our studies. The question may arise, *why* we have decided to select just these four algorithms (decision tree C4.5, random forest, naive Bayes and support vector machines). Our initial studies in music classification [205, 220] were done using C4.5, since it provided interpretable classification models and integrated feature pruning. The main focus of the subsequent studies was to investigate different evolutionary feature selection paradigms, starting with large initial sets of up-to-date audio features. We decided to *omit any classifier tuning*, avoiding large experiment computing times for this task, and to concentrate on feature design and feature selection. However, it was important to test the impact of feature selection on several classification methods with different underlying concepts and individual advantages and disadvantages. Starting with [157, 214, 217], we extended our set of classification methods[16]:

- Random forest (RF) is an ensemble method which creates a large number of unpruned decision trees from different random subsets of features. Classification is faster and often better compared to C4.5, but the classification models are not interpretable anymore. The operating methods of C4.5 and RF are discussed in Sect. 2.4.1.

- Naive Bayes (NB) is a probabilistic algorithm which estimates the conditional probabilities for predicted categories based on independent distributions of features. This algorithm is simple and very fast, the models have a high interpretability, but the classification quality is sometimes inferior to more complex methods. It is described in Sect. 2.4.2.

- Support vector machines (SVM) are state-of-the art methods in many classification tasks, and often achieve very good classification results combining original feature dimensions for a better class separation. On the other side, they are slow, are sensitive to parameter settings, and the models are less comprehensible. The basic SVM concepts are introduced in Sect. 2.4.3.

It should be kept in mind that a high complexity of a classification method does not help, if the features are poor and do not capture the characteristics of music categories – and on the other side well-designed features may lead to the high performance even with simple classifiers, as stated in [146].

[16] A further note is that according to the No Free Lunch Theorem [233, 45], no 'perfect' classifier exists.

2.4.1. Decision trees and random forest

A classification method, which generates perhaps the most interpretable models, is the DECISION TREE. Figure 2.11 illustrates an example from the study described later in Section 5.1.1. Two features, or *attributes*, are used here for the categorisation into chords with guitar (category 'Guitar') or chords without guitar (category 'NOT Guitar'). The starting tree node is called the *root*, and in each node a decision is made, if the corresponding feature value is above or below a certain threshold. In general, each node may have more than two successors, and also more complex queries are possible, e.g., 'if (feature 1 < 0.5) AND (feature 2 $= 0.4$), go to the left child node'. The tree *leaves* contain the instances, which are identified by the attribute queries on the path from the root to a leaf. The tree from Fig. 2.11 uses only two features and enables some *misclassifications* as a strategy against overfitting (see Def. 4.1 in Section 4.2). For example, a leaf on the left side of the tree contains 614 chords without guitar, which have 'envelope 1' values less than 0.057, but also 141 chords with guitar.

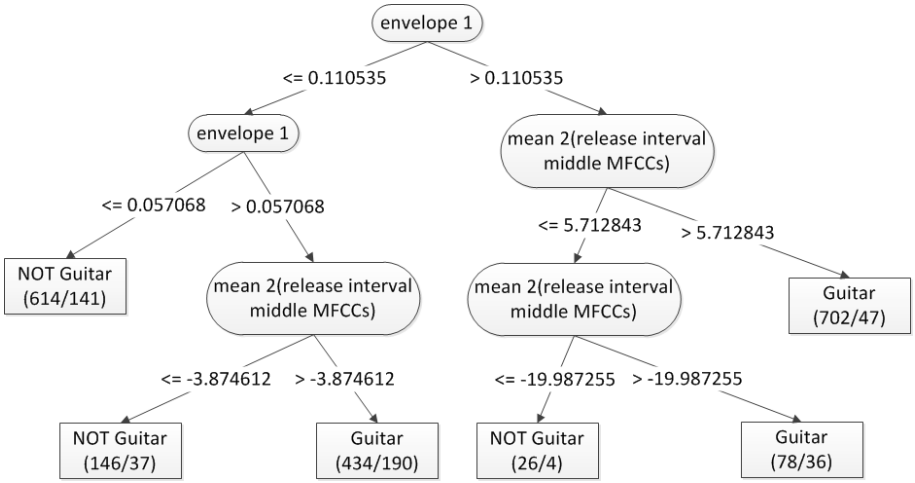

Figure 2.11.: A decision tree example.

One of the most critical decisions during tree construction is the choice of the attributes for the queries. The well-established algorithms ID3 and C4.5 [177] derive the concepts from information theory, investigated by Claude E. Shannon in [194]. In this theory, the value of a message is measured by the *minimal* number of trials, which are required to guess it. As an example, if a 4-letter word consists of exactly two 'A' and two 'B' symbols, the number of possible words is $\frac{4 \cdot 3}{2} = 6$: AABB, ABAB, ABBA, BABA, BAAB and BBAA. For guessing a word, at least three *binary* questions with a yes/no answer are required: for example, a first question could be: 'does a word belong to the left subgroup of the three words AABB, ABAB and ABBA?'. If a 4-letter word consists of exactly three 'C' and one 'D' symbol, we have only 4 possible words (CCCD, CCDC, CDCC and DCCC), and it is possible to guess any word by only two yes/no questions. The number of necessary questions is equal to $\log_2 |W|$, where $|W|$ is the number of different words.

Consider now that the symbols correspond to the categories of the T classification instances, which are organised by a subtree below a node. The node information content (the number of trials necessary for guessing a category of an instance below this node)

can then be measured by its ENTROPY $H(\mathbf{X})$:

$$H(\mathbf{X}) = -\sum_{i=1}^{C} \frac{freq(c_i)}{T} \cdot \log_2 \left(\frac{freq(c_i)}{T} \right), \text{ where} \tag{2.17}$$

C is the number of categories and $freq(c_i)$ is the number of instances from $\mathbf{X}$ which belong to category $c_i, i \in \{1, ..., C\}$.

The efficiency of candidate nodes can be measured by the INFORMATION GAIN $gain(\mathbf{X}, \mathcal{Q}^{DT})$, with the target to reduce the information content which is carried by a node with a query $\mathcal{Q}^{DT}$:

$$gain\left(\mathbf{X}, \mathcal{Q}^{DT}\right) = H(\mathbf{X}) - \sum_{j=1}^{k} \frac{|\mathbf{X}_j|}{T} \cdot H(\mathbf{X}_j), \text{ where} \tag{2.18}$$

$\mathbf{X}_j$ are the instances of k outcomes after the query $\mathcal{Q}^{DT}$.

Several further enhancements led to the development of the decision tree algorithm C4.5 (for details see [177]): handling of missing feature values, grouping of feature values, tree pruning, etc. Especially the last technique is very important, since too large trees increase the danger of *overfitting*: if a model describes the data perfectly, from which it has been trained, but is not suitable anymore for reasonable classification of other instances.

A forerunner of C4.5, the ID3 decision tree algorithm, incorporates REDUCED ERROR PRUNING, where a node is replaced by a leaf with the most frequent category of the succeeding instances. The performance of the original node and a leaf is measured by the classification error on a *validation* set. Because some of the classification instances must be reserved for this independent set, this restriction was removed by the RULE POST-PRUNING during the development of the C4.5. Here, a large and overfitted tree is built from the training data. Afterwards this tree is converted to a set of rules, which are partly pruned by sorting out rules with respect to their performance and its deviation on the training set.

A modification of the decision trees, the RANDOM FOREST (RF), builds an ensemble of unpruned trees and estimates the label output by majority voting [19]. During tree construction, for each tree node a number $m^{RF} \leq F$ of the random candidate features is selected and the best split is taken into account. The default RF algorithm uses $m^{RF} = \sqrt{F}$. The advantage of the RF is that it usually performs very well by averaging the tree outcomes. It is also fast, since no pruning is applied. However, the performance of the random forest suffers from a large number of noisy features because of the increasing share of irrelevant features from m^{RF} selected ones [82]. As we can see in the discussion of the experiments (Chapter 5), the RF method tends to increase its performance (as other classifiers), when the feature selection is previously applied. Another drawback is that the classification models are not interpretable anymore, compared to a single tree.

2.4.2. Naive Bayes

NAIVE BAYES (NB) is a classification method, which estimates the output label by its highest probability based on the feature distribution:

$$y_P = \underset{j \in \{1,...,C\}}{\arg\max}\ P\left(y_j | X_1, ..., X_F\right), \text{ where} \tag{2.19}$$

$P\left(A|B\right)$ is the *conditional* probability of the event A on the evidence of event B.

For the calculation of y_P, the BAYES THEOREM is applied:

$$P\left(y_j | X_1, ..., X_F\right) = \frac{P\left(X_1, ..., X_F | y_j\right) \cdot P(y_j)}{P(X_1, ..., X_F)}, \text{ where} \tag{2.20}$$

- $P\left(y_j | X_1, ..., X_F\right)$ is the POSTERIOR PROBABILITY of category y_j on the evidence of the feature distribution $X_1, ..., X_F$,
- $P\left(X_1, ..., X_F | y_j\right)$ is the CATEGORY LIKELIHOOD that the instance with the label y_j has a feature distribution $X_1, ..., X_F$,
- $P(y_j)$ is the PRIOR PROBABILITY to get an instance of the category y_j and
- $P(X_1, ..., X_F)$ is the EVIDENCE of the corresponding feature distribution.

Thus, we get:

$$y_P = \underset{j \in \{1,...,C\}}{\arg\max}\ \frac{P\left(X_1, ..., X_F | y_j\right) \cdot P(y_j)}{P(X_1, ..., X_F)} = \underset{j \in \{1,...,C\}}{\arg\max}\ P\left(X_1, ..., X_F | y_j\right) \cdot P(y_j), \tag{2.21}$$

since $P(X_1, ..., X_F)$ is not dependent on j.

$P(y_j)$ can be simply estimated as a fraction of the classification windows, which belong to the category j:

$$P(y_j) = \frac{1}{T} \sum_{\substack{i \in \{1,...,T\} \\ y_L(i) = \frac{j}{C}}} y_L(i) \cdot \frac{C}{j} \tag{2.22}$$

(we assume here, as introduced in Section 2.4, that the label of the classification instance i is set to a discrete value $\frac{j}{C}, j \in \{1, ..., C\}$, if it belongs to category j).

The estimation of $P\left(X_1, ..., X_F | y_j\right)$ is not so straightforward. However, NB makes the assumption that all features $X_1, ..., X_F$ are *independent* and *equally relevant* for the classification. Then:

$$P\left(X_1, ..., X_F | y_j\right) = \prod_{k=1}^{F} P(X_k | y_j) = \prod_{k=1}^{F} \frac{1}{\sqrt{2\pi}\widetilde{X}_k(j)} \cdot e^{\frac{(X_k - \overline{X}_k(j))^2}{2\widetilde{X}_k(j)^2}}. \tag{2.23}$$

The last term is estimated as a PROBABILITY DENSITY FUNCTION of the Gaussian distribution of feature X_k for all instances of category j, characterised by its mean value $\overline{X}_k(j)$ and the standard deviation $\widetilde{X}_k(j)$.

Although the assumption that the features are independent and equally relevant does not hold in reality, the performance of NB is often only slightly worse or even comparable to the performance of more complex classification methods, and the algorithm is very fast. It is obvious that NB may well benefit from feature selection, which removes irrelevant features. This is also confirmed by the studies described in Chapter 5.

2.4.3. Support vector machines

SUPPORT VECTOR MACHINES (SVM) belong to the group of *kernel* methods, which transform the original feature domain (referred to as *input space*) to a high-dimensional *feature space*, with the target to enable a linear discrimination between categories. We provide here only a short overview of the algorithm operation principle. A more comprehensive introduction with further references is provided for example in [237].

The original SVM assigns classification instances to the two classes: $y_P, y_L \in \{-1; 1\}$. For multi-class prediction, several binary SVM can be combined.

Let us assume that both categories are *linearly separable* as in the example in Fig. 2.12 (left subfigure), i.e. it is possible to draw a hyperplane, which separates the positive examples from the negative ones. This hyperplane can be in general defined by:

$$y(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + w_0. \tag{2.24}$$

The distance between $y(\mathbf{x})$ and the closest instances on both sides is called *margin*. The basic principle of the SVM is MARGIN MAXIMISATION, so that the distance between the instances of different classes separated by $y(\mathbf{x})$ is as large as possible. The target is now to find such weights $\mathbf{w}$ and a bias w_0, so that:

$$y(\mathbf{x}_i)\,(\mathbf{w} \cdot \mathbf{x}(i) + w_0) \geq 1, \text{ where} \tag{2.25}$$

$i \in \{1, ..., T\}$ are the indices of the classification instances.

Because the distance between the hyperplane and the feature vector $\mathbf{x}_i$ is equal to $\frac{|y(\mathbf{x})|}{||\mathbf{w}||}$, the maximisation of the margin can be solved by the minimisation of $||\mathbf{w}||$, and we can formulate this as a quadratic optimisation problem:

$$\min \frac{1}{2}||\mathbf{w}||^2 \text{ subject to Inequ. 2.25.} \tag{2.26}$$

This problem can be solved by a minimum search for the corresponding LAGRANGE FUNCTION ($\alpha_i^{SVM} \geq 0$ are the Lagrange multipliers):

$$L(\mathbf{w}, w_0, \alpha^{SVM}) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^{T} \alpha_i^{SVM} \left[y(\mathbf{x}_i)\,(\mathbf{w} \cdot \mathbf{x}(i) + w_0) - 1\right], \tag{2.27}$$

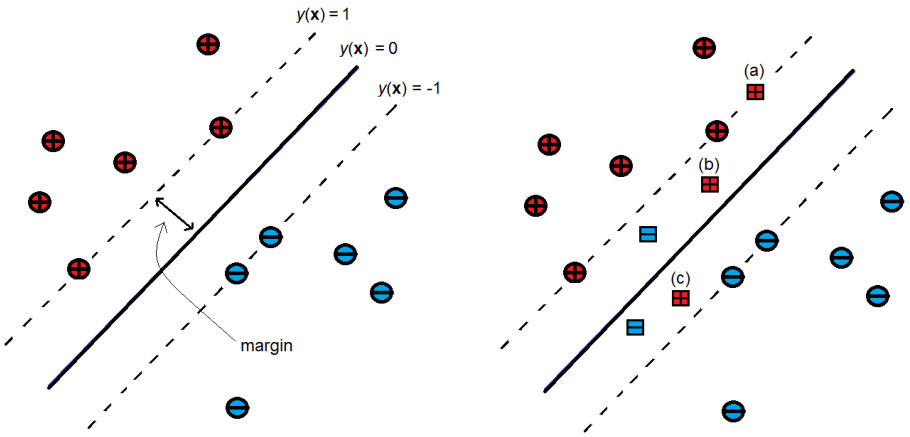

Figure 2.12.: Hard-margin (left subfigure) and soft-margin (right subfigure) maximisation. Positive instances are marked with circles with plus signs, negative instances with circles with minus signs. Instances, which are penalised by soft-margin maximisation, are marked with squares.

or the solution of the *dual problem*:

$$L(\mathbf{w}, w_0, \alpha^{SVM}) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_{i=1}^{T} \alpha_i^{SVM} y(\mathbf{x}_i)\mathbf{w}\cdot\mathbf{x}(i) + w_0 \sum_{i=1}^{T} \alpha_i^{SVM} y(\mathbf{x}_i) + \sum_{i=1}^{T} \alpha_i^{SVM}. \quad (2.28)$$

Because the data are not usually linearly separable, SVM apply the two following techniques:

- SOFT-MARGIN MAXIMISATION allows misclassifications, which are penalised by so-called *slack variables* $\xi^{SVM}(\mathbf{x}_i)$. The right subfigure of Fig. 2.12 illustrates several additional classification instances, marked with squares. In case (a), an instance lies on the margin line. It is classified correctly, and $\xi^{SVM}(\mathbf{x}_i) := 0$. If the instance is classified correctly, but is positioned within the margin, as it holds for instance (b), $0 < \xi^{SVM}(\mathbf{x}_i) \leq 1$. If an instance is not classified correctly, e.g., instance (c), $\xi^{SVM}(\mathbf{x}_i) > 1$.

- If the instances are not linearly separable in the original input space, it might be possible to transform them into a higher dimensional domain, where they can be linearly separated by a hyperplane. Equ. 2.24 can be then rewritten as:

$$y(\mathbf{x}) = \mathbf{w} \cdot \varphi^{SVM}(\mathbf{x}) + w_0, \text{ where} \quad (2.29)$$

  $\varphi^{SVM}(\mathbf{x})$ is a (nonlinear) mapping to the higher dimensional domain. Solving the dual problem in the feature space becomes more complex, because of the required estimation of the inner vector products. The KERNEL TRICK enables the efficient calculation of this inner product by the *kernel function* in the input space. The most often used kernels are linear, polynomial, radial basis and sigmoid.

The advantages of SVM are that no probability density estimation or complex pruning techniques are required, and the estimation of the new feature space can be done very efficiently using a kernel trick. On the other side, the performance depends on the tuning of hyperparameters, the method is often rather slow compared to other classifiers, and the models are less interpretable.

3. Feature Selection

3.1. Targets and methodology

FEATURE SELECTION (FS) is a method for processing of the feature dimension (see the categorisation in Section 2.3), which removes irrelevant, noisy, and redundant features from the classification instances. In formal terms, the task of FS is to find an optimal subset of features by minimisation of a relevance function, or evaluation metric m, e.g., the classification error[1]:

$$\theta^* = \arg\min_{\theta} \left[m\left(\mathbf{y}_L, \mathbf{y}_P, \Phi(\mathbf{X}', \theta)\right)\right], \text{ where} \tag{3.1}$$

$\Phi(\mathbf{X}', \theta)$ is the selected feature subset described by feature indices θ, $\mathbf{y}_L$ are the labelled category relationships, and $\mathbf{y}_P$ are the predicted category relationships. We denote here the feature matrix with the complete feature set by $\mathbf{X}'$, according to the note 12 in Section 2.3.

In case of multi-objective FS (multi-objective optimisation is introduced below in Section 3.2.2), O evaluation metrics are taken into account:

$$\theta^* = \arg\min_{\theta} \left[m_1\left(\mathbf{y}_L, \mathbf{y}_P, \Phi(\mathbf{X}', \theta)\right), ..., m_O\left(\mathbf{y}_L, \mathbf{y}_P, \Phi(\mathbf{X}', \theta)\right)\right]. \tag{3.2}$$

The features can be rated individually in terms of relevance and redundancy. Let X' be a feature *set* from the feature matrix $\mathbf{X}'$. One possible definition of strong relevance was proposed in [104], see also [240].

Definition 3.1 *A feature X_i is* RELEVANT*, iff its removal will decrease the performance of a Bayesian classifier:*

$$\begin{aligned} &P(y_P|y_L = y_P, X') < P(y_P|y_L = y_P, X' \setminus \{X_i\}) \text{ and} \\ &P(y_P|y_L \neq y_P, X') > P(y_P|y_L \neq y_P, X' \setminus \{X_i\}), \text{ so that in general} \end{aligned} \tag{3.3}$$

$$P(y_P|X') \neq P(y_P|X' \setminus \{X_i\}). \tag{3.4}$$

[1] We adapt the definition from [207] and assume that m should be minimised. If m should be maximised, it can be easily redefined for minimisation, see the note 9 in Section 2.1.3.

Definition 3.2 *A* REDUNDANT *feature X_i can be replaced without decrease of a Bayesian classifier's performance by at least one subset S^j, which does not contain X_i:*

$$\exists S^j \subseteq X', \{X_i\} \notin S^j : P(y_P|X') = P(y_P|S^j). \tag{3.5}$$

The reasons for automatic FS are:

- The classification quality is often improved after FS, since too large feature sets with many irrelevant and noisy characteristics typically overwhelm existing classification methods. It is confirmed by our previous and current experiments, described in [220, 223, 217, 219] and Sections 5.1 to 5.2, but follows also theoretical observations: with an increasing number of features the probability increases that features, which are indeed irrelevant, but have a certain degree of relevance for the training data set, become a part of a classification model. This leads to a decreased performance on other data sets. Also, for decision trees it is not advantageous to start with a too large feature set, which contains many irrelevant features[2].

- The manual design of features for a certain classification task may lead to the optimal performance, but is often far too expensive. The automatic selection of relevant features from a large original feature set, applied for each new category, does not require such high expert costs. Only the creation of ground truth (labelled instances) for new classification categories remains to be done.

- The classification becomes faster, if prediction models are trained with small feature sets of the most relevant features. In many classification scenarios and also in music classification it is a common situation that the training is done once per classification task (for example a categorisation of a private music collection into a genre), but the classification can be repeated over and over again (if new songs are added later to this collection).

- Storage demands are also usually reduced after FS. This holds for classification models (in particular, for classifiers with large models, such as k-nearest neighbours), but also for preprocessed features. It is also possible to create generic feature sets, which perform comparably well for several related classification tasks, as we could show in [216]. In that study, feature sets were evaluated and optimised w.r.t. their average performance on four different instrument categories.

- Feature selection may decrease the probability of highly overfitted models, where some of the features are identified as relevant by chance. However, this cannot be achieved by the application of FS only. It is essential to apply a proper organisation of data using independent holdout set(s) (see Section 4.2).

- The understanding of the dependency between relevant features and the corresponding categories can be significantly improved. This is especially reasonable for high-level features, as applied in this work: for example, it can be derived, which instrument or mood characteristics are strongly represented in the (sub)optimal feature subsets for the recognition of a genre or personal preferences.

[2] See [232] for a more detailed description. It is argued that the addition of a random attribute during the C4.5 classification caused typically a decrease of the classification performance between 5% and 10%. The impact of noisy features on the performance of random forest classification is discussed in [82].

[114] refers to FS as a search problem and provides a list with the "four basic issues that determine the nature of the heuristic search process":

- Starting point defines an initial set of features. The three typical possibilities are to start with either an empty feature set, the full set, or a part of the full set, for example using half of all features. In our experiments we use an initial feature rate parameter if_r, as introduced in Section 3.2.4.
- Search organisation describes the algorithm for feature selection. A list of several established strategies is provided below.
- Evaluation strategy defines one or several criteria for the evaluation of feature subsets. We discuss different groups of evaluation metrics in Section 4.1.
- Stopping criterion describes a condition, which should be fulfilled to stop the search, for example if the addition of new features does not bring any significant improvement of performance. In our experiments, we use a limited number of evolutionary algorithm generations as stopping criterion, see Section 3.2.4.

Some of the most common search strategies are (a categorisation is provided in [184]):

- Exhaustive evaluation of all possible feature combinations is a straightforward approach, which is very expensive with an increasing number of features, since $2^F - 1$ different sets should be evaluated for F features.
- Sequential selection methods either start with an empty feature set, adding features one-by-one according to some criterion, for example correlation with the label (forward selection), or removing features one-by-one, starting with the full feature set (backward selection). The first application of sequential selection was introduced in [130].
- Floating search, which was proposed in [175], enables the change of the feature subset size in both directions, where it is switched between the stages of forward and backward selection.
- Heuristic search works in a *non-deterministic* way with some integrated random component, so that the results are not the same for different repetitions of an experiment with the same starting conditions. We discuss a variant of heuristic search by evolutionary multi-objective algorithms in detail in Section 3.2.

The FS approaches can be in general categorised into three classes [75]:

- Filters are the oldest and fastest methods, which rate the features without any training of the classification models. Sequential selection by means of correlation is an example for a filter method.
- Wrappers evaluate feature subsets based on model training and classification, such as evolutionary feature selection, described in Section 3.2.
- Embedded methods integrate feature selection into a certain classification algorithm. The discussion of embedded methods is provided in [111].

For further reading about theoretical and practical issues of FS, we refer to [75].

3.2. Evolutionary feature selection

In the following sections, we describe in detail how multi-objective evolutionary algorithms can be applied for feature selection. Section 3.2.1 gives a short introduction into the basics and the history of evolutionary computation. The succeeding Section 3.2.2 introduces a formal definition of the multi-objective optimisation problem and describes how the solutions in a multi-objective space can be compared. In Section 3.2.3, we list the advantages of evolutionary FS, but also refer to several limitations of this approach. In Section 3.2.4, the working principle and the parameters of the multi-objective evolutionary algorithm SMS-EMOA, which has been applied for the multi-objective FS, are discussed.

3.2.1. Basics of evolutionary algorithms

Evolutionary algorithms (EA) are metaheuristics which simulate natural evolution processes in their operating method. An EA evolves a group of optimisation task solutions. They are together called population, whereas a single solution is referred to as individual (we use the words solutions and individuals as synonyms). Each individual describes the parameters of the corresponding optimisation problem solution in the decision space. The concrete numerical *representations* of individuals are also often referred to as belonging to the search space, which is the basis for evolutionary operators described below. The evaluation of individuals is done with respect to one or more objectives (also called fitness functions), which build the objective space. If exactly one objective is used for the evaluation of solutions, the EA is described as single-objective. If at least two objectives are optimised at the same time, the EA is multi-objective.

The basic principle of the evolutionary loop is illustrated in Fig. 3.1. It consists of the following steps:

- Population initialisation: The first optimisation task solutions are created, often in a random way.
- Fitness estimation: The population fitness value(s) in the objective space are calculated.
- Parent selection for breeding: One or more individuals are selected for the generation of offspring.
- Application of recombination, or the crossover operator: (a) new offspring is/are generated by some strategy, which derives the offspring position in the search space from the parent positions.
- The target of the mutation operator is to overcome local optima in the objective space and to enable a *stochastic exploration* by some strategy, which changes the offspring representation(s).
- Offspring fitness estimation: The offspring population fitness value(s) in the objective space are calculated.
- Selection of the next population: Based on the fitness values, some of the individuals are discarded.

- Check of the EXIT CONDITION: If some stopping criterion becomes true (a certain number of evolutionary loop iterations is achieved, the search progress slows down, etc.), the loop is finished and the final solutions are reported.

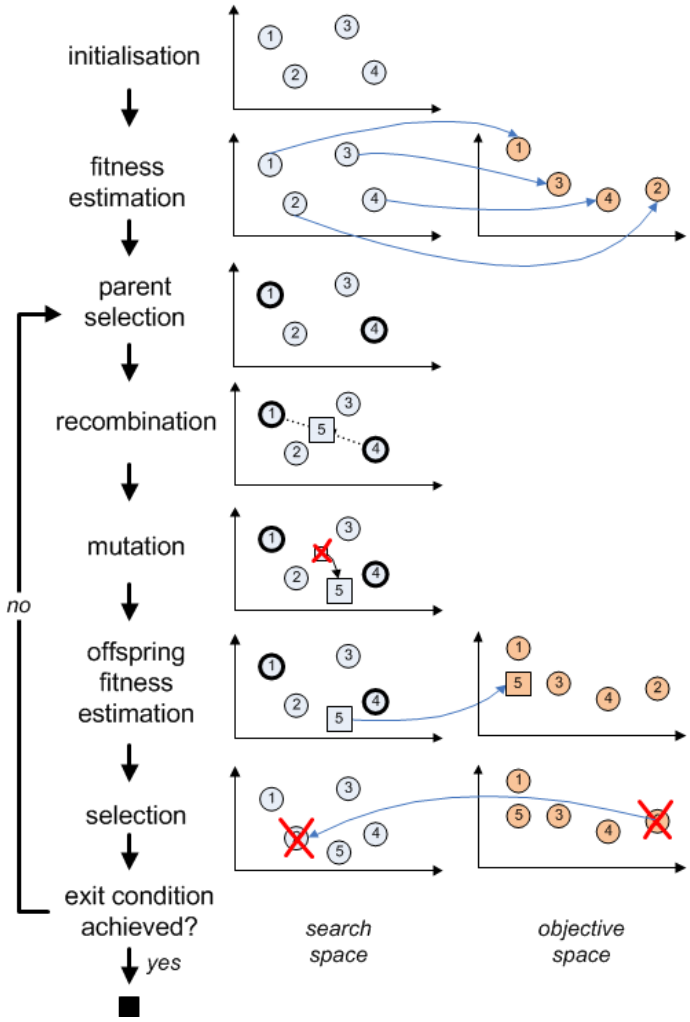

Figure 3.1.: The evolutionary loop for two-dimensional decision and objective spaces.

The population size is denoted by μ and the number of offspring by λ. A $(\mu+\lambda)$-EA selects the new population from both parents and offspring, after the offspring fitness evaluation step. A (μ, λ)-EA always replaces the parent population by the selected offspring, and in that case $\lambda \geq \mu$.

The first three groups of EA were developed independently of each other from the 1960s for more than a decade:

- EVOLUTIONARY STRATEGIES (ES) were designed in Germany by Ingo Rechenberg and Hans-Paul Schwefel [192].
- In USA, Lawrence Fogel, Al Owens, and Michael Walsh developed the concept of EVOLUTIONARY PROGRAMMING (EP) [64], and
- John Holland introduced GENETIC ALGORITHMS (GA) [83].

The roots of these methods go back to the proposal by Alan Turing for "the genetical or evolutionary search by which a combination of genes is looked for, the criterion being the survival value" in his essay "Intelligent Machines" from 1948 [88]. Though GA, EP and ES followed the same concepts, there were some differences especially in the earlier development stages, such as deterministic selection in ES or the emphasised application of crossover in GA.

Later, further evolution-inspired methods were developed. To name a few of the most prominent algorithms, genetic programming (GP) extends the GA concepts to the evolution of computer programs [105]. Particle swarm optimisation (PSO) integrates the characteristics of flocking behaviour, such as social interaction, movement velocity, and inertia, into an evolutionary algorithm [46]. Ant colony optimisation (ACO) simulates path creation in ant colonies, where the shortest paths have the strongest pheromone distribution [43]. From the 1990s, all these nature-inspired concepts are seen as a part of the evolutionary computation (EC) research field, and the algorithms are in general referred to as evolutionary algorithms (EA) [9].

Because of a strong increase in the number of studies which combine different approaches or introduce new enhancements and adaptations it is often not possible anymore to provide a clear boundary between the EA method groups. One of the already well-established concepts, the self-adaptation, enables the adjustment and control of algorithm parameters during the iteration progress [143]. The memetic algorithms integrate a deterministic local search procedure into metaheuristics for the systematic scan of the solution neighbourhoods [107]. The predator-prey approach simulates another nature phenomenon, the population of predators, which force the individuals (prey population) to explore new areas for survival [118].

Even if many improved algorithms outperformed the older concepts in extensive experimental studies, it should be kept in mind that according to the No Free Lunch Theorem (which has been also formulated for optimisation methods [234]), all algorithms have the same performance, if they are evaluated across all possible fitness functions. For further reading on EC, we recommend [187, 50].

3.2.2. Multi-objective optimisation

In many real world applications, among other classification scenarios, almost always several conflicting objectives play a role: algorithms with low classification errors require large computing times, the best models have a larger tendency to overfit against some data sets, or the achievement of an acceptable classification performance requires very high user efforts for ground truth creation (see also later the discussion in Section 4.1). In particular, if the objectives are not highly correlated with each other in all regions of the objective space, it is reasonable to search for the best compromise solutions. A multi-objective optimisation problem can be defined as provided in [241], p. 875[3]:

Definition 3.3 *A* MULTI-OBJECTIVE OPTIMISATION PROBLEM *(MOP) is a 5-tuple* $(\mathcal{X}, \mathcal{Z}, \mathbf{m}, \mathbf{g}, \leqslant)$*, where:*

- $\mathcal{X}$ *is the decision space,*

[3] As for feature selection definition, we assume that the objectives should be minimised.

- $\mathcal{Z} \in \mathbb{R}^O$ *is the objective space,*
- $\mathbf{m} = (m_1, ..., m_O)$ *is a vector-valued function of* O *objective functions* $m_i : \mathcal{X} \mapsto \mathbb{R}$*,*
- $\mathbf{g} = (g_1, ..., g_U)$ *is a vector-valued function of* U *constraint functions* $g_j : \mathcal{X} \mapsto \mathbb{R}$*, and*
- $\preccurlyeq \subseteq \mathcal{Z} \times \mathcal{Z}$ *is a binary relation on the objective space.*

The target is to find a decision vector $\mathbf{a} \in \mathcal{X}$*, so that:*

- $\forall j \in \{1, ..., U\} : g_j(\mathbf{a}) \leq 0$*, and*
- $\forall b \in \mathcal{X} : \mathbf{m}(\mathbf{b}) \preccurlyeq \mathbf{m}(\mathbf{a}) \Rightarrow \mathbf{m}(\mathbf{a}) \preccurlyeq \mathbf{m}(\mathbf{b})$

The comparison of solutions is based on the relation $\preccurlyeq$. Several of such relations were proposed, and one of the most conventional introduces the term Pareto dominance:

Definition 3.4 *A solution* $\mathbf{a} \in \mathcal{X}$ WEAKLY PARETO DOMINATES *the solution* $\mathbf{b} \in \mathcal{X}$ *(denoted by* $\mathbf{a} \preceq \mathbf{b}$*), iff:*

- $\forall i \in \{1, ..., O\} : m_i(\mathbf{a}) \leq m_i(\mathbf{b})$.

Definition 3.5 *A solution* $\mathbf{a} \in \mathcal{X}$ (STRONGLY) PARETO DOMINATES *the solution* $\mathbf{b} \in \mathcal{X}$ *(denoted by* $\mathbf{a} \prec \mathbf{b}$*), iff:*

- $\forall i \in \{1, ..., O\} : m_i(\mathbf{a}) \leq m_i(\mathbf{b})$ *and*
- $\exists k \in \{1, ..., O\} : m_k(\mathbf{a}) < m_k(\mathbf{b})$.

The Pareto front is built by the objective functions of those solutions, which are not dominated by any other solution:

Definition 3.6 *A solution* $\mathbf{a} \in \mathcal{X}$ *belongs to the* PARETO-OPTIMAL SET*, and* $\mathbf{m}(\mathbf{a})$ *belongs to the* PARETO FRONT $\mathcal{P}_f$*, iff*

- $\nexists \mathbf{b} \in \mathcal{X} : \mathbf{b} \prec \mathbf{a}$.

Because it is not always possible to find the Pareto front during a single algorithm run for complex optimisation problems, we speak of a NON-DOMINATED FRONT of solutions after the finished optimisation run.

Different criteria were proposed to evaluate the population of solutions (or the subset of the non-dominated solutions) which are output by a multi-objective optimisation algorithm. In [38], it is mentioned that multi-objective optimisation itself has the two targets: to find the solutions which are, firstly, as close as possible to the Pareto-optimal solutions and, secondly, as diverse as possible (the final number of solutions is limited to the population size, if an EA is applied for the solving of MOP). Therefore, it is distinguished between metrics which evaluate the closeness to the Pareto-optimal front and metrics which evaluate the diversity among the non-dominated solutions.

A metric introduced in [242], the HYPERVOLUME, or $\mathcal{S}$-metric, belongs to both groups and measures the united volume $vol(\cdot)$ of all hyperareas in the objective space, which are weakly dominated by the non-dominated set of solutions to be evaluated:

$$\mathcal{S}(\mathbf{a}_1, ..., \mathbf{a}_{P_{ND}}) = vol\left(\bigcup_{i=1}^{P_{ND}} [\mathbf{a}_i, \mathbf{r}]\right), \text{ where} \tag{3.6}$$

P_{ND} is the number of solutions in the non-dominated front, and $[\mathbf{a}_i, \mathbf{r}]$ is a hypercube spanned between solution $\mathbf{a}_i$ and a reference point $\mathbf{r}$, which is often positioned at the *worst possible* values of the respective objective functions ($\forall i : \mathbf{a}_i \prec \mathbf{r}$), cf. Fig. 3.3.

For lists with further evaluation criteria, see [38, 31].

3.2.3. Reasons for evolutionary multi-objective feature selection

Why are the evolutionary multi-objective algorithms (EMOA) well suited for feature selection?

- Except for very small feature sets, feature selection by an exhaustive search or by sequential strategies becomes very expensive. In [110], several approaches were compared, and a GA was recommended for LARGE FEATURE SETS with more than 100 variables. In general, the selection of optimal feature subsets is NP hard, as it was shown for related problems [104, 5]. EA were designed to solve such complex problems without any prior knowledge of the structure of the search space.
- It makes definitely sense to evaluate and optimise music classification in a multi-objective way, as discussed below in Section 4.1. In that case, it is not searched for a single solution, but for a front of non-dominated feature subsets. Therefore, POPULATION-BASED METAHEURISTICS, which evaluate a set of solutions in a single step, match these requirements very well, making EA unique for the MOP solving [38].
- The STOCHASTIC NATURE of EA helps to overcome local optima, and the shape or continuity of the Pareto front does not restrict the EMOA performance [31].
- Many common feature selection approaches estimate the INDIVIDUAL FEATURE RANKING, for example on the basis of the correlation with the label. However, it is possible that two or more features, which are irrelevant by themselves, are relevant in their combination [75], as illustrated in Fig. 3.2 (a,b). Furthermore, in [104] it was shown, that relevance does not imply optimality, and, on the other side, optimality does not imply relevance. The exploration of the feature space by EA operators is not biased towards any individual feature rankings.
- The search for REDUNDANT FEATURES (see Def. 3.2), which should be removed during FS, may be also dangerous, if this is done according to correlation. Fig. 3.2 (c,d) presents examples with two correlated features, which are differently relevant for a linear class separation. In subfigure (c), both features are not redundant and required for the linear separation. In subfigure (d), the combination of both features

does not increase the relevance, and they are more redundant. The individual dimensions of these features show the same separability, as shown by the projections near the axes.

In case of multi-objective evaluation, it may be even arguable only to concentrate on the removal of redundant features, because a part of the redundant features may be cheaper to extract (evaluation by costs) or more interpretable for a better understanding of the classification models (evaluation by interpretability).

- Compared to embedded FS methods, the evolutionary FS wrapper method is INDEPENDENT OF A CLASSIFIER, so that it is not required to redesign the FS method, if a new classification method should be used. This is useful when different classification methods are combined, as in our studies discussed in Chapter 5. Here, the combination of several different classification methods led to higher hypervolumes of non-dominated fronts.

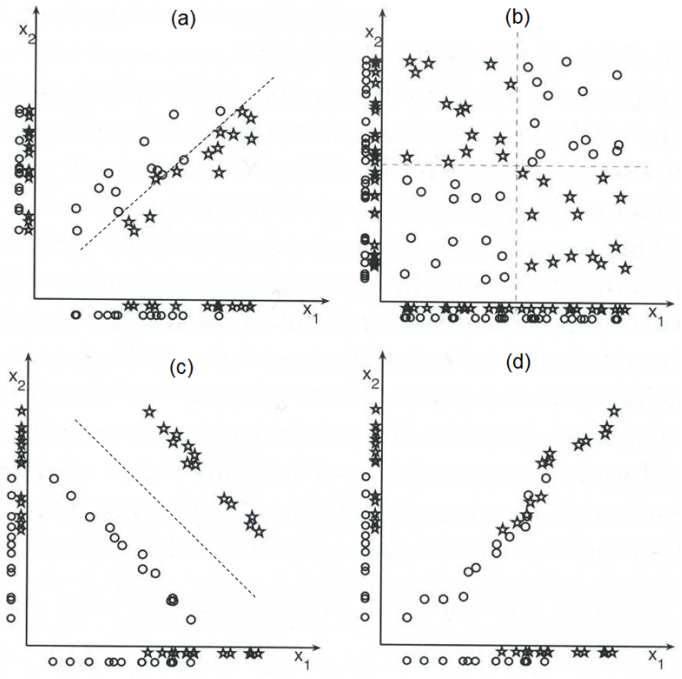

Figure 3.2.: Upper subfigures (a), (b): examples for features, which are individually irrelevant and relevant in combination. Lower subfigures (c), (d): examples for correlated features with different relevancies. Source: [75], p. 10.

It is also worth to mention the possible drawbacks of evolutionary-based FS, which are common for wrapper methods [104]:

- As all methods, which create and evaluate training models during the optimisation, EA require significantly larger runtimes than filters.
- Another limitation of wrapper methods is that the extensive training and validation of models may lead to overfitting. However, if enough labelled data instances are available, this can be avoided in most cases by the choice of an appropriate validation method, see Section 4.2.

- The last advantage from the list above (independency of a classifier) can also be interpreted as a drawback, depending on the classification scenario: the wrapper-based FS should be rerun for each classification method separately.

3.2.4. SMS-EMOA customisation for feature selection

The $\mathcal{S}$-metric selection evolutionary multi-objective algorithm (SMS-EMOA) was introduced in [54]. It is a $(\mu + 1)$-EA, which estimates hypervolume related metric for the individual selection, so that both the quality and the distribution of the solutions are evaluated. The original contribution of solution $\mathbf{a}_i$ to $\mathcal{S}$ of the complete population is measured as follows:

$$\Delta\mathcal{S}(\mathbf{a}_i) = \mathcal{S}(\mathbf{a}_1, ..., \mathbf{a}_{P_{ND}}) - \mathcal{S}(\mathbf{a}_1, ..., \mathbf{a}_{i-1}, \mathbf{a}_{i+1}, ..., \mathbf{a}_{P_{ND}}). \tag{3.7}$$

Figure 3.3 illustrates the difference between $\mathcal{S}$ and $\Delta\mathcal{S}(\mathbf{a}_i)$. The filled area in the left subfigure corresponds to the hyperarea covered by the population of solutions. The solutions are marked with small squares. In the right subfigure, the $\Delta\mathcal{S}(\mathbf{a}_i)$ areas correspond to the large filled rectangles.

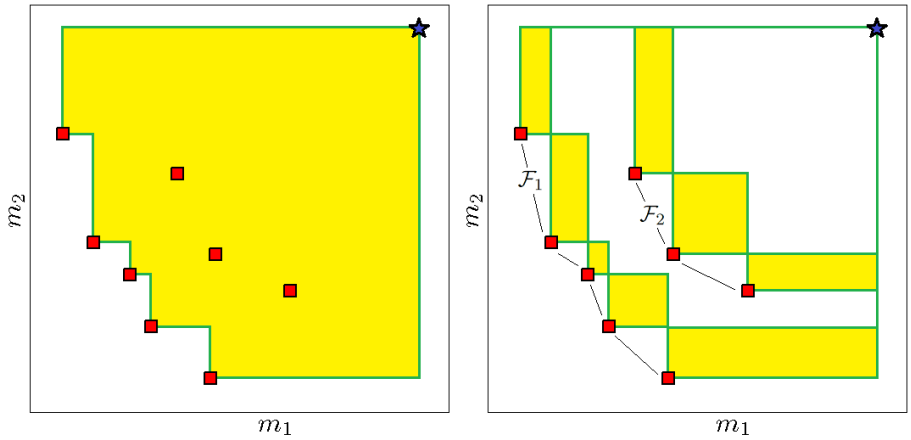

Figure 3.3.: Examples for the estimation of hypervolume (left) and $\Delta\mathcal{S}(\mathbf{a}_i)$ (right). The solutions in the objective space are marked with squares. The reference point is marked with an asterisk. The first two non-dominated fronts $\mathcal{F}_1, \mathcal{F}_2$ are marked with thin lines in the right subfigure.

SMS-EMOA applies the FAST NON-DOMINATED SORTING [39] before selection. The solution fronts are built according to the Pareto dominance relation. At the beginning, the individuals, which are not Pareto dominated by any other solution, are assigned to the first front. Then, the same procedure is applied on the remaining individuals, and it is repeated until the complete population is assigned to fronts. The right subfigure of Fig. 3.3 shows two fronts $\mathcal{F}_1, \mathcal{F}_2$, marked with thin lines.

The SMS-EMOA selection operator removes the individual j with the smallest $\Delta\mathcal{S}(\mathbf{a}_j)$ from the worst front. The advantage of this method is that with an increasing number of objectives the number of the non-comparable solutions according to the Pareto dominance

relation increases strongly, but it is still possible to estimate $\Delta\mathcal{S}(\mathbf{a}_j)$ for the comparison of solutions and to do it in an efficient way [10][4].

For a solution representation, it was self-evident to use an F-dimensional bit vector $\mathbf{q}$, where $q_j = 1$, if the feature X_j has to be selected, and $q_j = 0$ otherwise ($j \in \{1, ..., F\}$).

As a mutation operator, we integrated the ASYMMETRIC BIT FLIP, where the probability of switching a bit is equal to:

$$p_q(j) = \frac{\gamma}{F} \cdot (|q_j - p_{01}|), \tag{3.8}$$

where γ controls the general mutation probability and is equal to the expected number of flips during an offspring generation for the symmetric variant of the bit flip mutation. In the asymmetric mutation, the probability for a bit flip is reduced by $|q_j - p_{01}|$, as proposed in [91]. p_{01} controls the probability of a zero-to-one switch. The probability of a one-to-zero switch is set to $p_{10} = 1 - p_{01}$. Because we try to discard as many irrelevant, noisy, and redundant features as possible, it is reasonable to set $p_{01} \ll p_{10}$. In our previous studies, $p_{01} \in \{0.01; 0.1\}$ performed quite well [217, 219].

As the first crossover operator, we implemented a uniform crossover (UC), which selects each bit value either from the first or from the second parent with equal probability. The second operator was a commonality-based crossover (CBC), which was proposed for FS in [53]. Here, the non-shared bits of both parents are inherited from the parent k with the probability

$$p_c(k) = \frac{n_k - n_c}{n_u} \tag{3.9}$$

(n_k is the number of ones for parent k, n_c is the number of the shared ones for both parents, and n_u is the number of non-shared ones for both parents).

However, in [219] we could not observe any significant advantages of both UC and CBC operators. Therefore, in the further studies, which are described in Sections 5.1.2 to 5.2, we left out the crossover.

Besides, we have experimented with different settings of the other SMS-EMOA parameters:

- INITIAL FEATURE RATE if_r controls the expected number of features in the first population after the initialisation. Here, each bit is set to one with the probability if_r, and we used $if_r \in \{0.05; 0.2; 0.5\}$. In [219], we observed that this parameter played a role together with a classifier: SVM performed worse for lower if_r values, and this behaviour was not observed for other classifiers. In general, it is hard to provide an exact recommendation for this parameter. Small if_r values correspond to solutions with larger hypervolumes at the beginning. This situation may be sometimes advantageous but may also lead to a fast convergence to a local optima. Therefore, we used two or three different if_r values in further studies.

[4] In case of four and more objectives, the related optimisation problems are referred to as *many-objective* [89]. Such scenarios are currently unexplored for music classification. They can be reasonable, if several conflicting metrics listed in Sections 4.1.1 and 4.1.2 are considered.

- POPULATION SIZE μ should be large enough to provide a good distribution of solutions, and it was set to 30 for instrument recognition described in Section 5.1.1 and increased to 50 for other experiments which are described in Sections 5.1.2 to 5.2.
- As a STOPPING CONDITION we have chosen the number of SMS-EMOA generations, which was set after the preliminary experiments to 2,000 for the studies described in Section 5.1 and to 3,000 for the recognition of genres and styles which are discussed in Section 5.2. Setting this number to higher values may lead to a further increase of the classification performance, but on the other side to larger computing time requirements.

It is important to mention that a more exhaustive search for the optimal parameter settings was beyond the scope of our study. It is indeed reasonable to make more investigations in that direction in future.

3.3. Sliding feature selection

Until now, the predominant share of music classification studies is based on low-level characteristics close to the audio signal and spectrum, which are described in Section 2.2 and in the corresponding tables. The disadvantage of this approach is that many of these descriptors do not contain meaningful information for music listeners or music scientists, and it is very hard to understand the created classification models. However, user-centered music classification requires more comprehensible classification rules, so that the following questions can be answered (we provide only a few examples for a music recommendation scenario):

- Which instruments are important for a category?
- Which instruments are irrelevant for a category?
- Which instruments are undesired for a category?
- Is the tempo of the songs rather fast or slow for a category?
- How large is a share of songs with a major key in a category?
- Is a high vocal share acceptable for a category?

Some of these high-level characteristics can be derived from metadata or community tags, as investigated for example in [138]. However, the metadata are often imprecise, subjective, erroneous, or not available. Also, we are able to recognise a genre within a few seconds by hearing to music [70] (see also the discussion in Section 2.2.1), so that it should be theoretically possible to derive some important high-level characteristics from the audio signal alone. A related proposal to create so-called 'anchors' as high-level music descriptors was suggested in [11].

In our work we propose a novel SLIDING FEATURE SELECTION framework, which aims at the estimation of *mid-level* characteristics. They may be positioned in between the less interpretable low-level audio features and high-level categories such as genres and personal preferences. These characteristics should be comprehensible and related to music theory. Therefore, we refer to them as *high-level* features. The sliding feature selection, as applied

in our study, is sketched in Fig. 3.4, where several multi-objective evolutionary feature selection steps are applied after each other.

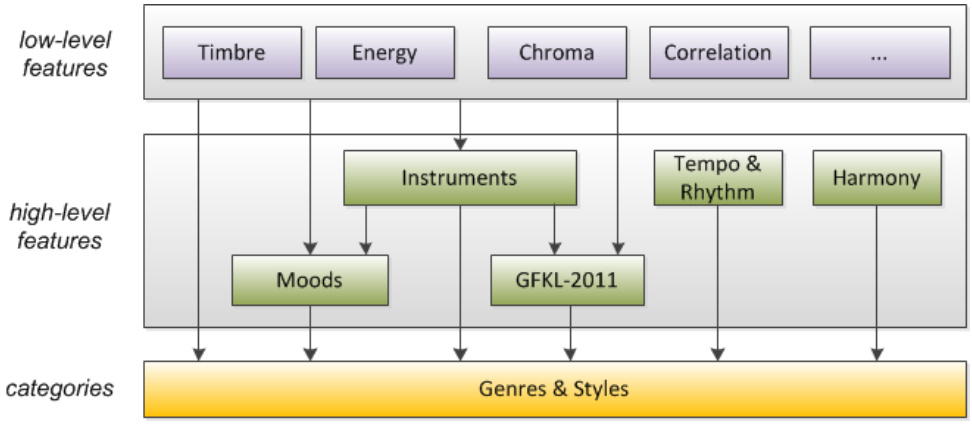

Figure 3.4.: Sliding feature selection for the studies described in Chapter 5. The arrows mark the applied evolutionary multi-objective feature selection.

As an example, in the first step low-level features are integrated into the training of classification models for instrument identification (see Section 5.1.1). Then, these binary models are used for the extraction of new features, which measure the number of positive instrument detections in a selected time interval (for example, 'share of recognised piano onsets in 10 s'). These high-level features are used together with low-level characteristics for the subsequent recognition of moods. In the final step, features built from the mood models (e.g., 'share of 4 s classification windows with energetic mood label in a larger frame of 24 s') contribute to the recognition of genres and styles, together with the instrument features and the original low-level features. This approach has similarities to the concept, which is proposed in [182], where binary classifiers are applied after each other, and the predicted categories are integrated as features for the next step in the categorisation sequence for multi-label classification.

If high-level features are created from low-level ones by application of the classification models, which have been previously optimised with the help of feature selection, a set of high-level characteristics does not contain more source information than the original low-level feature set. Therefore, we cannot implicitly expect that the genre classification based on such high-level features would perform better than the classification based on low-level features.

However, the combination of individual features may increase the classification performance: it is illustrated in Fig. 3.2, and it is a basic concept of the SVM, where a linear combination of original feature dimensions allows the linear separation of the previously linearly non-separable categories. For example, if a music listener wishes to distinguish pop songs with percussion from classical piano pieces, and the 'drum share' high-level feature is estimated from several low-level timbre characteristics, it is indeed more preferable to run categorisation only with this single drum feature.

Even if such performance increase cannot be expected for every category, the unique advantage of this method is that it helps to determine interpretable features. In Section 5.2.3, we compare the classification results based on low-level and high-level feature sets.

3.4. Related works

Considering related publications for evolutionary feature selection (EA-FS) in classification and, in particular, music classification, the following tendency can be observed:

- The first EA designed for FS was introduced by Wojciech Siedlecki and Jack Sklansky more than 20 years ago, in 1989 [195]. Since that time, many further works were published for different research domains, in which evolutionary feature selection has proven its suitability. Evolutionary multi-objective feature selection (EMO-FS) was originally proposed around ten years later, by Christos Emmanouilidis in 2000 [53]. Until now, the number of the corresponding studies remains rather low, we refer to several works in the succeeding section. A possible explanation is that in many research domains it is still established to solve the problems using only one single evaluation/optimisation criterion. We provide some references to EA-FS and EMO-FS in Section 3.4.1.
- Single-objective EA-FS was proposed for the first time for music classification (instrument recognition) by Ichiro Fujinaga in 1996 [66]. Introduced in our previous work [217], EMO-FS became a part of a music classification task in 2011. A discussion of related studies, in particular with EA-FS, is provided in Section 3.4.2.

Figure 3.5 illustrates the intersections of the research fields related to feature selection and music classification.

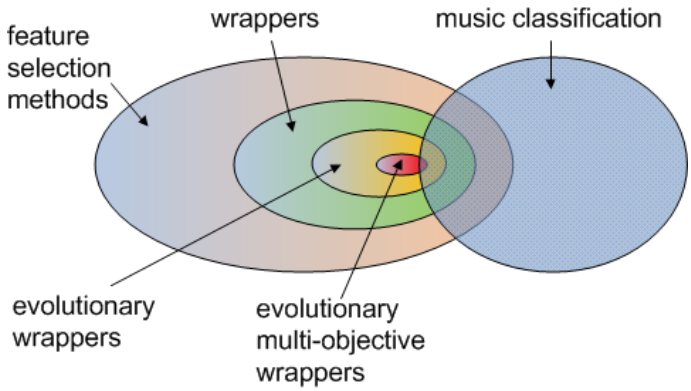

Figure 3.5.: Intersections of several research areas related to FS and music classification.

3.4.1. Evolutionary feature selection

The first studies using EA for FS were characterised by rather small feature sets, compared to the actual situation, where datasets with hundreds and thousands of features are not uncommon anymore:

- The first application of genetic algorithms for feature selection is reported in [195], where the k-nearest neighbours (kNN) classifier was used for tasks with 24 and 30 features. The GA outperformed other methods (exhaustive and sequential selection), and it was recommended to use it in general for "large-scale" feature selection with more than 20 variables.

- The next two works, which were investigated independently of each other, extended GA with an optimisation of the feature weights. Each feature was represented with different values instead of only one bit [98, 176]. This method outperformed the original GA. However, it should be mentioned that the weighting scheme was integrated into the kNN distance estimation, so that this method belongs to the category of embedded FS methods and is not completely independent of the underlying classifier.
- A more extensive study by Kudo and Sklansky compared more than 15 FS algorithms, among others a GA, for different classification tasks with up to 65 features [110]. For the tasks with larger feature sets, the GA outperformed other methods, and was explicitly recommended. Two main advantages over the other methods were outlined: it was easy to control the execution time, compared to the sequential FS, and the repetition of experiments and adjustment of GA parameter settings led to further classification quality improvements.
- An interesting modification related to the fitness evaluation, early stopping, was proposed in [124] to avoid overfitting. Inner cross-validation was implemented for the identification of a GA iteration step, where its generalisation ability became deteriorated. Outer cross-validation was used for the evaluation on the holdout set (for a description of common validation methods see Section 4.2).

Later, the memetic paradigms were introduced, where a local search procedure was combined with stochastic exploration:

- Probably the first work which presented a hybrid GA was [74], where random hill climbing was applied as a local search. Classification was done with Euclidean decision tables, and several data sets of up to 204 features were used as classification problems.
- In [160], local search operators were designed either to remove the least significant feature or to add the most significant feature. On two datasets with up to 80 features, the hybrid GA outperformed a non-hybrid variant and three other baseline methods. Classification was done by kNN. Later, the advantage of the hybrid GA was also confirmed for 11 data sets with up to 100 features [161].
- The problem of local search costs was addressed in [239] by several improvements, where different local search operators were proposed. Memetic algorithms outperformed a simple GA and two baseline methods.

Multi-objective evolutionary feature selection became a part of many applications in recent years:

- In [53], FS was for the first time formulated as a multi-objective problem, which was solved with the help of evolutionary techniques, optimising the misclassification rate and the number of selected features. The categorisation tasks contained up to 60 features, and the classification was done with neural networks (NN). Related studies are discussed in more detail in [52], where also a list of older EA-FS works is provided and a multi-objective optimisation of the true positives and the true negatives is described.
- Several succeeding studies applied EMO-FS for handwritten digit recognition (with 132 features and NN as classifier) [163], bankruptcy prediction (30 features, classi-

fication by SVM) [69], and five classification tasks from the UCI repository (up to 180 features, classification by kNN) [78].

- The first extension to the unsupervised classification, where metrics for cluster quality evaluation were considered, is introduced in [99]. Unsupervised EMO-FS was investigated in more detail later in [80].
- The simultaneous optimisation of generic feature sets, which were relevant for several classification tasks, compared to specific feature sets, which were relevant for the individual tasks, was applied in [240].

Concluding our literature study, it is important to mention that the number of tested classification methods and categorisation tasks was rather low in almost all studies. Also, many earlier and some of the recent applications applied only simple classifiers, such as kNN, and the classification tasks contained at most up to several hundreds of features. In future, it might be promising to provide an exhaustive testing framework for the reliable comparison of enhancements and algorithm parameters for different application scenarios.

3.4.2. Feature selection in music classification

With a growing number of available audio features and also features from other domains, FS also became a relevant part in music classification. Table 3.1 provides several examples of the feature selection methods applied in music classification, except for EA-FS (the related works are listed below in Table 3.2).

Table 3.1.: Selected works with FS applied in music classification tasks, sorted by year (except for EA-FS). Abbreviations for FS methods (for details see the publications): CFS: correlation-based, FCBF: fast correlation-based filter, IRMFSP: inertia ratio maximisation using feature space projection, PCA: principal component analysis, SBE: sequential backward elimination, SFF: sequential feed forward, SFS: sequential forward selection, SFS-IG: sequential forward selection with information gain ranking, SFS-GR: sequential forward selection with gain ratio ranking.

Author(s)	Ref.	Year	Feat. No.	FS method
Burred & Lerch	[23]	2003	58	SFF
Grimaldi et al.	[73]	2003	143	SFS-IG, SFS-GR, PCA
Essid et al.	[57]	2006	160	IRMFSP
Fiebrink & Fujinaga	[60]	2006	74	SFS
Livshin & Rodet	[123]	2006	513	CFS
Blume et al.	[18]	2008	19	Mann-Whitney test
Krishnamoorthy & Kumar	[108]	2010	>22	Kolmogorov-Smirnov test
Mayer et al.	[137]	2010	60-1140	FCBF
Saari et al.	[189]	2011	66	SFS, SBE
Eichhoff & Weihs	[51]	2012	6-276	SFS

Several limitations can be observed:

- Often rather simple methods, like sequential forward selection, are integrated. This method was outperformed by floating search as reported in [110]. Other individual

feature ranking methods are also often used (their drawbacks are mentioned in the discussion of Fig. 3.2, Section 3.2.3).

- The number of features is usually relatively low compared to the EA-FS studies discussed in Section 3.4.1.
- Integration of a correct validation scheme is not always guaranteed, although this problem was mentioned in several publications: the necessity of an independent test set for the validation of feature selection is discussed in [60], and [189] points to the lack of reliable validation in MIR related FS applications.

Table 3.2 lists the works, where EA were applied for FS in music classification tasks (we excluded our own publications, they are listed below in Table 3.3).

Table 3.2.: Studies (except for own works) with evolutionary FS applied in music classification tasks, sorted by year. Abbreviations for EA methods (for details see the publications): GA: genetic algorithm, GAw: GA with feature weighting, GAr: GA with restricted number of features, ES: evolutionary strategy, MGAw: memetic GA with feature weighting, PSO: particle swarm optimisation, PSO-OPF: PSO with optimal path forest classifier.

Author(s)	**Ref.**	**Year**	**Feat. No.**	**EA method**
Fujinaga	[66]	1996	-	GAw
Fujinaga	[67]	1998	352	GAw
Fiebrink et al.	[61]	2005	8-57	GAw
Mierswa & Morik	[146]	2005	variable	(1+1)-ES
Essid et al.	[57]	2006	160	GA
Alexandre et al.	[3]	2007	76	GA,GAr
Olajec et al.	[162]	2007	63	GA
Kramer & Hein	[106]	2009	100	(5+25)-ES
Silla et al.	[196]	2009	30-168	GA
Nayak & Butani	[158]	2010	74	GA
Karkavitsas & Tsihrintzis	[96]	2011	81	MGAw
Marques et al.	[132]	2011	74 and 33,618[a]	PSO-OPF
Chmulik et al.	[30]	2012	137	GA, PSO
Knight	[103]	2012	168	GA

[a]The reason for a very high number of features is explained as follows: for one task, 26 cepstral coefficients were extracted for 1,293 extraction frames and treated as independent features. This approach has a high danger of overfitting, since only 999 classification instances were analysed.

Several historical tendencies can be observed:

- The first work, which directly mentions EA in a music classification scenario, is [66], where only a method description was provided. The related study results were published in [67]. Here, GA were used to optimise the feature weights for the kNN classifier, similar to the approach in [176]. This method also was later applied in [61] for snare drum classification. In [96], a GA with feature weighting was extended with local search for genre recognition. Another embedded FS algorithm combining a classifier (optimum-path forest) with an EA (particle swarm optimisation) was originally proposed in [179] and applied for genre recognition in [132].

- Later investigations applied GA with a bit string representation for a broader range of music classification scenarios: class pairwise selection in instrument recognition using GMM and SVM [57], applause detection with GMM [162], genre classification (5 classifiers and 3 different classification approaches) [196], or multi-class recognition of trumpet tone quality by SVM [103]. In [3], a GA with a restricted number of selected features outperformed a simple GA for the classification of speech and noise. However, the challenge of this modification is that it is necessary to select a proper boundary for the number of features.
- Apart of genetic algorithms, ES are rather underrepresented in the application of FS for music classification. In [146], a (1+1)-ES was applied for the recognition of genres and user preferences, and in [106], a (5+25)-ES was applied for drum categorisation. PSO was used together with a GA in [30] for sound classification.
- In [158], the fitness function was adapted to favour smaller feature sets. This could be theoretically described as a multi-objective approach, reduced to a single-objective by the objective weighting. However, to our knowledge, no study included an explicit multi-objective optimisation.
- Only a few studies involve classification tasks with more than 200 features. Because EA-FS performs especially well for large feature sets (in [110], GA was preferable for all problems with 100 and more variables), the advantage of the evolutionary methods over other methods cannot be fully exploited.

It is also worth to mention the studies with evolutionary-based *feature construction*, or generation, where the ways to estimate features were designed by EA. In the first related studies [166, 146], the features were estimated with method trees evolved by genetic programming. This approach was also applied in [153]. Later, a multi-dimensional PSO was applied for feature construction, classifying 16 different audio categories [127].

Table 3.3 lists some statistics of our own publications on evolutionary FS in music classification (for more details see Section 1.3).

Table 3.3.: Own studies with evolutionary FS applied in music classification tasks, sorted by year. The column marked with 'Asym. mut.' indicates the application of an asymmetric mutation.

Ref.	**Year**	**Tasks**	**Feat. No.**	**Classifier**	**EA method**	**Asym. Mut.**
[220]	2008	3 personal categories	33	C4.5	(1+1)-ES	no
[223]	2009	3 personal categories	33	C4.5	memetic (1+1)-ES	no
[15]	2010	7 AMG genres and styles	572	C4.5	several (1+1)-ES (memetic / self-adaptive)	yes
[217]	2011	6 AMG genres and styles	572	C4.5, RF, NB, SVM	(30+1) SMS-EMOA	yes
[218]	2012	3 AMG genres	674	RF	(30+1) SMS-EMOA	yes
[219]	2012	8 instruments	1,148	C4.5, RF, NB, SVM	(30+1) SMS-EMOA	yes
[216]	2013	4 instruments	1,250	RF	(30+1) SMS-EMOA	yes

4. Evaluation Methods

A key question when designing an algorithm is how it performs compared to other methods. A reasonable evaluation takes into account several different criteria (the terms evaluation criterion, measure, metric, and objective are used as synonyms within the scope of this thesis). Any improvement of an algorithm does not mean that it would lead to an increase of performance with regard to all relevant metrics. For example, it may be possible to achieve smaller classification errors, paying this price by higher computing demands for the tuning of parameters, classification training, and classification. Or, the performance of the binary classification may differ on positive and negative instances, which means that the algorithm is not suitable for highly imbalanced data sets. Several groups of evaluation metrics are presented in Section 4.1.

The data themselves are also a very sensitive part of the evaluation process. If the validation data are too similar to the training data, an algorithm will perform almost perfect, but we would not learn anything about the generalisation ability of the method to classify new data. On the other side, it is obvious that a model which has been trained to distinguish between classical recordings of piano and symphonic orchestra will perform completely unexpected, if it would be applied on electronic music. The validation data should not be completely different from the training data. Several recommendations how the instances for classification and evaluation should be organised are discussed in Section 4.2.

Finally, any comparison of significance of study outcomes should be assessed by statistical tests. In that case, the confidence of the statements, such as 'algorithm $\mathcal{A}$ significantly outperforms algorithm $\mathcal{B}$' or 'tuning of parameter $\mathcal{P}$ does not bring any significant improvements', can be estimated by means of assumed statistical distributions. The ideas behind the statistical tests and the descriptions of tests applied in this work are shortly outlined in Section 4.3.

4.1. Evaluation metrics

In Section 2.1.1, we discussed several applications of music data analysis. It is obvious that the criteria for algorithm evaluation should be carefully chosen, depending on the situation and the goals of the current task. In many related publications, often only one or a couple of well-known evaluation metrics, such as accuracy, number of misclassifications, precision, recall, etc. is estimated.

For an extensive comparison of classifiers, it can be indeed reasonable to estimate a larger number of metrics: even for the closely related confusion matrix metrics some of them may be hardly correlated, as affirmed by our study on music genre and style recognition [214]. Furthermore, a better performance w.r.t. one measure may correspond to a decreased performance according to another metric. Several examples are:

- A smaller classification error on positive instances in binary classification may correspond to larger errors on negative instances. We observed this tendency in [217]. The models with the highest recall rates had the lowest specificity rates, and vice versa; the non-dominated fronts were built from a large number of different trade-off solutions for the examined classification tasks. For some applications, it may be indeed preferable to achieve higher classification performance on the positive examples. It is not a problem, if an acceptable share of the negative examples is identified as positive (a *surprise* effect in music recommendation). For other situations, such as the generation of a party playlist, it is preferable to guarantee that no negative songs are categorised as belonging to the dance genre.
- A very common situation is that the algorithms with the lowest errors are slower than the less complex classifiers. In our experiments from the previous studies and the studies discussed in Chapter 5, the NB classifier was the fastest, but often had rather large error rates.
- Not only the runtime demands, but also other costs of music classification can be estimated for the evaluation of algorithms. For example, if some features require long extraction time and should be stored for very large music collections, it is thinkable to substitute them by less expensive and less powerful features (related to their ability for class separation) or to extract them only from very short song intervals, applying one of the methods for the processing of the time dimension of the feature matrix, which are discussed in Section 2.3.3.
- A trade-off between highly complex models, which provide very small classification errors, but are less suitable for unseen data, and smaller and more generalisable models, can be taken into account. As described above in Section 2.4.1, a part of the C4.5 algorithm is designed to search for a balance between too complex and overfitted trees and smaller pruned trees, which admit an acceptable rate of misclassifications.

The next step beyond multi-objective evaluation of algorithms is multi-objective optimisation, as introduced in detail in Section 3.2 for evolutionary feature selection. Though we have applied only a couple of metric combinations for EMO-FS, we believe that many further combinations up to the many-objective scenarios will be investigated in future. Therefore, we provide an extensive list of evaluation measures in the following two sections.

The classification performance metrics (Section 4.1.1) are often calculated in different supervised classification applications, however they are rather seldom combined for multi-objective evaluation – and even less used as a target for multi-objective optimisation. All these metrics are available in AMUSE. We compared many confusion matrix-related measures for music classification in [214].

Section 4.1.2 provides a discussion of some promising metric groups for future research. They are not so often applied in music classification studies or are designed only for specific classification tasks.

4.1.1. Confusion matrix and classification performance measures

Because our current research is focused on binary classification tasks, we provide below the formulas for metric estimation mostly for binary classification. They can be easily

extended for multi-class tasks.

The most commonly used evaluation metrics are estimated from the CONFUSION MATRIX, which is a quadratic $(C \times C)$-matrix, where the rows correspond to the labelled categories y_L, columns to the predicted categories y_P, and the entry in row j and column k is equal to the number of classification instances labelled as belonging to category y_L, which were predicted as belonging to category y_P.

Let T be the overall number of classification instances, or windows, and $\mathbf{x}_i$ denote the feature vector for a classification window $i \in \{1, ..., T\}$ from the processed feature matrix $\mathbf{X}$. For the binary classification $y_L, y_P \in \{0; 1\}$, and the confusion matrix consists of the four following entries:

- The number of TRUE POSITIVES corresponds to the number of positive instances predicted as positive:
$$TP = \sum_{i=1}^{T} y_L(\mathbf{x}_i) \cdot y_P(\mathbf{x}_i). \tag{4.1}$$

- The number of TRUE NEGATIVES corresponds to the number of negative instances predicted as negative:
$$TN = \sum_{i=1}^{T} (1 - y_L(\mathbf{x}_i)) \cdot (1 - y_P(\mathbf{x}_i)). \tag{4.2}$$

- The number of FALSE POSITIVES corresponds to the number of negative instances predicted as positive:
$$FP = \sum_{i=1}^{T} (1 - y_L(\mathbf{x}_i)) \cdot y_P(\mathbf{x}_i). \tag{4.3}$$

- The number of FALSE NEGATIVES corresponds to the number of positive instances predicted as negative:
$$FN = \sum_{i=1}^{T} y_L(\mathbf{x}_i) \cdot (1 - y_P(\mathbf{x}_i)). \tag{4.4}$$

Several metrics are derived from TP, TN, FP and FN (we provide here again the definitions for binary classification, which can be extended for a multi-class case):

- ACCURACY corresponds to the average rate of correctly predicted instances:
$$m_{ACC} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{T}. \tag{4.5}$$

  If a data set is highly imbalanced, and a classifier performs well on the stronger represented class (in worst case classifying *every* instance as belonging to the strongest category), the accuracy may be indeed high. Therefore, the calculation of other metrics is reasonable. On the other side, it depends on the classification scenario,

if the classification performances on the stronger and weaker classes have an equal relevance.

- PRECISION describes the fraction of the correctly identified positive instances to the number of instances identified as belonging to this category:

$$m_{PREC} = \frac{TP}{TP + FP}. \tag{4.6}$$

- RECALL, or SENSITIVITY, is the fraction of the correctly identified positive instances to the number of positive instances:

$$m_{REC} = \frac{TP}{TP + FN}. \tag{4.7}$$

- SPECIFICITY measures the percentage of the negative instances, which were predicted as negative:

$$m_{SPEC} = \frac{TN}{FP + TN}. \tag{4.8}$$

Numeric prediction errors measure the number of misclassifications, and can also be applied for binary classification (the corresponding formulas are marked with '$\overset{bin}{=}$'):

- ABSOLUTE ERROR is equal to the number of misclassifications:

$$m_{AE} = \sum_{i=1}^{T} |y_L(\mathbf{x}_i) - y_P(\mathbf{x}_i)| \overset{bin}{=} FP + FN. \tag{4.9}$$

- RELATIVE ERROR corresponds to the average number of misclassifications:

$$m_{RE} = \frac{1}{T} \cdot \sum_{i=1}^{T} |y_L(\mathbf{x}_i) - y_P(\mathbf{x}_i)| \overset{bin}{=} \frac{FP + FN}{TP + TN + FP + FN}. \tag{4.10}$$

- MEAN SQUARED ERROR can be estimated, if the ground truth is not always binary as defined in our earlier studies [205, 223] (we used a slightly modified m_{MSE} version there):

$$m_{MSE} = \frac{1}{T} \sum_{i=1}^{T} (y_L - y_P)^2 . \tag{4.11}$$

Some metrics are designed especially for the measurement of classifier performance on imbalanced sets:

- BALANCED RELATIVE ERROR is the mean of the relative errors estimated separately for the instances of both classes:

$$m_{BRE} = \frac{1}{2} \left(\frac{FN}{TP + FN} + \frac{FP}{TN + FP} \right). \tag{4.12}$$

- F-MEASURE is the weighted harmonic mean of precision and recall:

$$m_F = \frac{(\alpha_F + 1) \cdot m_{PREC} \cdot m_{REC}}{\alpha_F \cdot m_{PREC} + m_{REC}}, \text{ where} \tag{4.13}$$

  α_F adjusts the balance between m_{PREC} and m_{REC} and is often set to 1.

The following three metrics are combinations of recall (sensitivity) and specificity, which are also helpful for classifier evaluation on imbalanced sets. Their application for the evaluation of classification is motivated in [198].

- YOUDEN'S INDEX is a simple combination of m_{REC} and m_{SPEC}:

$$m_Y = m_{REC} + m_{SPEC} - 1. \tag{4.14}$$

- POSITIVE AND NEGATIVE LIKELIHOODS measure the performance on positive and negative instances separately, however with respect both to sensitivity and specificity values:

$$m_{L+} = \frac{m_{REC}}{1 - m_{SPEC}}; m_{L-} = \frac{1 - m_{REC}}{m_{SPEC}}. \tag{4.15}$$

- GEOMETRIC MEAN is the squared product of m_{REC} and m_{SPEC}:

$$m_{GEOM} = \sqrt{m_{REC} \cdot m_{SPEC}}. \tag{4.16}$$

Another possibility to evaluate the classification quality is to measure the correlation between the sequence of labels for all classification windows $\mathbf{y}_L$, and the sequence of predicted categories for all classification windows $\mathbf{y}_P$:

- STANDARD CORRELATION COEFFICIENT is equal to 1 in case of the strongest dependency between the input variables, -1 in presence of the strongest anticorrelation (an increase of the first variable leads to a decrease of the second one) and is equal to 0, if the variables are not dependent on each other. It is defined as follows:

$$m_c = \frac{Cov(\mathbf{y}_P, \mathbf{y}_L)}{\sqrt{Var(\mathbf{y}_P) \cdot Var(\mathbf{y}_L)}}, \text{ where the COVARIANCE is:} \tag{4.17}$$

$$Cov(\mathbf{y}_P, \mathbf{y}_L) = \frac{\sum_{i=1}^{T} (\mathbf{y}_P - \overline{\mathbf{y}}_P) \cdot (\mathbf{y}_L - \overline{\mathbf{y}}_L)}{T - 1} \text{ and the VARIANCES are:} \tag{4.18}$$

$$Var(\mathbf{y}_P) = \frac{\sum_{i=1}^{T} (\mathbf{y}_P - \overline{\mathbf{y}}_P)^2}{T - 1}; Var(\mathbf{y}_L) = \frac{\sum_{i=1}^{T} (\mathbf{y}_L - \overline{\mathbf{y}}_L)^2}{T - 1}. \tag{4.19}$$

- SPEARMAN'S RHO RANK COEFFICIENT is a special case of the Pearson product-moment correlation coefficient, where $R(\cdot)$ measures a rank of the input variable,

based on the preceeding sorting:

$$c_\rho = \frac{\sum_{i=1}^{T} \left(R(y_P(\mathbf{x}_i)) \cdot R(y_L(\mathbf{x}_i))\right) - T\left(\frac{T+1}{2}\right)^2}{\sqrt{\left(\sum_{i=1}^{T} \left(R^2(y_P(\mathbf{x}_i))\right) - T\left(\frac{T+1}{2}\right)^2\right) \cdot \left(\sum_{i=1}^{T} \left(R^2(y_L(\mathbf{x}_i))\right) - T\left(\frac{T+1}{2}\right)^2\right)}} \tag{4.20}$$

Because we build many classification windows from a single song for genre and style prediction (see Section 2.3.4), we distinguish between SONG-LEVEL and CLASSIFICATION WINDOW-LEVEL evaluation metric estimation. The classification window-level evaluation calculates the performance for all classification windows. For binary song-level evaluation, which is based on a binary partition-level classification with $y_P(\mathbf{x}_i) \in \{0; 1\}$, we estimate the predicted song category by majority voting across all predicted labels for classification window feature vectors:

$$y_P(\mathbf{x}_1, ..., \mathbf{x}_{T'}) = \left\lceil \frac{\sum_{i=1}^{T'} y_P(\mathbf{x}_i)}{T'} - 0.5 \right\rceil, \text{ where} \tag{4.21}$$

T' is the number of classification windows in a song. The $y_P(\mathbf{x}_i)$ and $y_L(\mathbf{x}_i)$ values in Equations 4.1 to 4.20 can then be replaced by the corresponding labels for songs. If a metric m_i was estimated on the song level, we denote it by m_i^s.

The metrics estimated on window level evaluate a classifier more precisely. On the other side, for user-driven scenarios it is almost always acceptable or even desired that complete songs are assigned to categories. In the last case, the classification performance is usually better than the classification window-level performance: for example a 'classic' song is identified correctly for the share of 'classic' classification windows between 50% and 100%. Therefore, we applied partition-level FS optimisation for the recognition of the high-level features, and song-level FS optimisation for the recognition of genres and styles.

Figure 4.1 illustrates this effect. Both subfigures plot the balanced relative error and the selected feature rate (defined later in Equ. 4.24) on the holdout set from the feature subsets, which have been generated during 2,000 evaluations of the 10 experiments for the recognition of the Classic category with RF. The runs in the left subfigure were evaluated and optimised using m_{BRE} (partition-level) and m_{SFR}. For the right subfigure, the metrics were m_{BRE}^s (song-level) and m_{SFR}. Song-level classification has significantly lower errors than window-level classification: even with larger feature sets almost always $m_{BRE}^s < 0.04$, and for window-level classification in most cases $0.04 < m_{BRE} < 0.1$.

4.1.2. Further metrics

In [217], we discussed five categories of metrics, which can be estimated for the evaluation of music classification: common quality-based, resource, model complexity, user interaction, and specific metrics. The measures from the first group are listed in Section 4.1.1, and most of them were used in our studies. The metrics from the last four groups are not so commonly used in music classification, but are, in our opinion, very promising for the multi-objective evaluation of music classification in future.

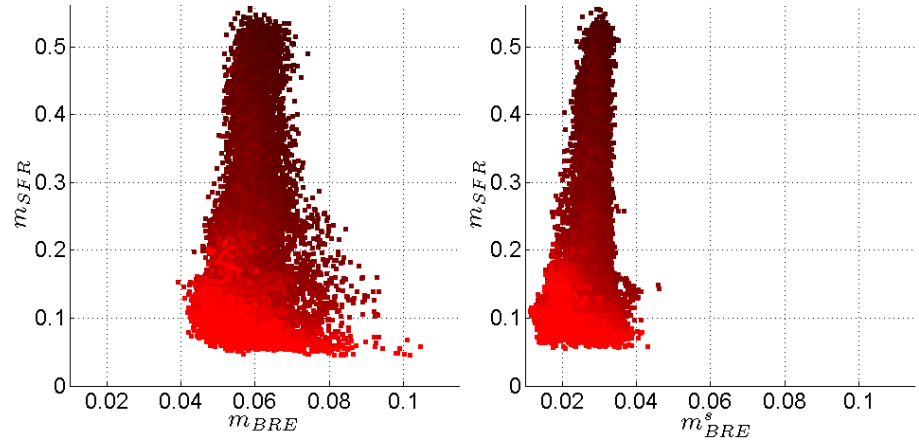

Figure 4.1.: All solutions found during the 10 statistical repetitions of FS optimisation by SMS-EMOA and classification by RF, category Classic. Left subfigure: partition-level optimisation. Right subfigure: song-level optimisation.

Resource metrics estimate algorithm runtime and storage demands. One of the few works which provide a general categorisation of these metric group is [210]. They can be calculated for each stage of the music classification chain discussed in Section 2.1.3, for example:

- The $\mathcal{CT}$ runtime is relevant, if new music categories are frequently created.
- The $\mathcal{C}$ runtime becomes crucial, if the same categorisation models are applied for different music collections, for example, if a music online shop applies automatic classification of new songs each day.
- The same holds for the $\mathcal{FE}$ runtime: although feature extraction is usually done only once for each new music track, it can be very costly. For example, it was observed in [18] that the estimation of autocorrelation, fundamental frequency, and power spectrum required more than 65% of the overall extraction time for the set of 25 common audio features. Therefore, too long extraction times may lead to problems for often updated music collections as well as for devices with limited resources.
- The $\mathcal{FP}$ reduction rate measures the number of entries in the processed feature matrix $\mathbf{X}'$ divided by the number of all feature dimension values before any processing:

$$m_{FPRR} = \frac{F \cdot T'}{\sum_{i=1}^{F^*} \left(T^{**}(i) \cdot F^{**}(i)\right)}. \tag{4.22}$$

  For each feature i, the number of extracted values is equal to the product of the number of dimensions $F^{**}(i)$ and the number of extraction windows $T^{**}(i)$, see also note 8 in Section 2.1.3. m_{FPRR} provides a rough estimation for the storage demands which are required to index the music files.
- A modified version of m_{FPRR} was used in [222] for the comparison of different time dimension processing methods (see the discussion of Fig. 2.10 in Section 2.3.3). The time windows reduction rate corresponds to a relative number of the selected time windows, compared to the number of the smallest extraction frames, which are

required for the harmonisation of the feature matrix $\mathbf{X}^H$ (see Section 2.3):

$$m_{TWRR} = \frac{T'}{T^H}, \text{ where} \tag{4.23}$$

T^H is the time dimensionality of the harmonised feature matrix $\mathbf{X}^H$ before further processing.

MODEL COMPLEXITY METRICS estimate the complexities of the classification models. The more complex models often have a higher tendency to be overfitted towards certain data sets, in particular, the training data, so that the classification performance on other data sets is deteriorated. This metric group is sometimes closely related to the resource metrics: a more complex model is often built from a larger amount of features and has higher storage demands.

- A crude measure for model complexity is the SELECTED FEATURE RATE:

$$m_{SFR} = \frac{F}{F^*}. \tag{4.24}$$

  A larger number of input variables often leads to more complex models, and the danger increases that some noisy features are coincidently recognised as relevant. This especially holds, if the number of features is larger than the number of classification instances.

- The generalisation performance of classification models can also be evaluated according to stability criteria, such as the deviation of the classification performance on different validation sets. An example for such a measure is proposed in [113].

- The classifier-specific model complexity metrics compare models, which are created by the same classifier, but with different parameters. An SVM-specific complexity measure is discussed in [145]. For C4.5, the number of tree nodes measures the tree complexity.

A group of USER RELATED METRICS makes sense for any classification scenario, where the users are either involved in ground truth labelling, or the categorisation itself aims at user satisfaction. Examples for these metrics are:

- Listener satisfaction with the music classification results.
- Feedback efforts, if the user plays a role in active learning [86].
- Efforts to create the ground truth are usually in conflict with the classification performance: the smallest number of misclassifications can be achieved, when a large number of the labelled songs from different genres exists. However, high manual efforts for labelling are necessary for that case.
- High interpretability of the classification models and the involved features helps to understand the category properties, for example, if a decision tree model is built with high-level features. Each step of the algorithm chain (Fig. 2.3), which aims at the increase of the classification performance, may on the other side reduce the interpretability: e.g., if the $\mathcal{FE}$ outputs a large number of complex and less comprehensible audio signal characteristics, $\mathcal{FP}$ applies statistical feature dimension processing,

or the $\mathcal{CT}$ transforms the original feature dimensions into the higher-dimensional SVM models.

The main disadvantage of the user related metrics is that many of them are rather subjective, and/or cannot be calculated automatically, so that high manual efforts are required. Some of the MIR related user-centered metrics are discussed in [26, 122].

Specific performance evaluation metrics are designed with the primary goal to evaluate the music classification for a concrete application, and not music classification in general, e.g.:

- For playlist generation based on genres and other preferences, it is reasonable to measure the playlist diversity or novelty effect. These metrics also play a role for other music recommendation scenarios, where the user would not be satisfied, if the music from the same artist will be recommended over and over again.
- For some of the music classification applications, the structure and the order of the classification instances plays a role. The standard evaluation measures introduced in Section 4.1.1 are then not optimal anymore, since they are invariant to the order of instances. One of such tasks is hierarchical music segmentation, which detects several structure levels: from larger segments, characterised by their instrumentation, to shorter harmony-related sequences and rhythm-related note groups in the bar. For music segmentation in general, specific evaluation metrics are proposed in [125]. For hierarchical segmentation, it is possible to adapt the measures from the image processing domain, which take structural information into account [225].
- For tempo recognition, sometimes an *octave* error occurs, so that the tempo is estimated as the double of the original one. This is often a consequence of several autocorrelation peaks with an (almost) similar amplitude or of several tempo levels from different instruments. Slow music pieces would be then identified as fast [40]. Here, it is possible to define specific measures, which consider these different ground truth peaks. The doubling tempo error may be less penalised than other deviations from the original tempo. Another possibility is to allow some acceptable deviation from the labelled tempo [140].

4.2. Organisation of data for evaluation

Another essential part of algorithm evaluation is the careful choice of the *data* for classification and evaluation. First, we should clarify the differences between data sets, which are used in the evaluation process. The following definitions hold within the scope of this thesis and may differ from the same terms in other literature, in particular, if it is distinguished only between training and validation:

- Training set is a subset of labelled classification windows for the training of classification models.
- Validation set is used for the evaluation of the models, which have been previously created from the training set. Because the estimated metrics can be used as an optimisation criterion, as it is done in this work for the optimisation of feature selection (see Chapter 3), we also refer to this set as optimisation set.

- EXPERIMENT SET contains both the training and the validation sets.
- HOLDOUT SET is kept for the independent evaluation of the models, which have been trained from the training set, and have been optimised using the optimisation set. The experiment and holdout sets should be disjunct for all experimental studies, to avoid overfitting (see below Def. 4.1).

The organisation of the experiment set into training and validation sets can be done with the help of many strategies. The most popular are the following (see [14] for pseudocode examples):

- n-FOLD CROSS-VALIDATION (n-fold CV) is one of the most common techniques. Here, the experiment set is divided into n disjoint partitions of equal size. Based on the rotation principle, each partition with $\frac{T}{n}$ instances is exactly once reserved for the validation set, and the remaining $n-1$ partitions with $T \cdot \left(\frac{n-1}{n}\right)$ instances are used to train the classification models. If $n = T$, this method is called leave-one-out cross-validation, since each validation set always consists of exactly one instance.
- In BOOTSTRAP, the partitions are not disjoint. In the beginning, T instances are drawn for the training set with equal probability and independently from each other. Each instance has a chance of $\frac{1}{T}$ of being selected in each draw. The probability of not being selected is approximately equal to $e^{-1} = 0.368$, so that approximately 36.8% of the instances are not selected at all, and these instances are used for the validation set. The whole process is repeated n times and n is usually set to much higher numbers than for n-fold CV. Several enhancements of bootstrap led to the development of the more complex .632 and .632+ bootstrap validation techniques [49].
- NESTED CROSS-VALIDATION contains several levels of CV. For example, the algorithm parameters can be first tuned within an inner CV loop, and the results of this process can be evaluated within an outer CV loop. An illustrative example is provided in [14]. This method was proposed for EA-FS in [124]. Although this validation method supports the most reliable estimation of the generalisation performance of algorithms (compared to other methods discussed above), its main drawback is that the runtime increases exponentially by the insertion of the additional levels.

For an explanation, why an independent holdout set is necessary for algorithm evaluation, we provide the following explicit definition of OVERFITTING from [150], p. 67:

Definition 4.1 *Given a hypothesis space $\mathcal{H}$, a hypothesis $h \in \mathcal{H}$ is said to* OVERFIT *the training data if there exists some alternative hypothesis $h' \in \mathcal{H}$, such that h has smaller error than h' over the training instances, but h' has a smaller error than h over the entire distribution of instances.*

For the calculation of the risk to create overfitted models or, in other words, for the measurement of the GENERALISATION ABILITY to perform well on the unseen data, it is essential to keep a certain proportion of the available labelled instances untouched during the tuning and optimisation of algorithms. One of the most common mistakes in the evaluation of classification is that the training and the validation sets are rotated based on n-fold CV, and the algorithm performance is measured as the mean error across all

validation sets. This error (or any other estimated metric) is not completely independent from the training data involved into the classification models.

To name a couple of references for further reading, in [183] it is discussed how the danger of overfitting can be minimised for the evaluation of feature selection. [180] provides several experimental suggestions that the overfitting risk increases, when many classification methods are compared, and large data sets are used for model creation during n-fold CV. [60] gives an interesting insight into the re-evaluation of the previous study on snare drum classification where no independent holdout set was used originally.

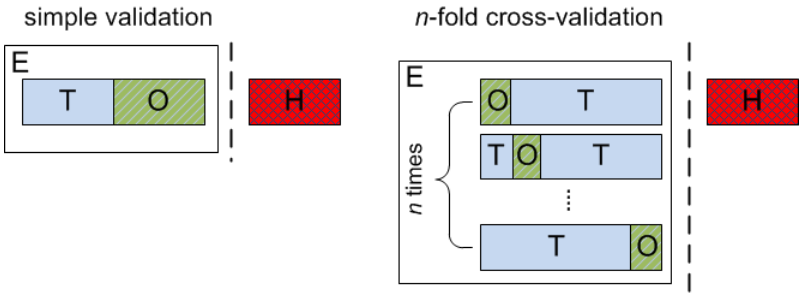

Figure 4.2.: The two validation methods used in our studies. T: training set, O: optimisation set, E: experiment set, H: holdout set.

Figure 4.2 illustrates the organisation of data sets, as applied in our studies. Because of the long experiment runtimes, we had to choose a compromise between the number of experiment parameters, number of categories to predict, and the number of evaluations. Therefore, we did not integrate a nested CV, but reserved an independent holdout set before each experiment for the reliable evaluation of the optimised feature sets. For the studies described in Sections 5.1.1 and 5.1.3, we used n-fold CV on the experiment set (right subfigure). Even without a nested strategy, the optimisation of feature selection and independent validation required high computing efforts, see the numbers of model evaluations in Tables 5.1 and 5.3, and the corresponding explanation of the optimisation parameters in Section 5.1.1. The simple validation (left subfigure) was applied for mood and genre and style recognition (Sections 5.1.2 and 5.2). For moods, the number of the albums with positive song examples was not sufficient to provide a reasonably large experiment set with enough positive instances for model creation based on n-fold CV. For genres and styles, we have integrated training sets of only 10 and 20 songs, as motivated in our first studies [205, 220] – the reason was to create a classification scenario close to the real world situation, where a listener defines a music category with a limited number of songs.

Another relevant issue is the assignment of instances (classification windows) to the sets, which are defined above. As we have already briefly mentioned in the beginning of this chapter, it does not make sense to validate an algorithm on the test data which are too similar or too different from the training data. Therefore, it is reasonable to create experiment and holdout sets which

- are completely disjunct, i.e. none of the instances appear in both sets, and which
- have an approximately equal category distribution.

The sets should not be too small: too many features and too few instances increase the danger of overfitting, so that the noisy and irrelevant features would be misleadingly recognised as relevant. Although we use training sets of only 10 and 20 songs, we benefit from a large number of classification windows per song. Even if 10 songs are chosen for the training set, the number of classification instances is more than thousand, assuming that a typical song of 4 min length consists of 119 classification windows with $W_c = 4$ s and $S_c = 2$ s. It is important to mention that classification windows of the same song might be very different because of the varying instrumentation, note distribution, harmonic and energy characteristics, etc.

Finally, according to the No Free Lunch theorem [233], no algorithm can be treated as considerably better than any other, if the algorithms would be evaluated across all possible classification tasks. For each classification method, parameter setting, or the selected feature set, there exist some data sets for which a particular method performs very well – and there exist other data sets, which are especially hard to classify with this algorithm. This does not mean that it is not possible to give some recommendations – but it makes sense:

- to precisely set up and describe the classification task,
- to find out, which algorithms are preferable within the scope of this and similar scenarios, and
- to choose proper evaluation metrics.

For further recommendations concerning the organisation of data for evaluation, see, e.g., [45, 4, 14].

4.3. Statistical hypothesis testing

The task of INFERENTIAL STATISTICS is to measure how probable is the fact that a study outcome was not achieved by chance. On the other side, it should be kept in mind that a significant difference between two distributions does not necessarily measure the dimension of this difference: for example, method $\mathcal{A}$ may outperform $\mathcal{B}$ with regard to the misclassification error, but the advantage could be so marginal that it does not make sense to replace $\mathcal{B}$ by $\mathcal{A}$. Therefore, it is also reasonable to describe the differences and trends by means of DESCRIPTIVE STATISTICS.

The choice of a proper test should be done very carefully. It depends on the data characteristics: the number of observations and their mathematical distributions, the value domain, the relation between the observations (paired and not paired), etc.

First, the test objectives should be considered. Usually, the following disjoint hypotheses are formulated (cf. [84], p. 2 and [36], p. 4):

Definition 4.2 NULL HYPOTHESIS (H0) *postulates that there is no difference between the probability distributions of some study outcomes, i.e. they belong to the same sample probability distribution with unknown parameters.*

Definition 4.3 ALTERNATIVE HYPOTHESIS (H1) *assumes that the distributions of the study outcomes are not equal. If the distribution difference is proposed to have a direction*

(a sample set, which describes a study outcome, contains either only significantly larger values, or only significantly smaller values than another sample set), H1 *is* ONE-TAILED. *If the direction does not play a role,* H1 *is* TWO-TAILED.

Because it is usually desired to show that a certain algorithm modification or improvement leads to a *higher*, and not *equal* performance, the principle of the contradiction is applied: it is assumed that H0 is true. If this suggestion can be rejected for the estimated test statistic with a certain significance level, the alternative hypothesis H1 is accepted, otherwise H0 is kept.

Table 4.1.: Errors in hypothesis testing.

	H0 is actually true	H0 is actually false
H0 is rejected	TYPE I ERROR $p = \alpha$	correct decision $p = 1 - \beta$
H0 is not rejected	correct decision $p = 1 - \alpha$	TYPE II ERROR $p = \beta$

Table 4.1 illustrates the probabilities of the two possible errors, which may occur in this approach. By the choice of the significance level α, we can reduce the danger of a type I error. The most common value is $\alpha = 0.05$. If H0 is then not rejected, it means that this was done correctly, and not by chance, with the probability of 95%.

The type II error occurs, if H0 is not rejected, but it does not actually hold. The probability of the correct rejection of H0, where H1 is indeed true, is $1 - \beta$. This value is also called the TEST POWER, since it usually corresponds to the acceptance of the desired suggestion that the two distributions of the study outcomes are unequal (see above).

The test plan should contain the following steps, according to [199, 36]:

- As exact as possible description of the problem and the corresponding data.
- Formulation of the hypotheses H0 and H1 with respect to Definitions 4.2 to 4.3.
- Choice of α as a test risk level.
- Selection of the test statistic U, which should have different distributions for H0 and H1.
- Estimation of U distribution on the evidence of H0 ($\mathcal{F}_0(U)$), which depends on the number of observations (DEGREES OF FREEDOM). Usually, these distributions are listed in the corresponding tables or are calculated by statistical software.
- Selection of the CRITICAL AREA $\mathcal{A}_C$ under $\mathcal{F}_0(U)$, which will lead to rejection of H0, if the test statistic would be in this area.
- Estimation of the test statistic value T_S.
- Rejection of H0, if $T_S \in \mathcal{A}_C$, and acceptance of H0, if $T_S \notin \mathcal{A}_C$.
- Interpretation and reporting of results.

The P-VALUE corresponds to the probability that the same or a more extreme test statistic value would be achieved by chance for a repetition of the experiment. In case of H0 rejection, p $\leq \alpha$.

Nonparametric statistical tests have several advantages over parametric statistical tests [84]:

- No assumptions about the probability distributions of the observations are required. Many parametric tests assume a Gaussian or other distribution.
- The application procedure is simple and easy to understand.
- These tests can be applied, when the sample sizes are relatively low. In our studies, we used 10 statistical repetitions for each experiment. Several references, which are mentioned in [36], suggest that the application of parametric tests requires more than 10 repetitions as the absolute minimum.
- The nonparametric tests are hardly influenced by outliers.

We apply the two following tests in this work:

- Wilcoxon signed rank test is used for paired observations, which are dependent on each other. The test statistic T_S^W is estimated, as follows:

$$T_S^W = \sum_{\forall i \in \{1,...,A\}: u_i > v_i} R^W(|u_i - v_i|), \text{ where} \tag{4.25}$$

  $\mathbf{u}$ and $\mathbf{v}$ are sample vectors of the same length A, and $R^W(\cdot)$ is a rank function, which estimates the rank of its argument from all sorted values $\{|u_1 - v_1|, ..., |u_A - v_A|\}$.

  After the calculation of T_S^W, it can be decided, if $T_S^W \in \mathcal{A}_C$. The two-tailed Wilcoxon signed rank test rejects H0, if:

$$T_S^W \geq \tau_{\alpha/2} \text{ or } T_S^W \leq \frac{A(A+1)}{2} - \tau_{\alpha/2}, \text{ where} \tag{4.26}$$

  τ_α is the critical value from the corresponding table for the Wilcoxon signed rank test.

- Mann-Whitney U-test, which is also referred to as Wilcoxon, Mann and Whitney test, compares two not paired observations. The sample vectors may have different dimensionalities. Let A be the length of $\mathbf{u}$, and B the length of $\mathbf{v}$. The test statistic T_S^U is:

$$T_S^U = \sum_{i=1}^{B} R^U(v_i), \text{ where} \tag{4.27}$$

  $R^U(\cdot)$ is the rank function, which estimates the rank of its argument from all sorted values $\{u_1, ..., u_A, v_1, ..., v_B\}$.

  The two-tailed Mann-Whitney U-test rejects H0, if:

$$T_S^U \geq \tau_{\alpha/2} \text{ or } T_S^U \leq B\,(A + B + 1) - \tau_{\alpha/2}, \text{ where} \tag{4.28}$$

  τ_α is the critical value from the corresponding table for the Mann-Whitney U-test.

The application of statistical tests for classification tasks is discussed in [150, 4].

5. Application of Feature Selection

5.1. Recognition of high-level features

One of the major targets of this work is to approach music genre and style classification with high-level features. As introduced in Section 3.3, sliding feature selection allows both the optimisation of FS and the categorisation on several levels:

- 1st level: prediction of high-level features from low-level features.
- Intermediate levels: prediction of high-level features from low-level features and high-level features from the lower levels.
- Last level: prediction of high-level categories from low-level features and high-level features.

The models for the identification of three high-level feature groups were optimised by the multi-objective evolutionary feature selection, as described in Section 3.2.4. The setups and outcomes of the related studies are briefly discussed in the next sections:

- INSTRUMENTS: The recognition of four instrument groups in polyphonic audio mixtures is described in Section 5.1.1.
- MOODS: The recognition of 8 AMG moods is outlined in Section 5.1.2.
- GFKL 2011 FEATURES: The recognition of 16 high-level features related to music theory, which were proposed in [186], is described in Section 5.1.3.

5.1.1. Instruments

The instrument recognition study was originally conducted for the work reported in [219], where the instruments were identified from intervals (2 tones played at the same time) and chords (3 and 4 tones played simultaneously). For the genre and style recognition task in Section 5.2, we selected only the chord-based models. The study overview is summarised in Table 5.1.

In the following, we explain the relevant study details:

- CLASSIFICATION TASKS: Binary recognition of four different instrument groups (guitar, piano, wind, and strings) from polyphonic mixtures of 3 and 4 instrument samples from McGill University MUMS collection [48], the RWC database [72], and the University of Iowa instrument samples[1]. First, 3,000 chords were randomly generated. 2,000 chords were reserved for the experiment set and 1,000 for the holdout

[1] http://theremin.music.uiowa.edu, date of visit: 15.02.2013.

Table 5.1.: Parameters of the instrument recognition study.

Parameter name	Values	No.
CLASSIFICATION TASKS		
Classification tasks	Guitar, Piano, Wind, Strings	4
Experiment set	2,000 chords	1
Holdout set	1,000 chords	1
FEATURES AND PROCESSING		
Initial features	1,148 audio features	1
Feature processing	NaN elimination, frame selection from attack and release interval middles and onset events for 795 features; larger building blocks for 353 features from [51]	1
Classification frames	$W_c = -1$ and $S_c = -1$	1
Feature aggregation	-	1
CLASSIFICATION METHODS		
Algorithms	C4.5, RF, NB, SVM with a linear kernel	4
OPTIMISATION PARAMETERS		
Optimisation metrics	m_{RE} and m_{SFR}	1
Optimisation algorithm	(30+1) SMS-EMOA	1
Mutation	Asymmetric bit flip with $p_{01} = 0.01$ and $\gamma = 32$	1
Crossover	No crossover, uniform or commonality-based	3
Initial feature rate	$if_r \in \{0.5; 0.2; 0.05\}$	3
Number of evaluations	2,000	1
Evaluation method	Optimisation by 10-fold CV on the experiment set; independent validation on the holdout set	1
Statistical repetitions	-	10
Number of experiments		1,440
Number of model train.		29,232,000
Number of model eval.		32,155,200

set, for the independent validation of the models after FS (see Section 4.2 for the data set descriptions).

- FEATURES AND PROCESSING: 265 mostly low-level audio descriptors were estimated from the middle frames of the attack intervals, the middle frames of the release intervals, and the interonset frames, resulting in a 795-dimensional feature vector. Another 353 characteristics were extracted from larger blocks, as described in [51]. The onset events were identified by the MIR Toolbox. If none or more than one onset was estimated, we selected the frame with the highest RMS energy as an onset frame. Therefore, each chord provided exactly one onset event. The classification window size W_c and the step size S_c were set to -1 (this setting corresponds to the building of the classification frames from complete audio recordings). Because only one feature value could be extracted from each chord through the aforementioned procedure, no further feature aggregation was required.

- CLASSIFICATION METHODS: Four classifiers were used for model training: decision

tree C4.5, random forest, naive Bayes and support vector machines with a linear kernel (see Section 2.4 for the description of algorithms). The default parameters of these methods were used as provided in RapidMiner [147]. For the explanation, why we selected these algorithms, see the last paragraphs of Section 2.4.

- OPTIMISATION PARAMETERS: The relative error m_{RE} (Equ. 4.10)[2] from the 10-fold cross-validation on the experiment set[3] and the selected features rate m_{SFR} (Equ. 4.24) were optimised.

  SMS-EMOA with asymmetric bit flip mutation ($p_{01} = 0.01$ and $\gamma = 32$), as described in Section 3.2.4, was used as optimisation method. These parameters were selected after a preliminary study and experiments from [217]. Three different crossover possibilities (no crossover, uniform, and commonality-based) and three different values for the initial feature rate $if_r \in \{0.5; 0.2; 0.05\}$ were tested. The number of evaluations was set to 2,000 after the preliminary experiments.

  The overall number of the statistical repetitions was set to 10. This limitation was necessary because of the long experiment runtimes between several hours and several days. The last lines in Table 5.1 illustrate the high computational load. The number of model trainings was equal to: $10 \cdot 30$ (training of models based on 10-fold CV on the experiment set for the initial population of 30 individuals) + $10 \cdot 2,000$ (the same procedure for each offspring solution during 2,000 EA loop steps) = 20,300 for each experiment. Since 10 statistical repetitions for 144 combinations of parameter settings were run, the overall number of model trainings is $20,300 \cdot 10 \cdot 144$ = 29,232,000. Similarly, the number of model validations for each experiment was equal to: $10 \cdot 30 + 30$ (independent validation of the initial solutions on the holdout set) + $10 \cdot 2,000 + 2,000$ (validation of the offspring solutions on the holdout set) = 22,330.

  It should be mentioned that the number of model trainings and evaluations provides a very rough effort measurement for many reasons: NB experiments were more than 10 times faster as SVM, and training and classification with fewer features for $if_r = 0.05$ was significantly faster at the beginning of the corresponding experiments than for $if_r = 0.5$. Also, the number of classification instances in the data sets is different for this study and the studies described in Sections 5.1.2 to 5.2, so that it is not possible to exactly compare the computing efforts across all studies.

Figure 5.1 shows the best non-dominated fronts of the final solutions after the experiments. The holdout m_{RE} is plotted on the horizontal axis and m_{SFR} on the vertical axis. C4.5 solutions are marked with blue circles, RF with red squares, NB with green diamonds, and SVM with yellow triangles.

The trade-off solution fronts can be observed: the smallest m_{RE} is achieved by the models with the largest numbers of features, and on the other side there exist models, which are built by a very limited numbers of features. These boundary solutions may be sometimes less promising for a decision maker. However, several models closer to the upper left side

[2] In [219], we refer to the mean squared error - the implemented metric was indeed MSE, but for this study setup it is equal to m_{RE}.

[3] For simplicity reasons, we denote the metrics which were estimated as an average from the n CV folds by the same symbols as the metrics estimated by a single model building and evaluation (both approaches are illustrated in Fig. 4.2, Section 4.2).

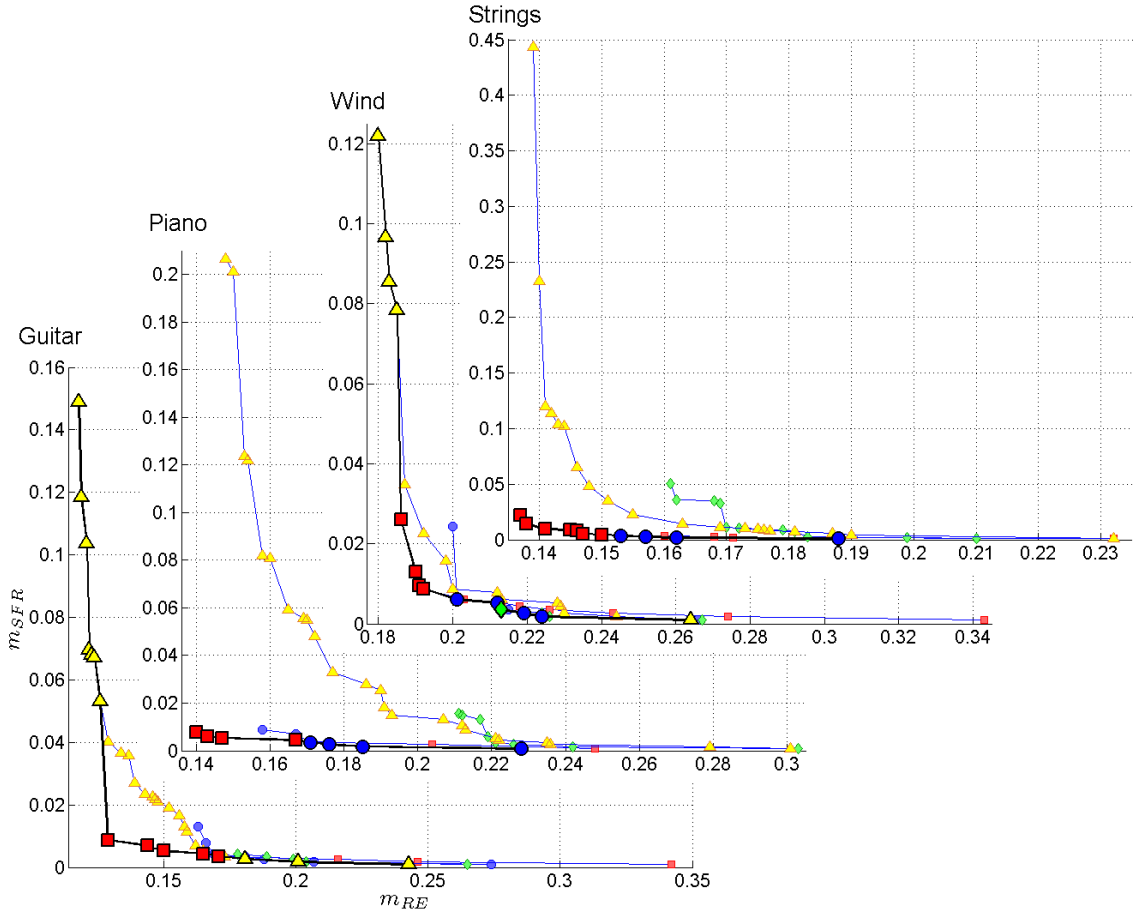

Figure 5.1.: The best ND fronts after all instrument recognition experiments. Circles: C4.5, squares: RF, diamonds: NB, triangles: SVM. The ND fronts for each classifier are indicated with thin lines. The ND fronts across all classifiers are indicated with thick lines, and the markers of the corresponding models are enlarged.

of the ND front can be indeed relevant. Consider, for example, the upper right subfigure (Strings category), the two SVM solutions in the upper left corner of the subplot (marked with triangles): it is possible to reduce the amount of features from 44.34% to 23.26%, keeping almost the same m_{RE}, which increases from 0.139 to 0.14. This situation illustrates very well that the number of features can be strongly decreased, where the classification quality remains almost the same. Models with less features and a slightly diminished classification performance can be even more preferable: we already discussed in Section 3.1 that the models built from smaller feature sets allow faster training and classification, save storage demands, and have a reduced tendency to be overfitted.

The classification models built by RF and SVM have the largest share of the overall ND solutions across all classifiers (this front is marked with the thick line). But there also exist some NB and C4.5 solutions, which belong to the best ND front, and are not dominated by any RF or SVM solution. This means that it is reasonable to combine different classification methods. This observation was measured in terms of hypervolume contributions of different classification methods in [215].

It is also important to mention that the solutions, which are plotted in Fig. 5.1, may not correspond to the very best trade-off solutions, because the validation was done on the independent holdout set. However, we measured the generalisation performance of the models in [219], and it was very similar for the optimisation and the holdout sets. Another derived suggestion is that some solutions from earlier generations provide better classification results on the holdout set than the final solutions after 2,000 evaluations. This effect may be reduced by early stopping, as discussed in [124] and mentioned in Section 3.4.1. However, it is then required to provide an additional outer validation loop, and to further increase the already very high number of model training and validation steps (see above the explaination of the last three lines from Table 5.1). These experiments were not possible within the scope of this thesis, but are reasonable in future.

5.1.2. Moods

The parameters of the mood study are listed in Table 5.2.

In the following, we describe only the parameters of the mood recognition study, which differ from the setup of the instrument recognition study already discussed in Section 5.1.1.

- Classification tasks: For our music database of 120 albums, listed in Appendix B, Table B.1, we labelled the songs with the corresponding AMG mood categories[4]. These moods are defined by music experts and can be treated as personal preferences. Because some categories had very small numbers of the labelled songs in our collection, we selected the following eight moods after the preliminary analysis: Aggressive, Confident, Earnest, Energetic, PartyCelebratory, Reflective, Sentimental, and Stylish.

  A problematic issue of this ground truth is that only the positive labels are available from the AMG web site. If an album is not labelled with a certain mood, it could mean that it is either a negative example or that it has not been analysed by the experts. On the other side, subjective descriptors such as moods almost always cannot guarantee a precise ground truth.

  The training sets were generated as follows: we selected all available albums with a certain mood tag and drew randomly one song per album. Then, the same number of songs was drawn randomly from the remaining albums, which were not labelled with this mood. The classification models were trained on these balanced sets. The solutions generated by SMS-EMOA were evaluated on the song set OS120, and validated independently on the song set TS120 (both sets are listed in Appendix B, Table B.3), as also done in previous studies [15, 217, 218]. During the random generation of the training sets, the songs from OS120 and TS120 were excluded, so that the number of the shared songs for all sets (training, optimisation and holdout) was equal to zero.

- Features and processing: For 439 low-level and high-level descriptors, which have been originally extracted from short frames with length < 4 s, mean and standard deviation were estimated for classification frames with $W_c = 4$ s and $S_c = 2$ s.

[4] http://www.allmusic.com/moods, date of visit: 15.02.2013.

Table 5.2.: Parameters of the mood recognition study.

Parameter name	Values	No.
CLASSIFICATION TASKS		
Classification tasks	Aggressive, Confident, Earnest, Energetic, PartyCelebratory, Reflective, Sentimental, Stylish	8
Training sets	30–52 songs	1
Optimisation set	120 songs	1
Holdout set	120 songs	1
FEATURES AND PROCESSING		
Initial features	1,318 audio features	1
Feature processing	NaN elimination, normalisation, interonset frame selection	1
Classification frames	$W_c = 4$ and $S_c = 2$	1
Feature aggregation	Mean and std. deviation for low-level features and high-level features with extraction windows < 4 s	1
CLASSIFICATION METHODS		
Algorithms	C4.5, RF, NB, SVM with a linear kernel	4
OPTIMISATION PARAMETERS		
Optimisation metrics	m_{BRE} and m_{SFR}	1
Optimisation algorithm	(50+1) SMS-EMOA	1
Mutation	Asymmetric bit flip with $p_{01} = 0.01$ and $\gamma = 32$	1
Initial feature rate	$if_r \in \{0.5; 0.2; 0.05\}$	3
Number of evaluations	2,000	1
Evaluation method	Optimisation on the optimisation set; independent validation on the holdout set	1
Statistical repetitions	-	10
Number of experiments		960
Number of model train.		1,968,000
Number of model eval.		3,936,000

Therefore, the number of feature dimensions was increased by the factor 2 leading to 878 features.

Another set of 70 low-level and high-level features with extraction frames larger than 4 s was processed directly without aggregation.

A next group of features was integrated according to the concept of sliding feature selection, as introduced in Section 3.3. Different instrument categorisation models described in Section 5.1.1 were applied on extraction windows around the onset events. Then, the relative share of the positive outcomes (an instrument was detected) was calculated for larger high-level feature extraction frames of 10 s. For example, a piano share of 0.8 in a frame means that 80% of the binary classification models identified a piano around the onsets in the analysed 10 s frame. Because all non-dominated instrument models for different classifiers were taken into account (see Fig. 5.1), the number of these instrument-related features was equal to 237.

Finally, the 133 structural complexity high-level characteristics listed in Table A.7 were also integrated into the complete feature set.

All features were normalised and the missing values were replaced by the medians. For features with short extraction frames, only the interonset frames were taken into account.

- CLASSIFICATION METHODS: For the mood recognition study and the following experiments described in Sections 5.1.3 to 5.2, we used the same 4 classifiers as for instrument recognition. The only change was the increased number of trees for the RF classifier – we replaced the default value of 10 trees by 100 according to the observations from [218].
- OPTIMISATION PARAMETERS: Because the optimisation (and holdout) sets were not balanced, we used the balanced relative error m_{BRE} (Equ. 4.12) and m_{SFR} as optimisation criteria. The parameters of SMS-EMOA were the same as for the instrument recognition study, except for the population size (it was increased to 50) and crossover: because this operator did not provide any significant improvement for all instrument recognition tasks, we removed it from the algorithm.

  For mood recognition, we did not use a 10-fold CV process for model training and validation, because of two reasons. First, 10-fold CV requires approximately 10 times longer runtimes than the single validation, and we had to select a compromise between the number of classification tasks and other parameter settings. This load could be in principle partly reduced by using, e.g., only a 3-fold CV procedure, as it was done for the GFKL2011 set recognition (Section 5.1.3). However, the second restriction was the limited number of positive mood albums (between 15 and 26), so that the balanced training sets consisted of 30–52 songs. Using only 2/3 of these sets for the model training would further decrease the number of positive songs used for the model creation.

Figure 5.2 illustrates the ND fronts of the final solutions after the experiments. It can be clearly observed that mood recognition tasks are more complex than instrument identification in polyphonic mixtures. It depends on the ground truth, which is not so precise, as for instrument recognition task: as discussed above, it cannot be guaranteed that negative examples are always really negative.

The share of each classification method in the overall-classifier ND front varies from category to category. NB provides the smallest m_{BRE} for the categories Aggressive and Sentimental, RF for the five other categories, and SVM only for Reflective. C4.5 contributes only seldom to the overall ND front.

It is important to explain why we did not use any established song database for the better comparison with other studies. Unfortunately, at least at the time point, when we started our studies, these databases had (and still have) several limitations:

- Several databases do not contain complete songs. For examples, GTZAN[5] consists of 30 s song excerpts, and the Music Audio Benchmark Data Set [85] of 10 s excerpts. However, a part of our studies was to examine different processing methods, starting with features from complete songs (see [221]). Another motivation for the $\mathcal{FE}$ from complete songs is that it is not straightforward to decide, which part of a song should

[5] http://marsyasweb.appspot.com/download/data_sets/, date of visit: 15.02.2013.

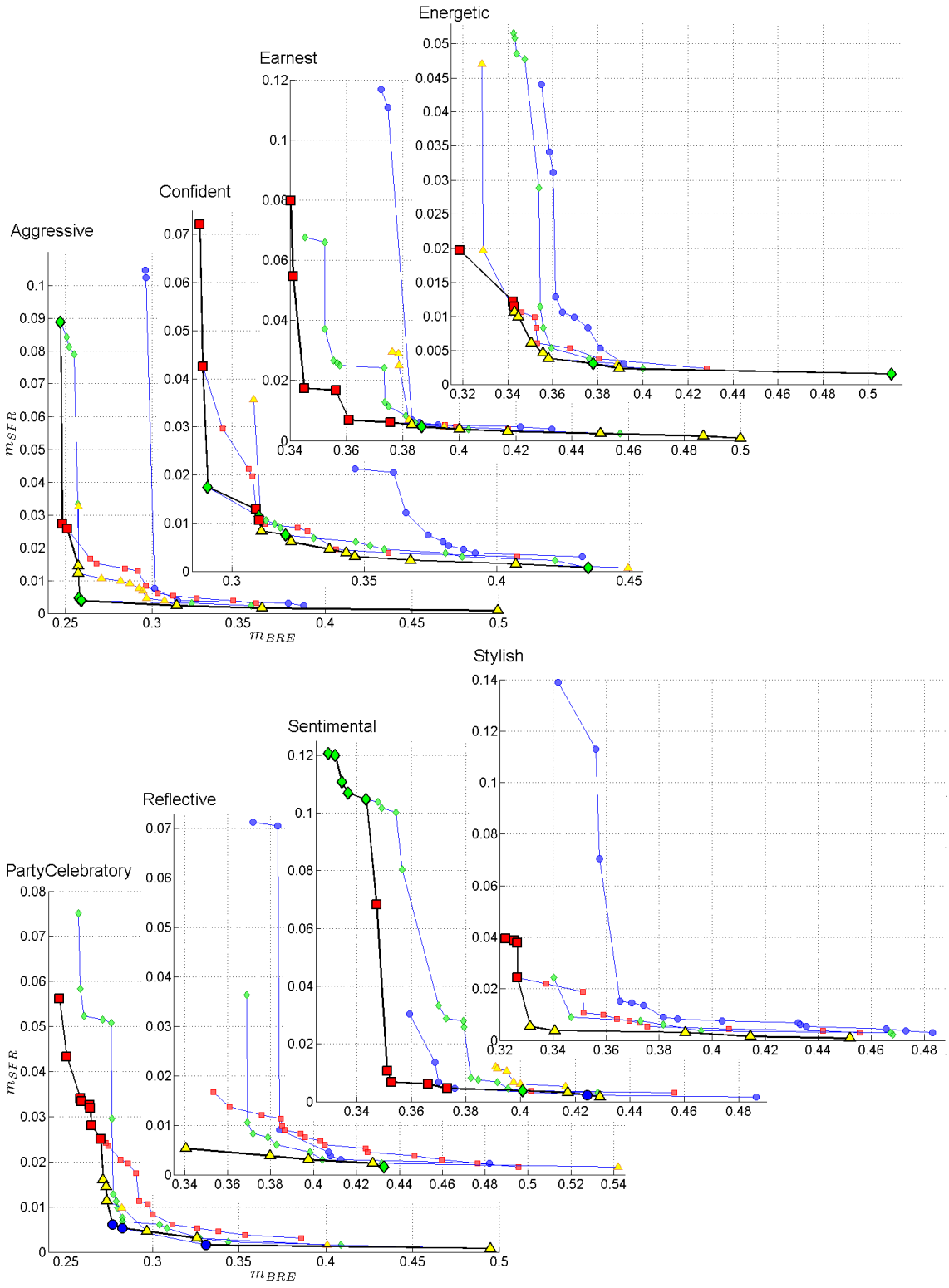

Figure 5.2.: The best ND fronts after all mood recognition experiments. Circles: C4.5, squares: RF, diamonds: NB, triangles: SVM. The ND fronts for each classifier are indicated with thin lines. The ND fronts across all classifiers are indicated with thick lines, and the markers of the corresponding models are enlarged.

be 'representative'. For example, in a study of Chinese pop songs, the representative segments marked by the listeners which understood Chinese were different from the representative sections marked by the remaining listeners [28]. Some genres, like progressive rock, contain parts with varying properties (orchestra, longer segments with distorted guitars without any vocals, vocal segments, etc.), and it would not be advantageous to restrict the feature extraction interval to, e.g., 30 seconds from the song middle.

- Free music data sets, such as RWC Magnatune[6], are often biased toward several genres and do not represent the popular commercial music well.
- Databases with large lists of commercial pop songs, such as USPOP[7] or SLAC dataset[8], contain only a limited number of features. It is not possible to extract self-implemented characteristics, and it is expensive to buy a large collection of songs.

5.1.3. GFKL 2011 features

The parameters of the GFKL 2011 study are summarised in Table 5.3.

The recognition of the GFKL 2011 high-level characteristics was done with the same input features as for the mood recognition study. However, there were some differences:

- Classification tasks: The task of our study presented at the Annual Conference of the German Classification Society (GFKL) [186] was to identify relevant high-level musical characteristics, if a personal music category was defined with a very limited number of positive song examples, in that case 5. This was motivated by the real-world situation, where a music listener does not want to select too many training songs for each new category and, in particular, to choose negative examples. A set of 61 binary high-level features with music theory background was designed, as illustrated in Fig. 5.3: instrumentation, singing, speech, melody, harmony, rhythm, dynamics, effects, structure, and level of activation. The experiments underlined that many of these high-level descriptors were indeed relevant for the user-defined categories, since the feature distribution differed strongly from a random binary distribution. In the second part of the study, these features were predicted with different success from a large set of mostly low-level audio characteristics, without feature selection.

  Because of the high number of GFKL 2011 characteristics, it was not possible to apply EMO-FS for all categories. Another limitation was that many of the high-level features had too imbalanced distributions. Therefore, we selected 16 characteristics with the most balanced distributions of the positive and negative songs (so that the share of the positive songs was between approximately 30% and 70%): Activation level high, Effects distortion, Harmony major, Harmony minor, Instrumentation drums, Melodic range $\leq$ octave, Melodic range $>$ octave, Melodic range linearly, Melodic range volatile, Singing solo clear, Singing solo man, Singing solo polyphonic, Singing solo, Singing solo unison, Singing solo woman, and Singing voice medium.

[6] http://www.music-ir.org/mirex/wiki/2005:Audio_Genre_Classification, date of visit: 15.02.2013.
[7] http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html, date of visit: 15.02.2013.
[8] http://jmir.sourceforge.net/Codaich.html, date of visit: 15.02.2013.

Table 5.3.: Parameters of the GFKL2011 recognition study.

Parameter name	Values	No.
CLASSIFICATION TASKS		
Classification tasks	Activation level high, Effects distortion, Harmony major, Harmony minor, Instrumentation drums, Melodic range $\leq$ octave, Melodic range $>$ octave, Melodic range linearly, Melodic range volatile, Singing solo clear, Sing. sol. man, Sing. sol. polyphonic, Sing. sol. rough, Sing. sol. unison, Sing. sol. woman, Singing voice medium	16
Experiment set	57 songs	1
Holdout set	31 songs	1
FEATURES AND PROCESSING		
Initial features	1,318 audio features	1
Feature processing	NaN elimination, normalisation, interonset frame selection	1
Classification frames	$W_c = 4$ and $S_c = 2$	1
Feature aggregation	Mean and std. deviation for low-level features and high-level features with extraction windows < 4 s	1
CLASSIFICATION METHODS		
Algorithms	C4.5, RF, NB, SVM with a linear kernel	4
OPTIMISATION PARAMETERS		
Optimisation metrics	m_{BRE} and m_{SFR}	1
Optimisation algorithm	(50+1) SMS-EMOA	1
Mutation	Asymmetric bit flip with $p_{01} = 0.01$ and $\gamma = 32$	1
Initial feature rate	$if_r \in \{0.2; 0.05\}$	2
Number of evaluations	2,000	1
Evaluation method	Optimisation by 3-fold CV on the experiment set; independent validation on the holdout set	1
Statistical repetitions	-	10
Experiment number		1,280
Number of model train.		7,872,000
Number of model eval.		10,496,000

The main problematic issue of this approach is that some of the features described only parts of songs – for example, the singing characteristics – but the classification windows were automatically created from the complete songs. The optimal solution would be to analyse the songs by experts and to exactly denote the first and the last occurrences of each instrument, effect, etc. This was not possible because of too high efforts. However, these features can be indeed understood (similar to moods) as subjective personal preferences or tags, e.g., 'songs with a large melodic range' or 'songs with several singers'. Also, a part of these characteristics was relevant for almost the complete recordings, such as activation level or drums.

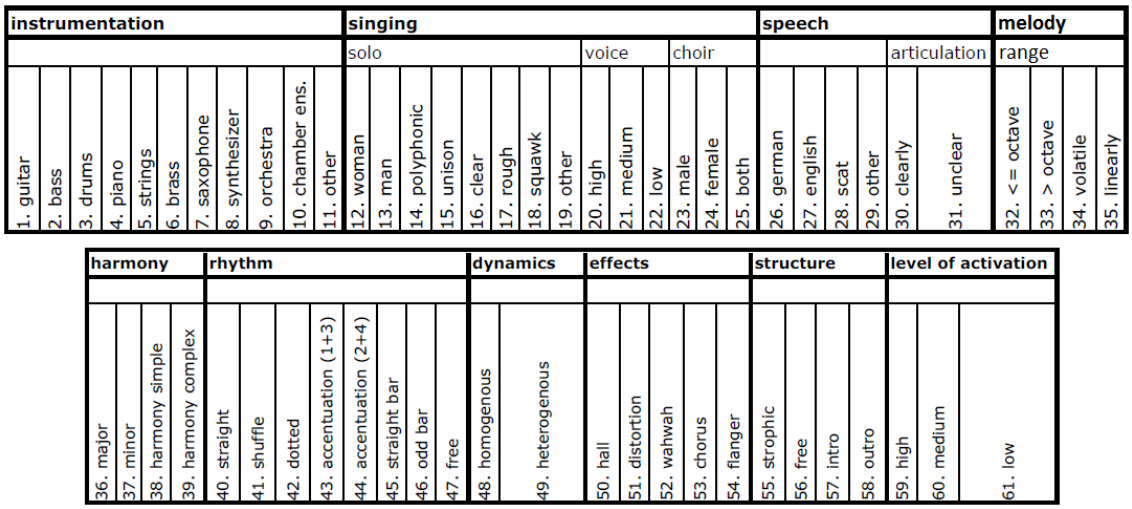

Figure 5.3.: List of GFKL2011 features from [186].

- Optimisation parameters: To run enough experiments for the large number of 16 classification categories, we had to limit $if_r \in \{0.05; 0.2\}$. It was acceptable because of the high overall number of input features: the initialisation with only 5% of the complete feature set corresponds to approximately 66 features.

  Since the number of positive examples in the training sets was larger than for the mood training sets, we decided to create an experiment set with 57 songs, and to apply 3-fold CV for model training and evaluation. 31 songs were kept out for the holdout set.

Figures 5.4 to 5.5 plot the ND fronts of the final solutions after optimisation. As it could be expected, some of the GFKL 2011 high-level categories are hard to predict – one reason discussed above is that classification windows are built from the complete music pieces. The precise ground truth, which marks the start and end positions of the related intervals, could improve the performance, but requires very high efforts by music experts, which have to label this information for each high-level feature and each song. Another challenge of this setup is that the holdout and optimisation sets contain not only different songs, but also songs from different albums. However, the classification errors are significantly smaller than errors, which would be provided by a random classifier, and are below 0.3 for three categories.

5.2. Recognition of genres and styles

The overview of the genre and style recognition study is provided in Table 5.4.

The recognition of genres and styles was our last EMO-FS study, and corresponded to the last step of sliding feature selection. The goal was, in particular, to evaluate the ability of the classification based on high-level features to achieve the results which are comparable to the classification based on low-level feature sets. In general, we cannot expect that audio-based high-level characteristics would allow a better class separability because of several reasons:

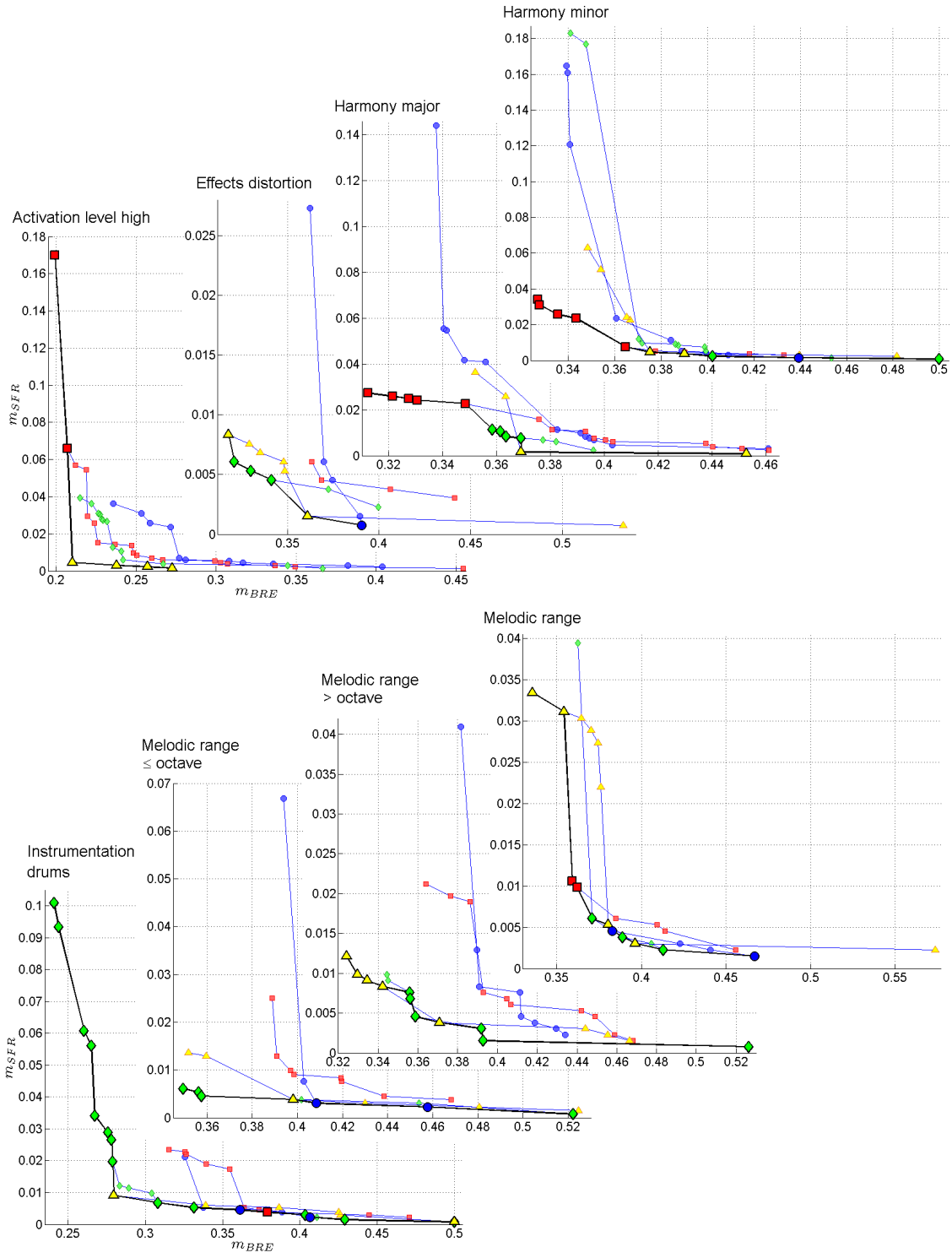

Figure 5.4.: The best ND fronts after all GFKL 2011 recognition experiments – the first 8 classification tasks. Circles: C4.5, squares: RF, diamonds: NB, triangles: SVM. The ND fronts for each classifier are indicated with thin lines. The ND fronts across all classifiers are indicated with thick lines, and the markers of the corresponding models are enlarged.

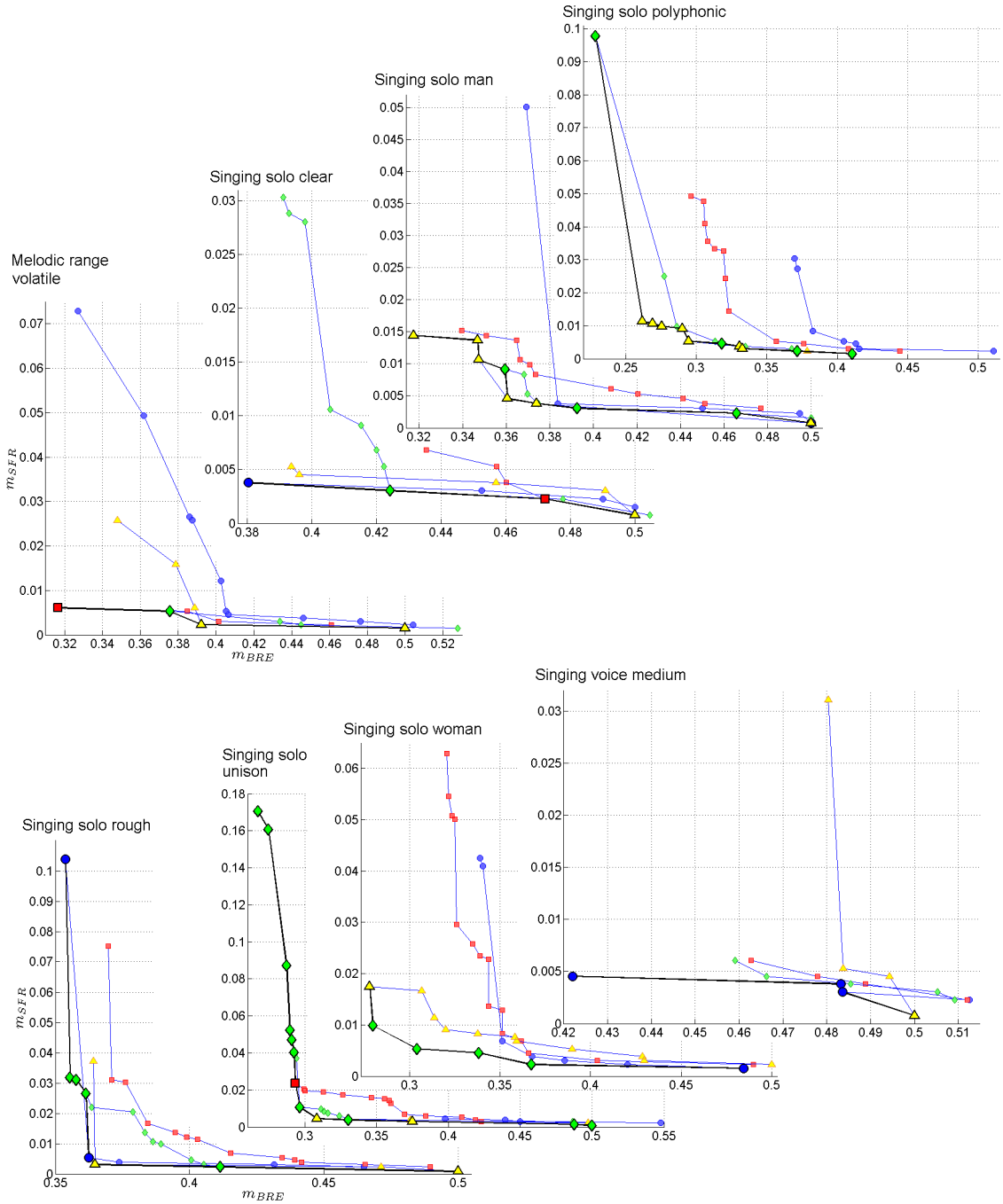

Figure 5.5.: The best ND fronts after all GFKL 2011 recognition experiments – the last 8 classification tasks. Circles: C4.5, squares: RF, diamonds: NB, triangles: SVM. The ND fronts for each classifier are indicated with thin lines. The ND fronts across all classifiers are indicated with thick lines, and the markers of the corresponding models are enlarged.

Table 5.4.: Parameters of the genre and style recognition study.

Parameter name	Values	No.
CLASSIFICATION TASKS		
Classification tasks	Genres: Classic, Pop, Rap Styles: ClubDance, HeavyMetal, ProgRock	6
Training sets	20 songs, listed in Table B.2, Appendix B	1
Optimisation set	120 songs, listed in Table B.3, Appendix B	1
Holdout set	120 songs, listed in Table B.3, Appendix B	1
FEATURES AND PROCESSING		
Initial features	the LL set (636 features) and the HL set (566 features)	2
Feature processing	NaN elimination, normalisation, interonset frame selection	1
Classification frames	$W_c = 4$ and $S_c = 2$	1
Feature aggregation	Mean and std. deviation for low-level features and high-level features with extraction windows < 4 s	1
CLASSIFICATION METHODS		
Algorithms	C4.5, RF, NB, SVM with a linear kernel	4
OPTIMISATION PARAMETERS		
Optimisation metrics	m^s_{BRE} and m_{SFR}	1
Optimisation algorithm	(50+1) SMS-EMOA	1
Mutation	Asymmetric bit flip with $p_{01} = 0.10$ and $\gamma = 32$	1
Initial feature rate	$if_r \in \{0.5; 0.2\}$	2
Number of evaluations	3,000	1
Evaluation method	Optimisation on the optimisation set; independent validation on the holdout set	1
Statistical repetitions	-	10
Number of experiments		960
Number of model train.		2,928,000
Number of model eval.		5,856,000

- As already discussed in Section 2.2.1, in some studies it is argued that a very short audio time interval is enough to recognise a genre or artist, so that this recognition task could be solved well with short-frame characteristics which are usually low-level.
- The performance of the classification based on up-to-date high-level audio features is in many cases still lower than the classification based on score-based high-level features: for example, the recognition of instruments in polyphonic recordings is a very hard task, especially, if the songs to classify contain many instruments which have not been used for training.
- High-level features derived from low-level using sliding feature selection do not contain new information.
- The boundary between low-level and high-level features is not always very clear (see Section 2.2.1).
- Many high-level features are strongly correlated with certain low-level features: for

example, the tempo with the statistics of autocorrelation or the share of percussions with characteristics of the phase domain [146]. Therefore, redundant feature groups of low-level and high-level characteristics are theoretically possible.

However, one of the major goals of our work was to provide a better interpretability of the selected feature subsets. Only the characteristics which are understandable by humans and are related to music theory may provide a deeper insight into the description of genres, styles, and personal preferences. For listeners which are experienced with music theory it is not hard to name a few most important high-level features, which represent a certain category. A very simple example is that a large share of vocals, distorted guitars, and a small variation of harmonics would well distinguish punk rock from jazz. Or, the presence of a bag pipe may be helpful for the identification of a folk rock subgenre.

A composer does not bear any low-level signal-based features in mind during the creation of a new music piece. These features even could not be extracted until the recent past, if we think about many centuries of the music notation (cf. Section 2.1.2), before analogue and digital recording became possible. Composing a music piece means to follow many rules, which vary from genre to genre: 'avoid parallels of the fifths', 'do not use too many non-harmonic notes', 'cadence as a final highlight of a piece', 'electric amplification of the guitar sound', etc. A skilled composer is aware of these rules, but they are often not so evident for a common music listener, and are not by default integrated into automatic recommendation systems. However, it is very promising to recommend music with a more comprehensible background.

In that case, FS may provide a solid way to sort out a large amount of irrelevant and less relevant characteristics, limiting the feature set to a small number of interpretable features, which are very well suitable for the separation of a certain genre or a category. The distribution of songs in a music collection plays an important role: the separation of punk rock in a collection of only punk rock and jazz songs may work perfectly using only a couple of features. The separation of punk rock in the collection of pop rock, folk rock, alternative rock, and crossover songs would require more features and more complex models.

The following details of the study on genre and style recognition differ from the aforementioned experiments:

- Classification tasks: Similar to the mood recognition task, which is described in Section 5.1.2, we use our music collection of 120 albums. However, the training sets are defined by only 10 positive and 10 negative examples, as proposed in our very first experiments in music classification [205]: a listener would be tired of selecting tens and hundreds of songs to learn of a single category.

  Three AMG genres (Classic, Pop, Rap), and three styles (ClubDance, HeavyMetal, ProgRock) are used as classification categories.

- Features and processing: Two feature sets are compared for genre and style recognition. LL set (636 features) is a baseline set with only low-level features (see Tables in Appendix A), and HL set (566 features) is a set with only high-level characteristics. 258 of them are the short-framed harmony features. A further part contains 239 long-framed features, where 224 of them have been derived with the help of sliding feature selection and the subsequent identification of instruments, moods,

and GFKL 2011 features. For each classifier, we applied the two classification models with the smallest classification errors from the non-dominated fronts. The last part comprises 70 structure complexity features. Compared to the mood and GFKL 2011 feature recognition studies, we *removed* timbre, chroma, and chroma related complexities (structural complexity feature groups are listed in Table A.7), because they cannot be clearly described as high-level features. Other structural complexity groups (chord, instruments, harmony, tempo and rhythm) indeed describe the variation of several high-level characteristics for large song analysis frames.

- Optimisation parameters: We increased the number of SMS-EMOA evaluations to 3,000 for a more extensive search of relevant features and set $if_r \in \{0.5; 0.2\}$. $if_r = 0.05$, as done in other studies, was omitted: the complete number of features in the LL and HL sets was lower than for other studies, so that the start with a higher number of random features, together with the asymmetric mutation, was preferable to increase the explorative ability of FS. The classification quality optimisation criterion m^s_{BRE} was estimated on the song level (see Equ. 4.21).

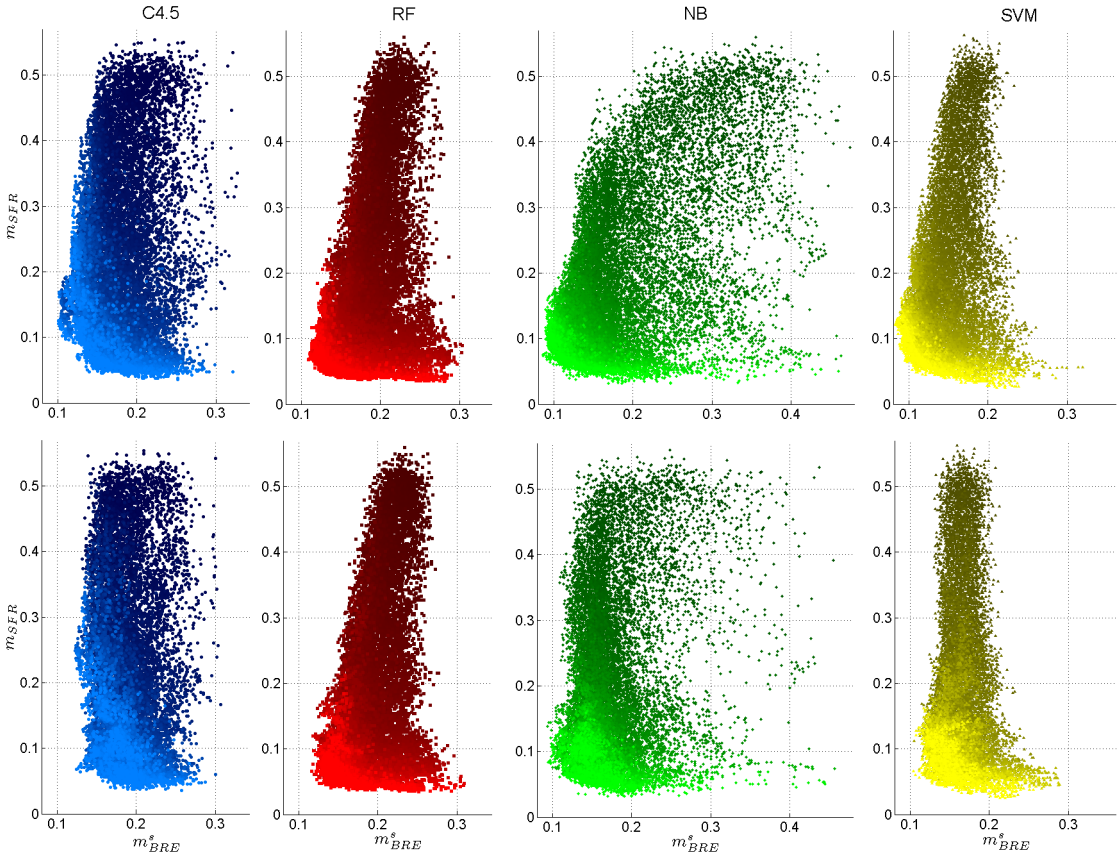

Figure 5.6.: All feature subset solutions found during 10 statistical repetitions of FS optimisation, category ProgRock, $if_r = 0.5$, the LL set. Top subfigures: m^s_{BRE} on the optimisation set. Bottom subfigures: m^s_{BRE} on the holdout set. Subfigure columns from left to right correspond to classification methods: C4.5, RF, NB and SVM.

Figure 5.6 illustrates the objective space and plots all solutions, which were found during

10 statistical repetitions, optimising FS for category ProgRock and using the LL feature set with $if_r = 0.5$. The subfigure columns correspond to the classification methods, from left to right: C4.5, RF, NB and SVM. The upper row shows the m^s_{BRE} on the optimisation set, and the lower row m^s_{BRE} on the holdout set. The older solutions are denoted by the darker colour markers, and the newer solutions by the brighter colour markers.

Though we show here only the figure for the ProgRock solutions, several observations can be also stated for the other categories:

- The optimised solutions do not have the same performance on the optimisation and holdout sets. But this difference is not very strong, and the regions with the newer solutions, marked with the brighter markers, are close to the non-dominated fronts for both the optimisation and holdout sets, so that an acceptable generalisation ability is provided. This tendency is also visible in Fig. 5.7, where the progress of the hypervolumes for the optimisation set (left subfigure), and the holdout set (right subfigure) is plotted. For all classifiers, the mean dominated hypervolumes increase for the holdout set, and the final mean holdout hypervolumes are only slightly lower than for the optimisation set.

- An increase of the number of features leads to a smaller m^s_{BRE} for the solutions around the non-dominated fronts. After a certain number of features is achieved, the classification performance suffers from a further increasing number of features. This means that the classification methods have their limitations when dealing with a high number of irrelevant or redundant features, and FS becomes essential for large feature sets.

- A very large proportion of solutions is dominated by a rather small number of solutions in the non-dominated fronts. This holds for all classifiers and categories.

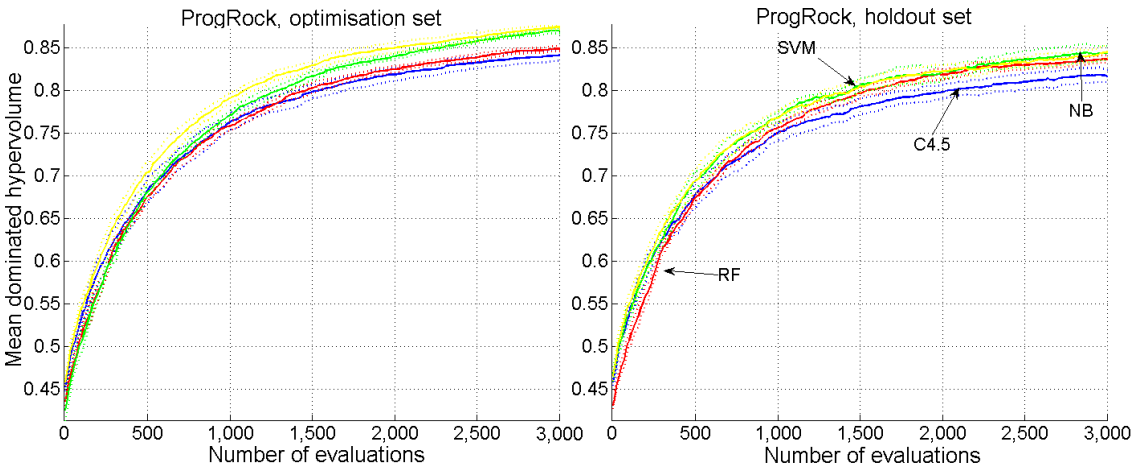

Figure 5.7.: Increase of the mean hypervolume on the optimisation set (left subfigure), and on the holdout set (right subfigure), during 3,000 SMS-EMOA evaluations for ProgRock, $if_r = 0.5$, the LL set. 95% confidence intervals are marked with lines of different colours. C4.5: blue, RF: red, NB: green, SVM: yellow.

In the next sections, we discuss several aspects of the study outcomes in more detail:

- In Section 5.2.1, the results of FS with the LL set are discussed. The increase of hypervolume is measured and confirmed by statistical tests. The classification based on feature subsets with the smallest m^s_{BRE} is compared to the classification based on the complete feature sets.
- Section 5.2.2 describes the same investigations for FS with the HL set.
- In Section 5.2.3, we compare the classification performance of the LL set and the HL set w.r.t. hypervolume and the solutions with the smallest m^s_{BRE}.
- Finally, Section 5.2.4 presents an analysis of the often selected high-level feature groups and high-level features for different classification categories.

5.2.1. Low-level feature selection

Figure 5.8 plots the non-dominated fronts of the final solutions after 3,000 SMS-EMOA generations. The identification of the classical music pieces is the simplest categorisation task: the lowest m^s_{BRE} is 0.0113 (classified with RF), and all solutions of the overall ND front, except for one C4.5 model, have $m^s_{BRE} < 0.05$. The most challenging categories are ClubDance (the smallest $m^s_{BRE} = 0.1442$) and Pop (the smallest $m^s_{BRE} = 0.1236$). However, in our opinion, these results are also promising for these hard to classify styles.

As it was stated in the other studies, the all-classifier ND front contains solutions of several classification methods, and for all categories at least three of four different classifiers contribute to this front. Also, non-dominated solutions with lowest m^s_{BRE} values are created by different classifiers across the tested categories. This strengthens the suggestion that it is reasonable to include several classification algorithms into genre and style classification.

For a general evaluation of EMO-FS, it is necessary to measure the INCREASE OF THE MULTI-OBJECTIVE PERFORMANCE between the first and last generations of SMS-EMOA. This can be done by the estimation of the mean hypervolume $\mathcal{S}$ on the holdout set across 10 statistical repetitions before and after optimisation. The increase of hypervolume on the holdout set means that the models built with the optimised feature subsets are better generalisable and also perform well on data which have been neither involved in model training nor their validation during the optimisation process.

As plotted in Fig. 5.9, it can be clearly observed that the dominated hypervolume increases. Here, its progress is measured in per cent, related to the initial dominated hypervolume on the holdout set. We denote the mean initial dominated hypervolume on the holdout set by $\overline{\mathcal{S}}^H_{init}$, and the mean final dominated hypervolume on the holdout set by $\overline{\mathcal{S}}^H_{fin}$. The larger markers correspond to the experiments with $if_r = 0.5$, and the smaller markers to the runs with $if_r = 0.2$. C4.5 is marked with blue circles, RF with red squares, NB with green diamonds, and SVM with yellow triangles. The categorisation tasks are separated by thick vertical lines. Because the experiments with $if_r = 0.2$ already start with smaller feature sets than the experiments with $if_r = 0.5$, the increase of hypervolume is not so high. The increase of the mean dominated holdout hypervolume during the optimisation is approximately the same for all categories in spite of their different complexity.

The increase of the hypervolume on the holdout set after the optimisation is confirmed as being significant in all cases by the Wilcoxon signed rank test for the following test setup:

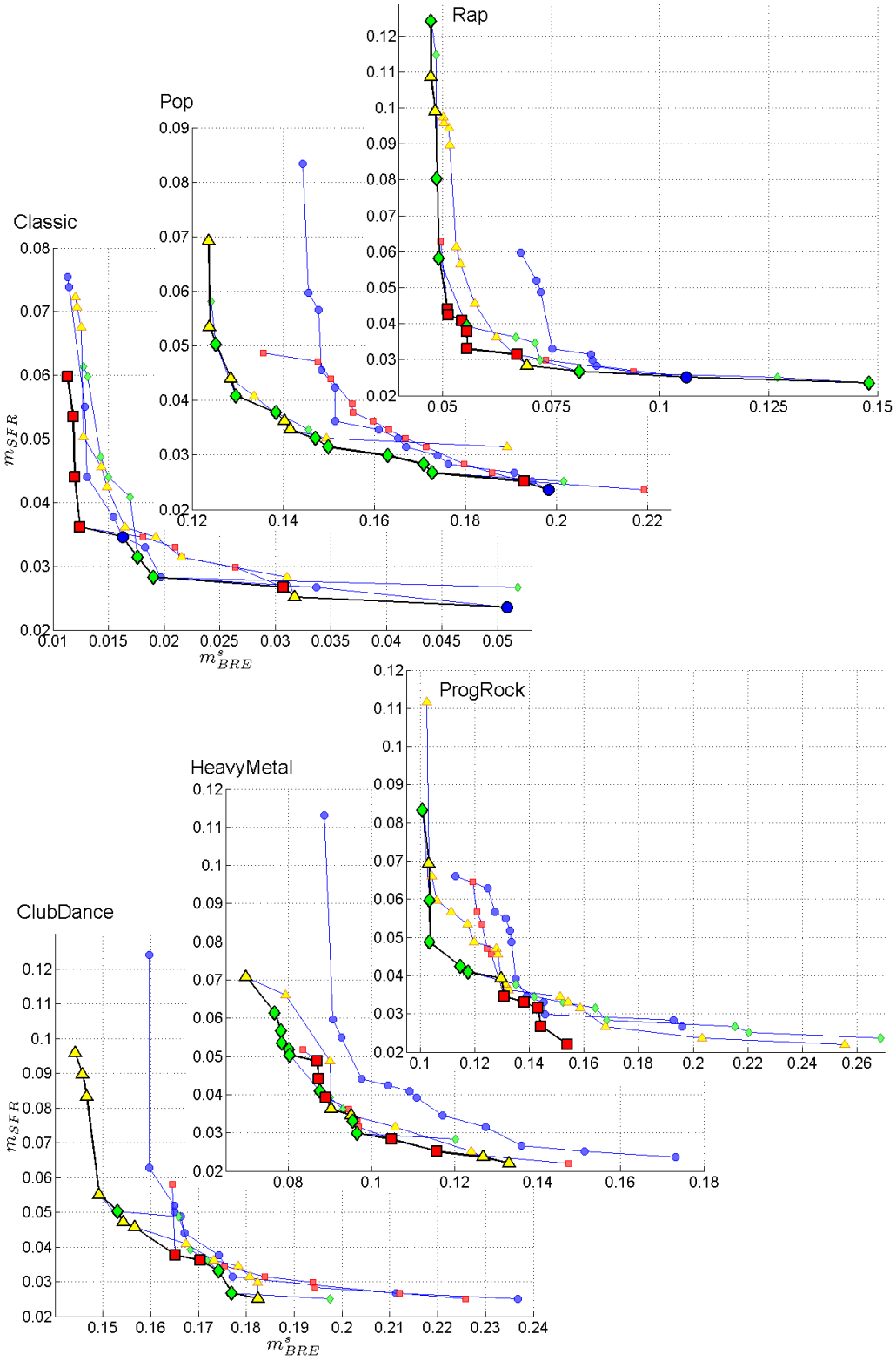

Figure 5.8.: The best ND fronts after genre and style recognition with the LL set. Circles: C4.5, squares: RF, diamonds: NB, triangles: SVM. The ND fronts for each classifier are indicated with thin lines. The ND fronts across all classifiers are indicated with thick lines, and the markers of the corresponding models are enlarged.

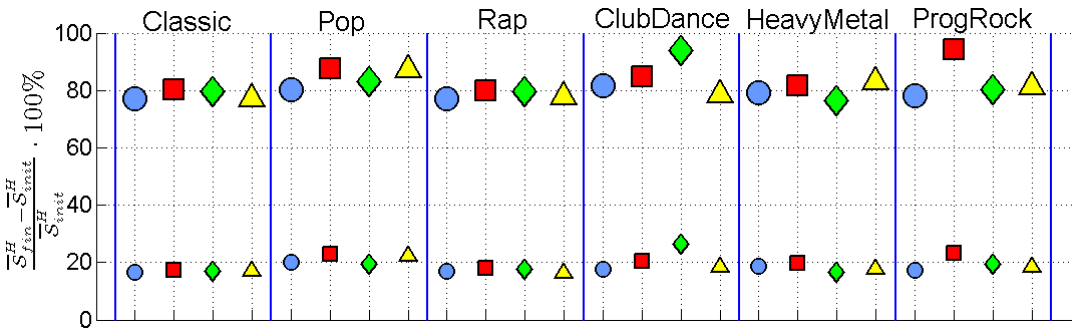

Figure 5.9.: Increase of the relative mean holdout dominated hypervolume after the optimisation. Circles: C4.5, squares: RF, diamonds: NB, triangles: SVM. Large markers: $if_r = 0.5$, small markers: $if_r = 0.2$.

- For a fixed classifier and if_r setting, denoted by the index $i \in \{1, ..., 8\}$, and a fixed classification task, denoted by its index $j \in \{1, ..., 6\}$, let $\mathbf{u}(i, j, \text{LL})$ be the vector of the initial dominated hypervolumes estimated on the holdout set for the experiments with the LL feature set, so that $u_k(i, j, \text{LL}) = \mathcal{S}^H_{init}(i, j, k, \text{LL})$ corresponds to the hypervolume value from the k-th statistical repetition, $k \in \{1, ..., 10\}$. Similarly, let $\mathbf{v}(i, j, \text{LL})$ be the vector of the final dominated hypervolumes estimated on the holdout set, so that $v_k(i, j, \text{LL}) = \mathcal{S}^H_{fin}(i, j, k, \text{LL})$.
- H0: $\mathbf{u}$ and $\mathbf{v}$ belong to the same probability distribution.
- H1: The distributions are not equal.

The p-value of the tests applied for each combination of a classification method and a categorisation problem is equal to 0.002, and H0 is always rejected.

It also makes sense to evaluate the INCREASE OF THE SINGLE-OBJECTIVE PERFORMANCE w.r.t. m^s_{BRE}, because the classification quality is usually more relevant than the number of features. For this goal, we estimated m^s_{BRE} using complete feature sets for each combination of a classification task and a classification method as a baseline method without FS. Then, the boundary solutions with the smallest m^s_{BRE} after the optimisation were saved for comparison. Figure 5.10 shows the mean m^s_{BRE} decrease over 10 statistical repetitions for the ND solution with the smallest m^s_{BRE} (and the largest m_{SFR}), denoted by $\overline{m}^s_{BRE}$, related to m^s_{BRE} produced by the complete feature set, denoted by $m^s_{BRE}(\Phi_{all})$.

For C4.5, the m^s_{BRE} decrease is between 22.66% and 51.94%. For RF, it is between 20.95% and 54.28%, for NB between 21.38% and 77.39%, and for SVM between 10.08% and 47.95%. This means that the optimised models are not only better with respect to the dominated hypervolume, but they achieve smaller error rates. In general, it cannot be expected that the full feature sets always perform worse with regard to a quality performance measure. But it is indeed often the case, because too many irrelevant features overwhelm classification methods, as discussed in Section 3.1. The benefit varies, depending on the classifier and the task: for example, all error decrease rates are below 40% for the ClubDance category, and above 40% for Classic. NB and RF profit stronger for Classic, Rap, HeavyMetal, and ProgRock, however achieve only smaller improvements for Pop and ClubDance.

The m^s_{BRE} decrease is also confirmed as being significant by the application of the

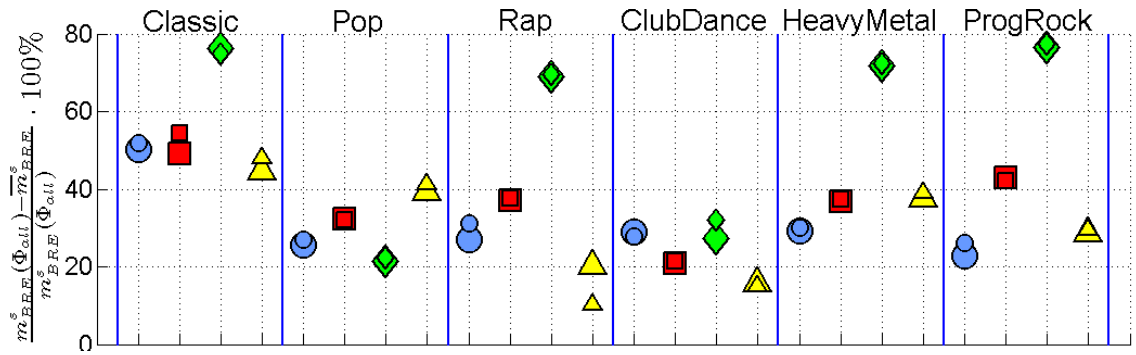

Figure 5.10.: Decrease of m^s_{BRE} for the best-m^s_{BRE} solution after the optimisation, compared to the error using the complete feature set. Circles: C4.5, squares: RF, diamonds: NB, triangles: SVM. Large markers: $if_r = 0.5$, small markers: $if_r = 0.2$.

Wilcoxon signed rank test. For each combination of a classifier and a categorisation task, H0 is rejected, using the following test setup:

- For a fixed classifier and if_r setting, denoted by the index $i \in \{1, ..., 8\}$, and a fixed classification task, denoted by its index $j \in \{1, ..., 6\}$, let $\mathbf{u}(i, j, \mathrm{LL}, \Phi_{best})$ be the vector of the smallest m^s_{BRE} estimated on the holdout set for the experiments with the LL feature set, so that $u_k(i, j, \mathrm{LL}, \Phi_{best}) = m^s_{BRE}(i, j, k, \mathrm{LL}, \Phi_{best})$ corresponds to the m^s_{BRE}-best value from the k-th statistical repetition, $k \in \{1, ..., 10\}$. Similarly, let $\mathbf{v}(i, j, \mathrm{LL}, \Phi_{all})$ be the vector of m^s_{BRE} estimated on the holdout set, if all features are switched on, so that $v_k(i, j, \mathrm{LL}, \Phi_{all}) = m^s_{BRE}(i, j, k, \mathrm{LL}, \Phi_{all})$ (in this case, $v_1 = v_2 = ... = v_k$).

- H0: $\mathbf{u}$ and $\mathbf{v}$ belong to the same probability distribution.

- H1: The distributions are not equal.

The p-value of the tests is in all cases 0.002, except for SVM with $if_r = 0.2$ for the Rap category, where p = 0.049 is only slightly below the 0.05 boundary.

Figures 5.9 and 5.10 can be compared. Starting with a smaller number of features ($if_r = 0.2$) obviously leads to a lower relative increase of hypervolume, but we cannot observe any significant impact of the choice of if_r on the solutions with the smallest m^s_{BRE}. A similar tendency was also observed in [223]: the initial population of feature sets with larger errors did not lead to a significantly different performance than an initialisation with feature sets, which produced smaller errors. Because the classification categories are very different, and the feature selection problem is also very complex, starting with 'better' feature subsets may lead to two very different outcomes: the probability may increase to get stuck in the local minima, or it could be indeed possible to benefit from the initial advantage of smaller feature sets and overcome the local optima, if the mutation strength is high enough.

5.2.2. High-level feature selection

Since the detailed comparison of the LL and HL feature sets is discussed in Section 5.2.3, we here only describe the study results for the classification based on high-level descriptors, as it was done for the LL feature set in the previous section.

Figure 5.11 plots the final ND solutions after EMO-FS, when only high-level features were used for classification. Two SVM ND solutions, one for Classic and one for HeavyMetal, are not plotted, because the corresponding models had both $m^s_{BRE} = 0.5$ in those cases, classifying all instances to one category.

The main tendencies are similar to the outcomes from the other studies:

- The large non-dominated fronts provide different trade-off solutions. However, the characteristics of these fronts are not the same: for example, for the category Classic relatively large feature sets using up to 13.07% of the features lead to the classification with the smallest m^s_{BRE}. For ProgRock, the situation is similar (maximal $m_{SFR} = 0.106$). On the other side, for HeavyMetal it does not make sense to increase the number of features above approximately 4% of the complete feature amount, and the number of solutions in the overall ND front across all classification methods is rather low. Similar trend can be observed for Rap.
- For all categories except one (ProgRock), at least three different classifiers contribute to the overall ND front.
- The complexities of the categories are very different: Classic is the easiest category, where at least one solution with $m^s_{BRE} < 0.02$ is provided by each classifier. The most complex categories are Pop (smallest $m^s_{BRE} = 0.1186$) and ClubDance (smallest $m^s_{BRE} = 0.1252$). A possible explanation is that Pop is a rather general genre: e.g., negative Pop examples songs, which belong to the categories Rap and R'n'B, can be in principle described also as popular, and may have several similar distributions of high-level characteristics as Pop songs. ClubDance is on the other side a very specific subgenre, which is more complex to distinguish from other music with strong beat impulses, e.g., dance pop or alternative rock.

As in the previous section, we first measure the INCREASE OF THE MULTI-OBJECTIVE PERFORMANCE after the optimisation. Figure 5.12 plots the increase of the mean dominated hypervolume on the holdout set. For all combinations of a classifier and a categorisation task, EMO-FS proves its general ability to create fronts with solutions which perform better w.r.t. both metrics on the independent holdout set. The increase of hypervolume is higher for $if_r = 0.5$, because the initial populations start with a significantly larger number of features. The large increase of hypervolume for SVM with $if_r = 0.5$ comes from the poor performance of the linear kernel with default parameters on larger feature sets: here, all instances are assigned to the same category. It also means that the implemented multi-objective feature selection helps to strongly reduce this disadvantage of the linear kernel.

The increase of hypervolume is again confirmed as being significant by the Wilcoxon signed rank test for the following test setup (we repeat it from Section 5.2.1), and the p-values are equal to 0.002 in all cases:

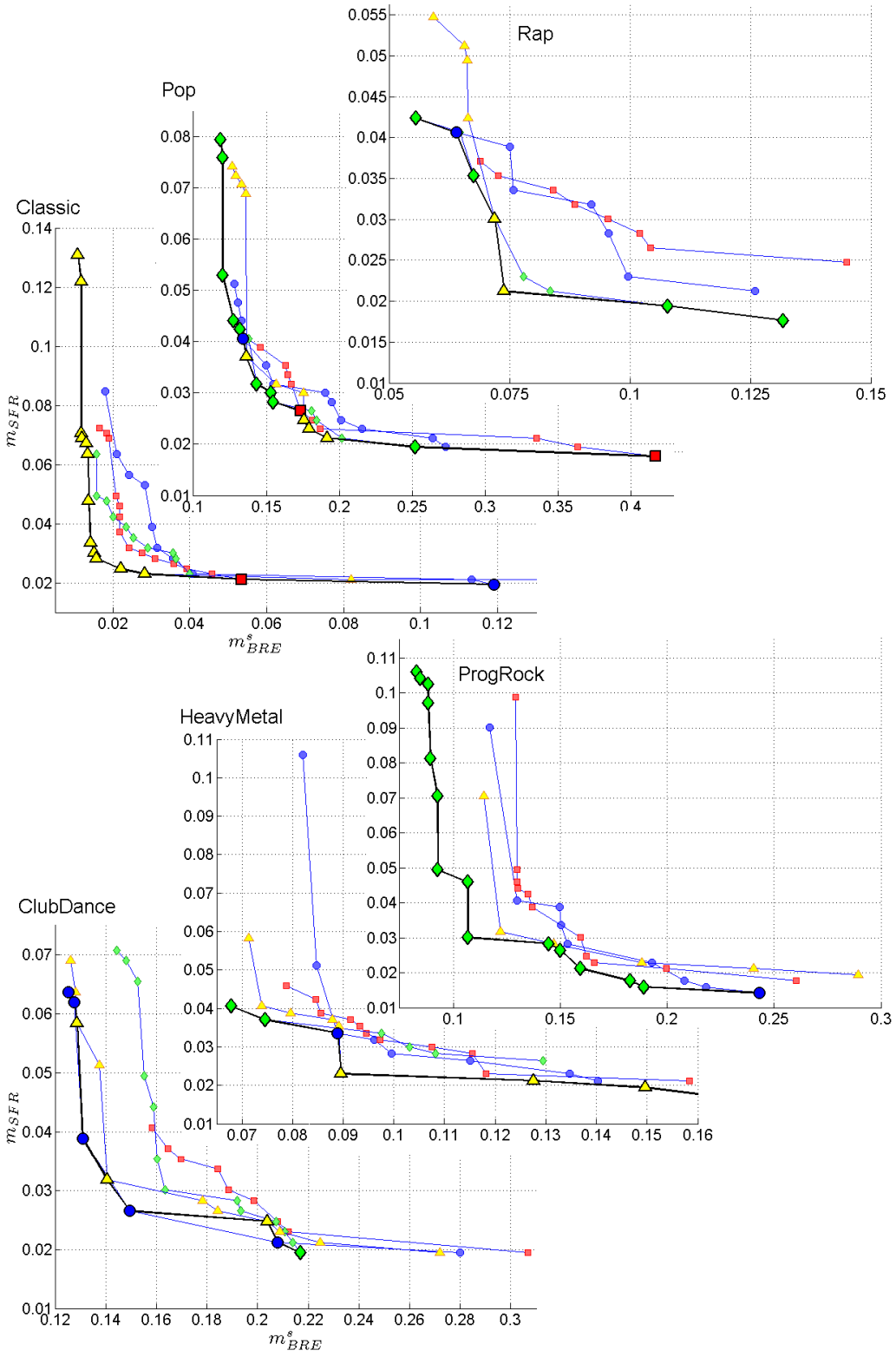

Figure 5.11.: The best ND fronts after genre and style recognition with the HL set. Circles: C4.5, squares: RF, diamonds: NB, triangles: SVM. The ND fronts for each classifier are indicated with thin lines. The ND fronts across all classifiers are indicated with thick lines, and the markers of the corresponding models are enlarged.

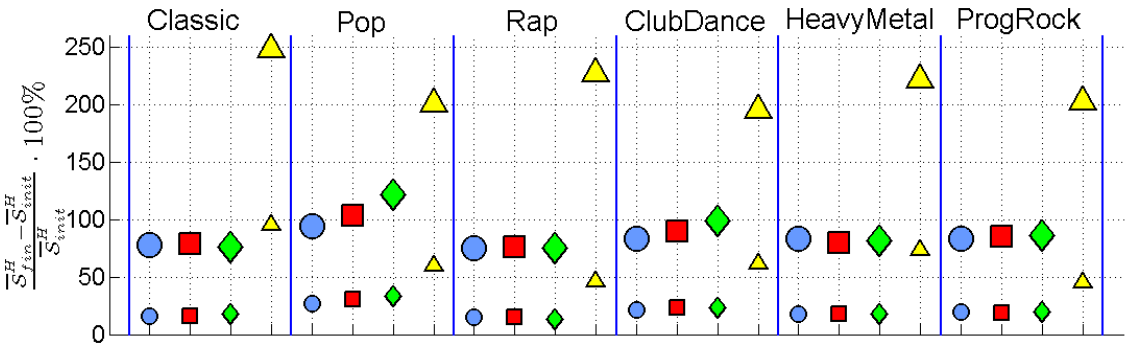

Figure 5.12.: Increase of the relative mean holdout dominated hypervolume after the optimisation. Circles: C4.5, squares: RF, diamonds: NB, triangles: SVM. Large markers: $if_r = 0.5$, small markers: $if_r = 0.2$.

- For a fixed classifier and if_r setting, denoted by the index $i \in \{1, ..., 8\}$, and a fixed classification task, denoted by its index $j \in \{1, ..., 6\}$, let $\mathbf{u}(i, j, \mathrm{HL})$ be the vector of the initial dominated hypervolumes estimated on the holdout set for the experiments with the HL feature set, so that $u_k(i, j, \mathrm{HL}) = \mathcal{S}^H_{init}(i, j, k, \mathrm{HL})$ corresponds to the hypervolume value from the k-th statistical repetition, $k \in \{1, ..., 10\}$. Similarly, let $\mathbf{v}(i, j, \mathrm{HL})$ be the vector of the final dominated hypervolumes estimated on the holdout set, so that $v_k(i, j, \mathrm{HL}) = \mathcal{S}^H_{fin}(i, j, k, \mathrm{HL})$.
- H0: $\mathbf{u}$ and $\mathbf{v}$ belong to the same probability distribution.
- H1: The distributions are not equal.

The goal of the next investigation is to measure the INCREASE OF THE SINGLE-OBJECTIVE PERFORMANCE. Figure 5.13 plots the mean m^s_{BRE} decrease of the best-m^s_{BRE} solutions for each statistical repetition, compared to the full feature sets. For almost all combinations of a category and a classifier, it is possible to achieve more than 20% reduction of the error. The first exception is the Classic category, which is characterised by smaller error decreases for C4.5 and RF. For Rap and C4.5, it even seems to be slightly preferable to use the complete feature set for classification. The design and integration of further high-level features, which have highly distinctive characteristics for Rap, might help to overcome this problem. And it should not be forgotten that classification with the complete feature set is significantly slower, requires higher storage demands for features and models, and the models have a higher tendency to be overfitted.

For the estimation of the significance of the error decrease, we repeat the application of the Wilcoxon signed rank test with the following setup:

- For a fixed classifier and if_r setting, denoted by the index $i \in \{1, ..., 8\}$, and a fixed classification task, denoted by its index $j \in \{1, ..., 6\}$, let $\mathbf{u}(i, j, \mathrm{HL}, \Phi_{best})$ be the vector of the smallest m^s_{BRE} estimated on the holdout set for the experiments with the HL feature set, so that $u_k(i, j, \mathrm{HL}, \Phi_{best}) = m^s_{BRE}(i, j, k, \mathrm{HL}, \Phi_{best})$ corresponds to the m^s_{BRE}-best value from the k-th statistical repetition, $k \in \{1, ..., 10\}$. Similarly, let $\mathbf{v}(i, j, \mathrm{HL}, \Phi_{all})$ be the vector of m^s_{BRE} estimated on the holdout set, if all features are switched on, so that $v_k(i, j, \mathrm{HL}, \Phi_{all}) = m^s_{BRE}(i, j, k, \mathrm{HL}, \Phi_{all})$ (in this case, $v_1 = v_2 = ... = v_k$).

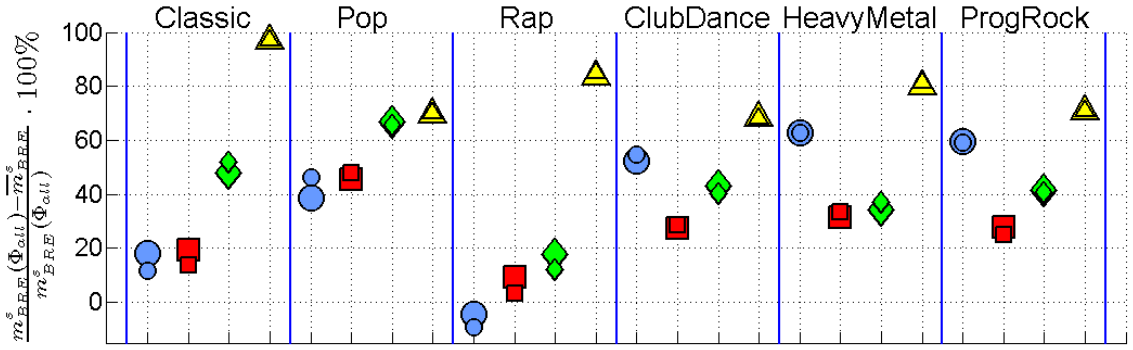

Figure 5.13.: Decrease of m^s_{BRE} for the best-m^s_{BRE} solution after the optimisation, compared to the error using the complete feature set. Circles: C4.5, squares: RF, diamonds: NB, triangles: SVM. Large markers: $if_r = 0.5$, small markers: $if_r = 0.2$.

- H0: **u** and **v** belong to the same probability distribution.
- H1: The distributions are not equal.

Since the error decrease rates were low or negative for several cases, it should not be expected that H0 will be rejected for all combinations of a classifier and a task. This is illustrated by Fig. 5.14, which plots the corresponding p-values. H0 is not rejected for 4 Rap experiments, and 3 Classic experiments. However, the overall H0 rejection rate is 83%.

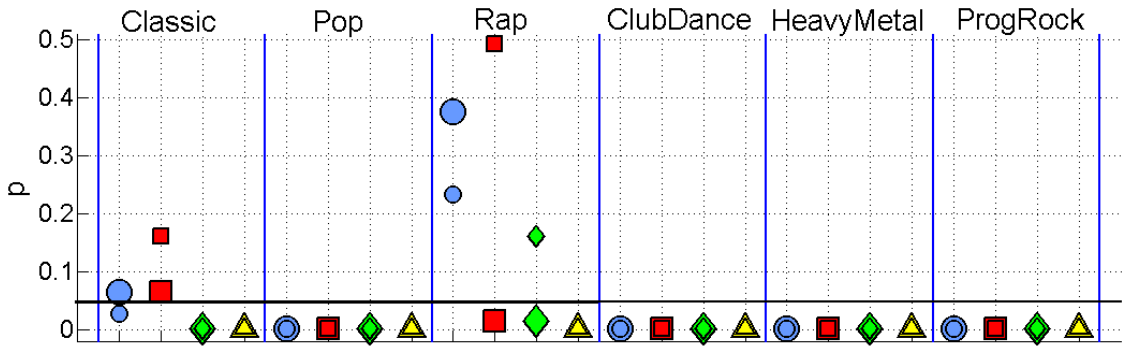

Figure 5.14.: p-values after the Wilcoxon signed rank test, comparing the best m^s_{BRE} solutions to solutions with the complete feature set (the exact test description is provided in text). H0 is rejected, if $p < 0.05$. $p = 0.05$ is marked with the thick horizontal line. Circles: C4.5, squares: RF, diamonds: NB, triangles: SVM. Large markers: $if_r = 0.5$, small markers: $if_r = 0.2$.

5.2.3. Comparison of low-level and high-level feature sets

In the previous sections, we discussed the performance of EMO-FS for the LL and HL sets separately. Another question is, if the designed high-level features can replace the baseline LL set. Then, only the interpretable musical characteristics would be integrated into the

classification models. As we have already discussed in Section 5.2, in general we cannot expect that the models based on high-level features would have a higher classification performance.

However, Fig. 5.15 illustrates that the HL set is in many cases comparable to the LL set according to the COMPARISON OF THE MULTI-OBJECTIVE PERFORMANCE. The mean holdout dominated hypervolumes $\overline{S}^H_{fin}$ from the experiments with the LL set are indicated with markers with the white background. The $\overline{S}^H_{fin}$ values for the same combination of a classifier and a task from the HL experiments are positioned slightly shifted to the right, and are indicated with the markers with different background colours: blue circles for C4.5, red squares for RF, green diamonds for NB, and yellow triangles for SVM (we explain the meaning of asterisks below). Higher $\overline{S}^H_{fin}$ corresponds to higher performance w.r.t. m^s_{BRE} and m_{SFR}.

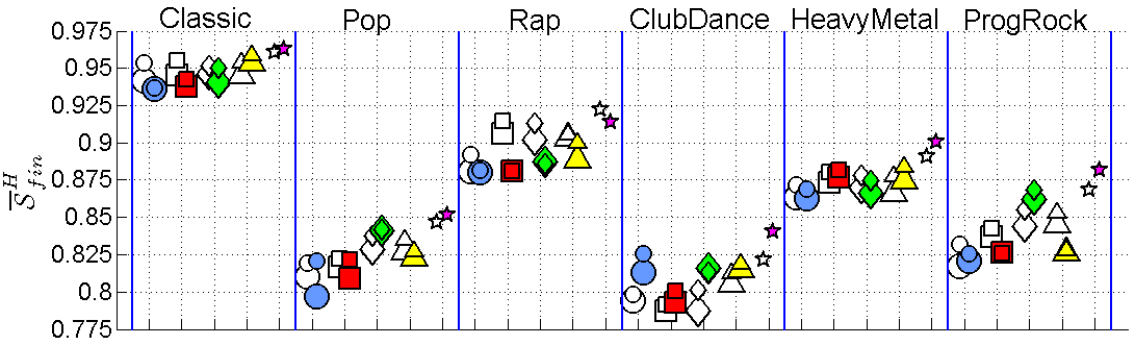

Figure 5.15.: Mean holdout dominated hypervolumes using the LL and HL feature sets. Markers with the white background: the LL set, markers with the coloured backgrounds: the HL set. Circles: C4.5, squares: RF, diamonds: NB, triangles: SVM, asterisks: combined experiments with all classifiers. Large markers: $if_r = 0.5$, small markers: $if_r = 0.2$.

The HL performance depends on the combination of a classifier and a task:

- For ClubDance, all classifiers benefit from the switch to HL features.
- For HeavyMetal, RF and SVM provide higher hypervolumes with HL features.
- For ProgRock, only NB has a stronger increase of hypervolumes for the HL set.
- For Rap, all classifiers perform better with LL features.

As we discussed for all EMO-FS studies, the non-dominated fronts built from all classifier solutions always contained solutions from several different classification methods. The integration of only one classifier would reduce the dominated hypervolume of these fronts. This was confirmed by statistical tests for instrument recognition in [215]. We may combine the solutions of the four classification methods and the two if_r settings to a 'multi-classifier' experiment. Then, we still have 10 statistical repetitions, and can estimate $\overline{S}^H_{fin}$ across them. These values are marked with asterisks in Fig. 5.15. In that case, it can be observed that the mean hypervolume performance of the classification based on the HL set is higher than the performance of the classification based on the LL set for all categories but Rap.

Another relevant observation for these 'multi-classifier' runs is the varying performance of high-level features for genres and styles. For the three genres, the HL set only slightly outperforms the LL set on average for Classic and Pop, and has a poorer performance for Rap. For the more specific style categories, the HL set always has a higher $\overline{S}^H_{fin}$ than the LL set: the switch from LL to HL leads to a relative $\overline{S}^H_{fin}$ increase to 102.33% for ClubDance, to 101.15% for HeavyMetal, and to 101.55% for ProgRock. Even if these improvements are relatively small, it means that the hardly interpretable low-level features can be *completely* replaced by high-level ones, even with a slight increase of performance. This supports well our theoretical suggestion that high-level features may be especially valuable for specific genres and personal preferences (see the discussion in Section 2.2.1, and note 10).

In the next step, we examine by means of statistical tests, if the performance difference for the classification based on the LL and HL sets is significant. Because the LL and HL experiments are independent from each other, the corresponding $\overline{S}^H_{fin}$ are not paired, and we apply the Mann-Whitney U-test for the following hypotheses:

- For a fixed classifier and if_r setting, denoted by the index $i \in \{1, ..., 8\}$, and a fixed classification task, denoted by its index $j \in \{1, ..., 6\}$, let $\mathbf{u}(i, j, \text{LL})$ be the vector of the final dominated hypervolumes estimated on the holdout set for the experiments with the LL feature set, so that $u_k(i, j, \text{LL}) = \mathcal{S}^H_{fin}(i, j, k, \text{LL})$ corresponds to the hypervolume value from the k-th statistical repetition, $k \in \{1, ..., 10\}$. Similarly, let $\mathbf{v}(i, j, \text{HL})$ be the vector of the final dominated hypervolumes estimated on the holdout set for the experiments with the HL feature set, so that $v_k(i, j, \text{HL}) = \mathcal{S}^H_{fin}(i, j, k, \text{HL})$.
- In case of the multi-classifier run comparison, $u_k(j, \text{LL}) = \mathcal{S}^H_{fin}(j, k, \text{LL})$ corresponds to the dominated hypervolume from the non-dominated fronts, which are created by the combination of solutions from all classifiers and all if_r settings from the k-th runs of these combinations, and the experiments with the LL set. $v_k(j, \text{HL}) = \mathcal{S}^H_{fin}(j, k, \text{HL})$ is estimated similarly for the experiments with the HL set.
- H0: $\mathbf{u}$ and $\mathbf{v}$ belong to the same probability distribution.
- H1: The distributions are not equal.

We distinguish between four cases:

- If H0 is rejected and $\overline{\mathbf{u}} < \overline{\mathbf{v}}$, it means that the HL set performance is significantly higher than the LL set performance.
- If H0 is rejected and $\overline{\mathbf{u}} > \overline{\mathbf{v}}$, the HL set performance is significantly lower than the LL set performance.
- If H0 is not rejected and $\overline{\mathbf{u}} < \overline{\mathbf{v}}$, the HL and the LL set performances are not significantly different, but the HL set slightly outperforms the LL set on average.
- If H0 is not rejected and $\overline{\mathbf{u}} > \overline{\mathbf{v}}$, the HL and the LL set performances are not significantly different, but the LL set slightly outperforms the HL set on average.

Table 5.5 provides the statistics about the accepted hypothesis and the comparison of $\overline{\mathbf{u}}$ and $\overline{\mathbf{v}}$. Several observations can be made:

Table 5.5.: Statistical comparison of $\mathcal{S}^H_{fin}$(LL) and $\mathcal{S}^H_{fin}$(HL). For the test outcomes in the column heads (accepted hypothesis and mean comparison), the number of the corresponding experiments is provided. The hypotheses setup and the detailed test explanation are provided in text.

Category	H1 and $\overline{\mathbf{u}} < \overline{\mathbf{v}}$	H0 and $\overline{\mathbf{u}} < \overline{\mathbf{v}}$	H0 and $\overline{\mathbf{u}} > \overline{\mathbf{v}}$	H1 and $\overline{\mathbf{u}} > \overline{\mathbf{v}}$
	$\Downarrow$ **HL preferable**	$\Downarrow$ **Both sets comparable**	$\Downarrow$	$\Downarrow$ **LL preferable**
CLASSIFIERS AND if_r SETTINGS ARE TREATED SEPARATELY				
Classic	2	0	4	2
Pop	1	2	4	1
Rap	0	0	5	3
ClubDance	6	2	0	0
HeavyMetal	0	4	4	0
ProgRock	2	1	2	3
$\sum$	11	9	19	9
EACH RUN IS BUILT FROM ALL CLASSIFIERS AND if_r SETTINGS				
Classic	1	0	0	0
Pop	1	0	0	0
Rap	0	0	0	1
ClubDance	1	0	0	0
HeavyMetal	1	0	0	0
ProgRock	1	0	0	0
$\sum$	5	0	0	1

- If we apply the tests separately for classifiers and if_r settings, the HL feature set performs better than or comparable to the LL set in 11 + 9 + 19 = 39 of 48 combinations (81.25%). In 22.92% of all combinations, using the HL set leads to even significantly higher hypervolumes.

- If we combine the solutions from all classifiers and the two different if_r settings to a single experiment, the HL feature set performs better than the LL set, for five of six categories (83.33%). Only the category Rap seems to be problematic for HL features. However, this does not mean that it is not possible to integrate further HL characteristics which are better suited for the recognition of Rap and related subgenres.

- We can compare the performances for genres and styles separately. For the tested genres, classification based on the HL set is comparable to classification based on the LL set in 62.5% of the combinations. Classification based on the HL set significantly outperforms classification based on the LL set in 12.5% if the combinations, and is outperformed by classification based on the LL set in 25% of the combinations. For the three styles, classification based on the HL set is preferable to classification based on the LL set in 33.33%, comparable to classification based on the LL set in 54.17%, and is outperformed by classification based on the LL set only in 12.5% of the combinations.

- For the multi-classifier runs, the HL set leads to significantly higher performance than the LL set for all three styles and the two of three genres. It is outperformed by the LL set only for the Rap genre.
- Another interesting statistic can be estimated from both Figures 5.8 and 5.11. For style categories, we can measure the number of classifier participations in the overall ND fronts across all classifiers, if we switch from the LL (Fig. 5.8) to the HL (Fig. 5.11) feature set. The number of corresponding C4.5 models increases from 0 to 7. For RF, this number decreases from 12 to 0. For NB, it slightly increases from 16 to 17. For SVM, it decreases from 14 to 7. These numbers cannot precisely describe the classifier performance, since the number of solutions does not measure the distribution of solutions in a front. However, the tendency is observed that the classifiers with simpler and more interpretable models (C4.5 and NB), contribute more often to the non-dominated fronts than the more complex classifiers. The latter either average the performance of many underlying trees, in our experiments 100 (RF), or estimate linear combinations of original features (SVM with a linear kernel). This suggestion comes hand in hand with the goal of interpretability. The high-level features seem to aggregate enough knowledge, so that they allow the style separability with simpler classification algorithms. More exhaustive investigations with other genres and styles are required in future. The HL set can be also extended, especially with instrumentation features, because many subgenres are characterised by the played instruments.

The COMPARISON OF THE SINGLE-OBJECTIVE PERFORMANCE of the LL and HL sets w.r.t. the best-m^s_{BRE} solutions across 10 statistical repetitions is shown in Fig. 5.16. The results are closely related to the results in Fig. 5.15. Here, higher classification performance corresponds to lower values. In almost all cases, if the performance of a classifier and a task combination w.r.t. hypervolume increases, the smallest $\overline{m}^s_{BRE}$ decreases. Sometimes, this does not hold, e.g., for SVM with $if_r = 0.5$ and HeavyMetal. Also, for Classic and the multi-classifier run comparison (marked with asterisks), $\overline{m}^s_{BRE}$ has a marginal increase, in spite of the higher hypervolume.

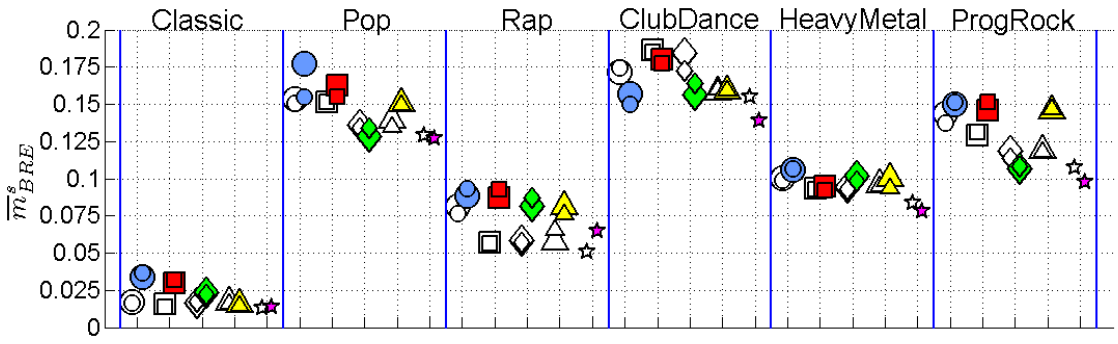

Figure 5.16.: Mean best m^s_{BRE} using the LL and HL feature sets. Markers with the white background: the LL set, markers with the coloured backgrounds: the HL set. Circles: C4.5, squares: RF, diamonds: NB, triangles: SVM, asterisks: combined experiments with all classifiers. Large markers: $if_r = 0.5$, small markers: $if_r = 0.2$.

Again, we apply the Mann-Whitney U-test to test if the observed differences between the

LL and HL sets are significant. The following hypotheses setup is applied:

- For a fixed classifier and if_r setting, denoted by the index $i \in \{1, ..., 8\}$, and a fixed classification task, denoted by its index $j \in \{1, ..., 6\}$, let $\mathbf{u}(i, j, \text{LL}, \Phi_{best})$ be the vector of the best m^s_{BRE}, estimated on the holdout set for the experiments with the LL feature set, so that $u_k(i, j, \text{LL}, \Phi_{best}) = m^s_{BRE}(i, j, k, \text{LL}, \Phi_{best})$ corresponds to the smallest m^s_{BRE} value from the k-th statistical repetition, $k \in \{1, ..., 10\}$. Similarly, let $\mathbf{v}(i, j, \text{HL}, \Phi_{best})$ be the vector of the smallest m^s_{BRE} values, estimated on the holdout set for the experiments with the HL feature set, so that $v_k(i, j, \text{HL}, \Phi_{best}) = m^s_{BRE}(i, j, k, \text{HL}, \Phi_{best})$.
- In case of multi-classifier run comparison, $u_k(j, \text{LL}, \Phi_{best}) = m^s_{BRE}(j, k, \text{LL}, \Phi_{best})$ corresponds to the smallest m^s_{BRE} from the non-dominated fronts, which are created by the combination of solutions from all classifiers and all if_r settings from the k-th runs of these combinations, and the experiments with the LL set. $v_k(j, \text{HL}, \Phi_{best}) = m^s_{BRE}(j, k)$ is estimated similarly for the experiments with the HL set.
- H0: $\mathbf{u}$ and $\mathbf{v}$ belong to the same probability distribution.
- H1: The distributions are not equal.

Table 5.6.: Statistical comparison of the smallest m^s_{BRE}(LL) and m^s_{BRE}(HL). For the test outcomes in the column heads (accepted hypothesis and mean comparison), the number of the corresponding experiments is provided. The hypotheses setup and the detailed test explanation are provided in text.

Category	H1 and $\overline{\mathbf{u}} > \overline{\mathbf{v}}$	H0 and $\overline{\mathbf{u}} > \overline{\mathbf{v}}$	H0 and $\overline{\mathbf{u}} < \overline{\mathbf{v}}$	H1 and $\overline{\mathbf{u}} < \overline{\mathbf{v}}$
	⇓ **HL preferable**	⇓ **Both sets comparable**	⇓	⇓ **LL preferable**
CLASSIFIERS AND if_r SETTINGS ARE TREATED SEPARATELY				
Classic	0	2	0	6
Pop	0	2	3	3
Rap	0	0	1	7
ClubDance	4	3	1	0
HeavyMetal	0	2	6	0
ProgRock	0	2	2	4
$\sum$	4	11	13	20
EACH RUN IS BUILT FROM ALL CLASSIFIERS AND if_r SETTINGS				
Classic	0	0	1	0
Pop	0	1	0	0
Rap	0	0	0	1
ClubDance	1	0	0	0
HeavyMetal	1	0	0	0
ProgRock	1	0	0	0
$\sum$	3	1	1	1

The results are in most cases similar to the hypervolume comparison and are listed in Table 5.6. The relevant observations are briefly outlined below.

- For the tests which are applied for each combination of a classifier and a task separately the classification based on the HL feature set produces comparably smallest m^s_{BRE} values as the classification based on the LL set in 50% of the combinations. The classification based on the HL set outperforms the classification based on the LL set in 8.33% and is outperformed by the classification based on the LL set in 41.67% of the combinations.
- The combination of all classification methods to a single experiment leads in 50% of the cases to a better performance on the HL set. For 33.3% of the categories, the performance on both sets is similar, and only for Rap the LL set leads to non-dominated solutions with significantly lower m^s_{BRE}.
- As for hypervolumes, the HL set leads to significantly smaller m^s_{BRE} of the non-dominated solutions with the lowest m^s_{BRE} of each run than the LL set for all style categories and multi-classifier experiments. If the performance is measured independently for each combination of a classifier and a task, the LL set leads to a significantly higher performance only in 16.67% of all style combinations: it holds for ProgRock, when classified by RF and SVM.
- The last observations support the statement that C4.5 and NB have a tendency to perform better with high-level features, in particular, for style recognition. We refer again to Figures 5.8 and 5.11. For the HL set, a NB solution has the smallest m^s_{BRE} for four of six categories (two styles and two genres), a C4.5 solution has the smallest m^s_{BRE} for ClubDance, and SVM has the smallest m^s_{BRE} for Classic. For the LL set, the situation is almost opposite. Here, the more complex classifiers RF and SVM produce more solutions with the smallest m^s_{BRE}: RF for Classic, and SVM for the three categories. NB has the smallest m^s_{BRE} for Rap and ProgRock, and C4.5 for none.

Summarising our exhaustive comparison of the classification based on the LL and HL feature sets w.r.t. both two-objective performance (dominated hypervolume), and single-objective performance (smallest balanced relative error), we can state the following:

- The replacement of low-level features with high-level ones, which were directly implemented, estimated after the application of the sliding feature selection, or calculated as structural complexity characteristics, leads to similar or even better classification performances in most cases. Only for the category Rap, the HL set leads to a significant decrease of performance.
- In Sections 5.2.1 and 5.2.2, it was observed that the classification errors were reduced by EMO-FS in a predominant number of experiments, compared to the complete feature set (only for Rap with classifier C4.5 and the HL feature set, the complete feature set provided a slightly smaller m^s_{BRE}). Therefore, we may state: it is possible, with an appropriate design of high-level audio features, to successfully approach the following three different objectives at the same time: the reduction of the cardinality of the feature set, the maximisation of the classification quality, and the enhancement of the interpretability of features and models.
- It is underlined by statistical tests that the classification based on the HL set performs especially well for the three style categories. This matches real-world situations, where it is more promising to recognise specific user-centered preferences and

not the rather general genres. Two further drawbacks of genre classification are that genres may evolve over time and, furthermore, in [165] it was argued that no common genre taxonomy exists.

- A tendency is observed that the more interpretable classification models created by C4.5 and NB outperform the less interpretable RF and SVM models, if the classification is done with high-level features.
- It is obvious that the results of this study have some limitations. In future, it would be promising to compare low-level and high-level feature sets for a significantly larger number of different genres, subgenres, and personal music preferences. Also, many high-level features are not always robust. For example, we cannot expect that an instrument identification model would identify the instruments correctly, if a polyphonic recording contains many instruments, which have not been used for the training of this model. The joint cooperation efforts of all involved interdisciplinary MIR research domains may help to deal with this challenge in future. Then, automatic music classification will provide a robust and fast possibility to learn personal music preferences not only with high classification quality, but also with interpretable outputs of all related methods.

5.2.4. Analysis of high-level features

High-level features allow a better interpretability of the classification models and help to understand, which characteristics are especially important for the identification of a certain genre or style. In this section, we compare different high-level feature groups. From 566 characteristics of the HL set, the following five partly overlapping subgroups are distinguished.

The 1st group of 346 HARMONY AND MELODY CHARACTERISTICS consists of the following features:

- 258 features are extracted from frames which are shorter than the classification frames ($W_e < W_c = 4$ s). They are listed in Table A.2 and are indicated with an 'H' in the last column. For the original 129 feature dimensions, the estimation of the mean and the standard deviation for each classification window leads to 258 features.
- 5 features which describe the characteristics of chords are estimated from extraction frames with $W_e > W_c = 4$ s. They are listed in the last section of Table A.2 ('Chord analysis' features).
- 48 features are derived from the GFKL 2011 descriptors. For six categories (Harmony major, Harmony minor, Melodic range $\leq$ octave, Melodic range $>$ octave, Melodic range linearly, and Melodic range volatile) and four classifiers, two models with smallest m_{BRE} were selected from the non-dominated fronts.
- 35 structural complexity characteristics describe the chord and the harmony complexities and are listed in Table A.7.

The 2nd group is related to INSTRUMENTS and consists of 118 features.

- 32 features are derived from the instrument recognition study described in Section 5.1.1. Two models with the smallest m_{BRE} of each classifier for the four classification

tasks (Guitar, Piano, Wind, and Strings) have been previously applied to extract the proportion of the positive instrument identifications in 10 s extraction frames.

- 72 features are built in the same way from the GFKL 2011 set, see Section 5.1.3. The categories are: Effects distortion, Instrumentation drums, Singing solo clear, Singing solo man, Singing solo polyphonic, Singing solo rough, Singing solo unison, Singing solo woman, and Singing voice medium.
- 14 structural complexity characteristics are listed as 'Instruments complexity' in Table A.7.

The 3rd group comprises 72 MOOD features. Here, two models with the smallest m_{BRE} from each classifier are applied for all eight mood categories (Aggressive, Confident, Earnest, Energetic, PartyCelebratory, Reflective, Sentimental, Stylish), and for the 'Activation level high' category from the GFKL 2011 set.

The 4th, rather small group, contains 30 TIME-RELATED features: 9 features which are marked as high-level in the last column of Table A.3 and 21 'Tempo and rhythm complexity' features from Table A.7.

The 5th and last group is built from 70 STRUCTURAL COMPLEXITY characteristics which are constructed from high-level features ('Chord complexity', 'Harmony complexity', 'Instruments complexity', and 'Tempo and rhythm complexity' from Table A.7).

It is possible to measure how often the features of these groups have been selected after the optimisation, compared to a random feature distribution with the same proportion of the selected features. This can be estimated as follows. Let I_k be the set with the indices of the features, which belong to the high-level feature group k, $k \in \{1, ..., 5\}$ (the indices mark the positions in the complete set of F features). Then, for a solution i with $m_{SFR}(i) \cdot F$ selected features, the number of the *expected* I_k selections is equal to $m_{SFR}(i) \cdot \frac{|I_k|}{F}$. The number of the *actually selected* I_k features is equal to $\sum_{j \in I_k} q_j$ ($\mathbf{q}$ is the bit vector, which represents the feature selections). Now, we can estimate the amount of the actually selected features to the expected features for solution i and group k:

$$\phi(i,k) = \frac{\sum\limits_{j \in I_k} q_j \cdot F}{m_{SFR}(i) \cdot |I_k|}. \tag{5.1}$$

If we analyse L different solutions, the mean HIGH-LEVEL FEATURE GROUP SHARE FACTOR can be estimated as:

$$\overline{\phi}(i,k) = \frac{1}{L} \sum_{i=1}^{L} \phi(i,k). \tag{5.2}$$

We distinguish between the two following ways to estimate $\overline{\phi}(i,k,l)$ for a fixed classification category:

- Estimate $\overline{\phi}(i,k,l)$ for all final non-dominated solutions i, all high-level groups k, and all statistical repetitions l. In that case, 4 classifiers, 2 different if_r settings, and 50 solutions from each run are taken into account: $i \in \{1, ..., 400\}$. If $\overline{\phi}(i,k,l) > 1$, it means that the features of high-level group k are selected for the final non-dominated

solutions after the optimisation more often than by chance. If $\overline{\phi}(i,k,l) < 1$, it means that the features of high-level group k are selected for the final non-dominated solutions after the optimisation less often than by chance.

- Estimate $\overline{\phi}(i,k,l)$ for solutions i with the smallest m^s_{BRE}, all high-level groups k, and all statistical repetitions l. In that case, 4 classifiers and 2 different if_r settings from each run are taken into account, and $i \in \{1,...,8\}$. If $\overline{\phi}(i,k,l) > 1$, it means that the features of high-level group k are selected for the final non-dominated solutions with the smallest m^s_{BRE} more often than by chance. If $\overline{\phi}(i,k,l) < 1$, it means that the features of high-level group k are selected for the final non-dominated solutions with the smallest m^s_{BRE} less often than by chance.

We denote the mean share factors for the five high-level feature groups by ϕ^{HARM}, ϕ^{INSTR}, ϕ^{MOOD}, ϕ^{TIME}, and $\phi^{STRCOMP}$. If the features of the group k are selected by chance, $\overline{\phi}(i,k,l) \approx 1$. We apply the Wilcoxon signed rank test with the following hypotheses[9].

- For a fixed classification task, $i \in \{1,...,400\}$ final solutions of each classifier and each if_r setting after the optimisation, and $k \in \{1,...,5\}$ high-level feature groups introduced above, let $\mathbf{u}(i,k)$ be the vector of the mean high-level feature group share factors, so that $u_l(i,k) = \overline{\phi}(i,k,l)$ corresponds to the mean high-level feature group share factor from the l-th statistical repetition, $l \in \{1,...,10\}$. Let $\mathbf{v}$ be the vector of ones, so that $v_1 = v_2 = ... = v_l = 1$. This artificially created vector corresponds to the situation, where the expected number of features from any group is equal to the actually selected number of features.
- For the analysis of solutions with the smallest m^s_{BRE}, the only difference is that $i \in \{1,...,8\}$ m^s_{BRE}-best solutions of each classifier and each if_r setting after the optimisation are used for the estimation of $u_l(i,k) = \overline{\phi}(i,k,l)$.
- H0: $\mathbf{u}$ and $\mathbf{v}$ belong to the same probability distribution.
- H1: The distributions are not equal.

Table 5.7 lists the mean high-level feature group share factors. The upper half of the table contains the values, which have been estimated from the complete final non-dominated populations, and the lower half for the m^s_{BRE}-best solutions. If H0 was rejected (the distribution of the mean factors has a significant difference to the 1-distribution), the corresponding value is marked with bold font.

If the share factor is above one and H0 is rejected, it means that a disproportionately high amount of the features from the corresponding group has been selected. If the share factor is above one and H0 is not rejected, it still means that on average more features of this group were selected than by chance.

If the share factor is below one and even if H0 is rejected, it does not mean that the features of this group are irrelevant and can be omitted. As an example, for the Classic category $\phi^{TIME} = 0.6416$, if the group share factor is averaged across all solutions. $\phi^{TIME} = 0.5921$, if this factor is averaged across the solutions with smallest m^s_{BRE}, and in that case H0 is rejected. However, 'Song duration' was most frequently selected by C4.5, RF and NB for Classic (the most frequently selected features are discussed later in this section).

[9] Because we compare the share factors and not concrete numbers of features, the second observation vector is constant and it is also possible to apply one sample tests, see [84].

Table 5.7.: High-level feature group share factors. The values are marked with bold font, if H0 is rejected. The hypotheses setup and detailed explanations are provided in the text.

Category	ϕ^{HARM}	ϕ^{INSTR}	ϕ^{MOOD}	ϕ^{TIME}	$\phi^{STRCOMP}$
MEAN PERFORMANCE					
Classic	**0.8942**	**1.2337**	**1.2745**	0.6416	**0.6573**
Pop	0.9469	**1.1162**	1.1753	**0.7350**	**0.6211**
Rap	1.0121	**0.8142**	1.2349	1.0280	**0.6840**
ClubDance	0.9930	**0.7415**	**1.3948**	1.1496	0.9111
HeavyMetal	**0.9353**	**1.1752**	1.0567	0.9212	0.9094
ProgRock	**0.9018**	1.0518	**1.4004**	0.9678	0.7638
PERFORMANCE OF FEATURE SETS WITH SMALLEST m^s_{BRE}					
Classic	**0.8662**	**1.2879**	1.3413	**0.5921**	**0.4475**
Pop	0.9464	1.2247	1.0100	0.7101	**0.5883**
Rap	0.9323	0.9695	1.3261	1.1180	0.8799
ClubDance	0.9698	**0.8049**	**1.3414**	1.2968	0.7626
HeavyMetal	**0.8751**	1.2360	1.1579	1.1331	0.9180
ProgRock	**0.8952**	1.1847	**1.2366**	0.9140	**0.6063**

In other words, a single feature or a few features of some group may be very relevant for a certain category, and the remaining features of this group may be completely irrelevant.

The group share factors of the high-level feature groups differ strongly from each other, as listed in Table 5.7:

- HARMONY AND MELODY CHARACTERISTICS contain more than half of all high-level features. It can be expected that many of these characteristics would be irrelevant for a certain task. For most categories, the share of these features is lower than the expected random share: $\phi^{HARM} < 1$ for 5 of 6 categories, if all final solutions are analysed, and $\phi^{HARM} < 1$ for all categories, if only the solutions with the smallest balanced relative errors are analysed. For the three categories Classic, HeavyMetal, and ProgRock, H0 is rejected. These categories have indeed a more varying distribution of harmonic characteristics in our database than Rap and ClubDance, so that the harmonic characteristics may be less suited for the separation of, e.g., Classic versus other songs.

- For the INSTRUMENT feature group, the situation is almost opposite. For four genres and styles, $\phi^{INSTR} > 1$. This means that the features of this group were selected more often than by chance. The disproportionately high rate of instrument feature selections is underlined by an H0 rejection in several cases.

- For MOOD features, $\phi^{MOOD} > 1$ in all cases. This group seems to be especially relevant for the categories ClubDance and ProgRock, where H0 is always rejected.

- The features of the TIME-RELATED group are in general less frequently selected. H0 is rejected only twice and the share factor has a relatively strong variation. As discussed above, the time-related features sometimes contain individually relevant

features, such as 'Song duration' for the Classic category.

- STRUCTURAL COMPLEXITY features are selected in general less often than by chance. For half of all corresponding values, H0 is rejected. This group seems to contain the largest share of irrelevant or redundant characteristics.

A possibility to measure the relevance of concrete features was proposed in [219]: the EXPERIMENTAL FEATURE RELEVANCE $\xi(i), i \in \{1, ..., F\}$ counts the occurrences of the feature i across all final solutions.

Figure 5.17 illustrates $\xi(i)$ estimated separately for a classifier and a task. Because of the two different if_r values, 10 statistical repetitions, and a population size of 50 individuals, $\xi(i) \in [0; 1,000]$. 15% of the lower $\xi(i)$ values are sorted out, so that only the more often selected individual features are plotted with small vertical dashes. The horizontal axis corresponds to the indices of features, and the vertical axis to the combinations of a classifier and a task. Deep red colour corresponds to higher $\xi(i)$ values, and blue colour to lower $\xi(i)$ values. The five high-level feature groups are separated by vertical dotted lines, and are marked with their abbreviations above the figure.

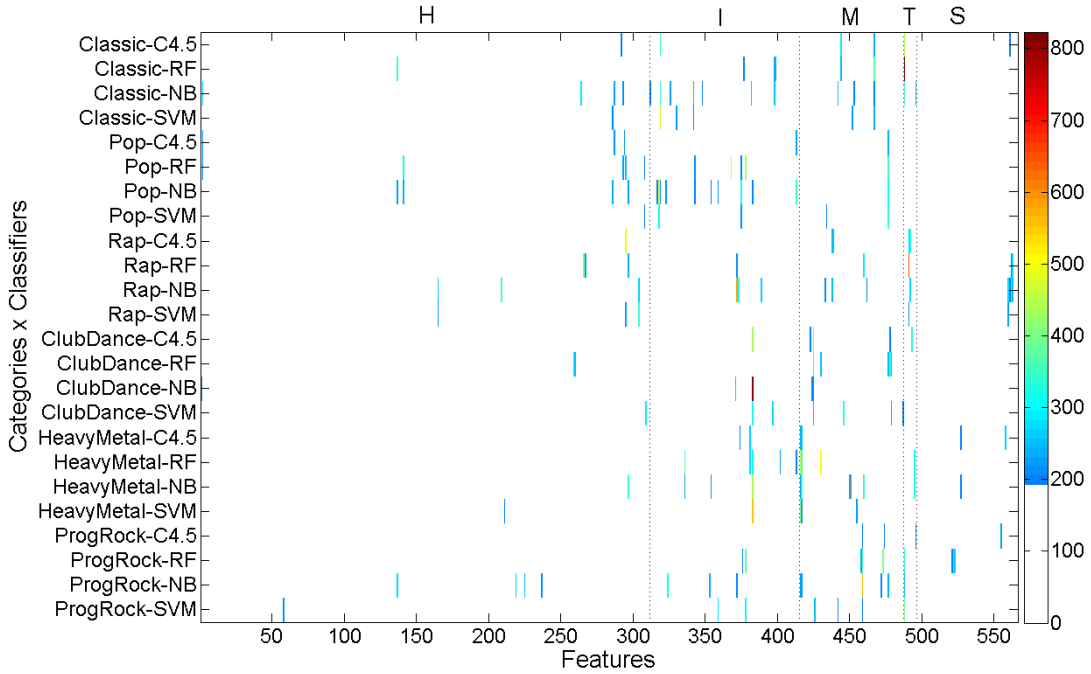

Figure 5.17.: Often-selected features for all combinations of a classifier and a category. The five high-level feature groups are separated by thin vertical dotted lines, and are marked with abbreviations above the figure: H: harmony and melody, I: instruments, M: moods, T: time-related, S: structural complexity. For a better comprehensibility, each feature is assigned to exactly one group (e.g., harmonic structural complexity features to 'S' and not 'H'). The feature which have been often selected are marked with vertical dashes. The deeper dark colour corresponds to the highest ξ values, the blue colour to lower ξ values.

We can observe a correlation between the results of Table 5.7 and Fig. 5.17. The harmony

and melody group has a small amount of the features, which have been often selected. The same holds for the structural complexity features. Instruments and moods provide a relatively high number of relevant features. Tempo is a rather controversial group: the number of selected tempo features is not high, but it sometimes contributes very relevant characteristics: 'Song duration' for Classic (with C4.5, RF and NB) and ProgRock (with RF, NB and SVM), or 'Estimated beat number' for Rap (with C4.5, RF and NB).

Another tendency was already reported in [217, 219]: some of the features are specific for a combination of a classifier and a task. As an example, the 'Song duration' feature was less often selected for Classic by SVM than by other classifiers. This means that FS must be applied separately for each classifier. If interpretability plays an important role, especially the features selected by C4.5 and NB may be more valuable, since these classifiers create simpler and more comprehensible models.

Finally, we list in Table 5.8 the three most often selected features for each category. It should be mentioned that these features sometimes may be only relevant, when they are combined with other features, as discussed in Section 3.2.3.

Several observations can be made:

- In most cases the meanings of the high-level features describe the corresponding categories well. For example, the most often selected feature for Classic and C4.5, 'Song duration', has a high interpretability: songs in the Classic category were indeed longer in our collection than popular songs. Or, for HeavyMetal and C4.5, the most often selected features are 'Singing vocals rough', 'Instrument complexity high', and 'Aggressive'. Whereas the first and the last characteristics are self-explanatory, the high rate of instrument complexity corresponds to a high instrumentation change for the large analysis windows. Indeed, some albums of our collections contained progressive and symphonic metal songs, where segments close to classical music are often alternated with segments with a high share of the distorted guitars. ProgRock songs are also often longer ('Song duration' is the most frequently selected feature for classification with SVM), have a large number of different segments ('Instr. complexity' for C4.5), and are not labelled with the PartyCelebratory mood ('PartyCeleb. C4.5' for RF).

  It is important to mention again that the individual performances of high-level features are still not perfect at the current research stage. The models trained to recognise guitar in the mixtures of guitar, piano, strings, and wind can be ineffective for other recordings. Or, the tempo estimation algorithms produce in certain cases an octave error, estimating the tempo as the double of the original. Then, slow pieces are recognised as fast [40]. Consider the following example: slow dance pop songs have to be identified in a music collection which contains mid-tempo alternative rock songs. The tempo is wrongly estimated as being twice as fast as the true tempo for the dance pop songs, and correct for all other songs. This high-level feature may indeed provide a perfect separation between misleadingly recognised 'fast' songs from the dance pop category and correctly identified 'mid-tempo' songs from the alternative rock category. But the tempo is not always correctly estimated. However, this feature indicates that the tempo *plays an important role* for this task, and this knowledge is indeed relevant for the description of the category.

- The high-level feature groups, which have a significantly lower high-level group fea-

Table 5.8.: The most frequently selected features, measured by $\xi(i)$. Feature names are abbreviated. Classifier names behind the features mean that they were estimated after the sliding feature selection.

Category	**1st**	$\xi(i)$	**2nd**	$\xi(i)$	**3rd**	$\xi(i)$
C4.5						
Classic	Song duration	444	Guitar SVM	341	Stylish NB	285
Pop	PartyCeleb. NB	240	Sing. polyph. NB	226	Max. chroma. ampl.	223
Rap	Mel. ran.≤oct. SVM	496	Tatum no. per min.	309	Beat no. per min.	262
ClubDance	Sing. rough SVM	448	Energetic C4.5	387	Tempo	322
HeavyMetal	Sing. rough NB	266	Instr. complexity	261	Aggressive C4.5	246
ProgRock	Instr. complexity	231	Diff. segment share	210	PartyCeleb. RF	202
RF						
Classic	Song duration	819	Earnest RF	375	Local tuning	357
Pop	Sing. medium C4.5	463	Sing. rough RF	427	PartyCeleb. NB	358
Rap	Beat no. per min.	652	Harmony major RF	409	Confident NB	289
ClubDance	PartyCeleb. SVM	327	Energetic C4.5	321	No. diff. chords in 10s	294
HeavyMetal	Energetic SVM	502	Aggressive C4.5	475	Strings C4.5	362
ProgRock	PartyCeleb. C4.5	400	Sing. rough RF	370	Confident RF	351
NB						
Classic	Song duration	370	Guitar SVM	328	Harmony major C4.5	288
Pop	Guitar NB	489	Sing. polyph. NB	335	Sing. medium SVM	309
Rap	Sing. medium NB	583	7. major key strength	353	Instr. complexity	317
ClubDance	Sing. rough SVM	823	Sing. medium RF	263	Energetic C4.5	220
HeavyMetal	Reflective RF	616	Sing. rough SVM	439	Aggressive C4.5	347
ProgRock	Confident RF	557	Piano C4.5	312	Local tuning	280
SVM						
Classic	Guitar SVM	467	Earnest RF	235	Wind NB	230
Pop	PartyCeleb. NB	347	Guitar NB	303	Sing. medium SVM	217
Rap	Mel. ran. volat. C4.5	338	Mel. ran.≤oct. SVM	224	Instr. complexity	223
ClubDance	Stylish SVM	329	Sing. rough SVM	316	Mel. ran. volat. NB	290
HeavyMetal	Sing. rough SVM	555	Aggressive C4.5	417	Reflective SVM	222
ProgRock	Song duration	404	Sing. rough RF	312	Effects distort. SVM	271

ture share factor than 1 (selection by chance), may contain individual features, which are very relevant for a certain category. E.g., the structural complexity of instrumentation is among the first three most selected features for Rap (with NB and SVM), HeavyMetal (with C4.5), and ProgRock (C4.5).

- Some features are among the three most selected ones for the same category, and also across several classifiers. On the other side, this is does not hold always, and we definitely recommend to apply EMO-FS for each classifier separately because of different operating principles of classification methods. SVM may profit from a transform of several original feature dimensions into the higher dimensional space, so that some of the less relevant features may become relevant in combination. This does not hold for NB, because it treats the features independently.

- The clear outcome of the study is that different categories require different high-level features. For an efficient automatic classification with less human efforts, it is reasonable to start with a large feature set. Then, feature selection, in particular,

EMO-FS, becomes essential, because the classification quality often decreases, when there are too many features. This is supported by both theoretical explanations and practical observations, as discussed before.

- The concept of sliding feature selection seems to provide a quite acceptable proportion of relevant features. These features correspond to 39.58% of the complete feature set of 566 features, and are among the three most often selected in 75% of all combinations of a classifier and a task. If we treat the 14 instrument structural complexity features as related to sliding FS, the original share of these features increases to 42.05%, and they contribute to the three most selected features in 80.56% of all cases.

6. Conclusions

6.1. Summary of results

In this work we have developed a multi-objective evolutionary feature selection framework for the optimisation of several supervised music classification tasks: recognition of instruments, moods, harmonic and melodic characteristics, genres and styles.

The first major goal is to address feature selection in a MULTI-OBJECTIVE way, optimising at least two different evaluation criteria at the same time. This approach leads to the following enhancements:

- Multi-objective feature selection helps to find a set of trade-off solutions. Then, one or several relevant feature subsets can be selected, depending on current preferences. A feature subset with a high number of features and a lower classification error provides better classification performance. A feature subset with a lower number of features and a higher classification error allows a significant decrease of storage and computing time demands. But such solutions also may be preferable because of the lower tendency to be overfitted against the training set. The comparison of the dominated hypervolumes of the initial and final feature subsets by means of statistical tests confirmed the increase of performance after the optimisation for all combinations of a classification method and a classification task.
- In another evaluation approach, we compared non-dominated set solutions with the lowest balanced relative error m^s_{BRE} to complete feature sets. The evolutionary multi-objective feature selection resulted in feature subsets with a significantly lower m^s_{BRE} than the classification with complete feature sets for almost all combinations of a classifier and a task.
- The two-objective evaluation of the optimised feature subsets (with hypervolume) and the single-objective evaluation (with the smallest balanced classification error) were both estimated on a holdout set. This means that the optimised models have proven their generalisation ability to perform well on data which neither have been used for model training nor into the optimisation of the feature selection.
- m^s_{BRE} and m_{SFR} (selected feature rate) are anticorrelated for non-dominated trade-off solutions: larger feature sets produce smaller errors and vice versa. However, this does not hold in general. As we could observe for different music classification tasks, the increase of a number of features above a certain threshold leads in many cases to higher errors (cf. Fig. 5.6). This threshold depends on the combination of a classification task and a classification method. Therefore, the best trade-off solutions can be efficiently found only by multi-objective optimisation. If the metrics would be anticorrelated for all feature subsets, it could be enough to apply a single-objective optimisation for one of the metrics.

- Because this work is the first, which to our knowledge applies an evolutionary multi-objective feature selection approach for music data analysis (except for several own preliminary contributions), we believe that in future many other promising combinations of less correlated evaluation criteria could be optimised. We discussed several metric groups and application scenarios, which are reasonable for the multi-objective optimisation in music classification. Classification quality, quality for different and less balanced data subsets, model stability, runtime and storage demands, feature extraction costs, interpretability, user efforts, etc. are often in conflict and are loosely correlated. The optimisation according to only one common criterion, e.g., accuracy, often leads to the diminished performance w.r.t. other relevant evaluation metrics.

The second major goal of this work is to enable genre and style classification based on HIGH-LEVEL AUDIO FEATURES. The advantage of these features is that the derived properties of genres and styles become comprehensible and interpretable. The general advantage of audio features is that they can be extracted from any digitally stored song, independently of its popularity or availability in digital music stores. Features from other domains are often hard to extract (the score is not always available), are subjective, incomplete, or erroneous (genre taxonomies, community tags). The main outcomes of our studies are listed as follows.

- A large set of audio high-level features is designed. A part of the features is directly extracted from existing algorithms, software tools, or implemented by ourselves. Another part is created after sliding feature selection, as discussed in Section 3.3. Here, the results of the classification models are averaged for larger extraction frames, so that, e.g., the share of guitar identifications in a 10 s frame or the share of the energetic mood identifications in a 24 s frame is estimated. The last part of the features consists of structural complexity characteristics, which are calculated as outlined in Section 2.3.3. These features describe structural changes in chord distribution, harmony, instrumentation, and temporal characteristics.

- It is possible to completely replace the baseline 636-dimensional low-level feature set by the 566-dimensional high-level feature set in most cases without any significant decrease of the classification quality. The subsequent feature selection helps to find the most relevant high-level features for each category. The three most often selected features for each tested category are listed in Table 5.8. With these features, comprehensible descriptions of the corresponding genres and styles are possible.

- If a single experiment integrates models built by all four classifiers, feature selection and classification with high-level features may even lead to a slight increase of hypervolume for all categories, except for Rap. This increase is confirmed by means of statistical tests to be significant and not achieved by chance.

- Classification based on high-level features performs especially well for the three style categories, when the models of the four classification methods are combined. The application of statistical tests confirms that for all styles the final hypervolumes are significantly higher, when the high-level feature set is used. The increase of performance also holds for the comparison of the solutions with the smallest balanced relative error. Hence, high-level features seem to be especially valuable, if more specific categories, such as personal preferences, should be learned.

- The designed approach is well suited to automatically derive the current user preferences, which may change over time depending on the situation. The same high-level features can be used as the initial feature set, where feature selection would detect the most relevant high-level descriptors for a certain category. The music expert efforts, which are necessary for the labelling of large music collections, can be strongly reduced. It is possible to provide the ground truth for a limited number of songs and to automatically categorise the remaining music pieces.
- For a better understanding of genre and style properties, not only the features, but also the classification models are relevant. Highly complex models, for example built by SVM with nonlinear kernels, transform the original feature dimensions into higher spaces and separate the classes in feature domains, which are not interpretable anymore. However, as we could see in our experiments, C4.5 and NB perform often comparably or even better than the RF and linear SVM, when they operate on high-level features.
- For the extraction of high-level features and music categories, we introduced the concept of sliding feature selection. First, a set of high-level features is learned from low-level characteristics, where the most relevant are identified with evolutionary multi-objective feature selection. These high-level features can be then used for the estimation of further high-level characteristics, and so on. Although the features, which are estimated using sliding feature selection, contribute to only approximately 42% of all high-level descriptors, they contain above 80% of the three most often selected features for genre and style recognition listed in Table 5.8.

Besides, we provide FORMAL DESCRIPTIONS OF THE STEPS OF THE MUSIC CLASSIFICATION CHAIN. This work has been already started in our previous publications:

- The algorithm chain for music classification was implemented step by step during the development of the Advanced MUSic Explorer (AMUSE) [221]. In Section 2.1.3, we define the inputs and the outputs of the corresponding tasks.
- Feature processing plays often an underrepresented role in music information retrieval studies. However, improper preprocessing and aggregation of features may lead to a decrease of the classification performance. A categorisation of the different feature processing methods is provided in Section 2.3. In particular, a part of the time dimension processing methods was developed and compared in our previous study [222].
- For different multi-objective evaluation and optimisation scenarios in music classification, we provide a categorisation of related evaluation metrics in Section 4.1. These metric groups have been already briefly described in [217]. Further, we discuss the preconditions for the evaluation of algorithms, which are still not always fulfilled in many current MIR investigations. This evaluation is based on four essential components: the choice of evaluation metrics, the choice of the evaluation method, the proper assignment of classification instances to training, optimisation and holdout data sets, and the measurement of significance with the help of statistical tests.

Finally, the DEVELOPMENT OF AMUSE allowed a large number of the related studies [205, 220, 223, 15, 157, 222, 214, 186, 217, 219, 215, 218, 216]. AMUSE makes it possible to concentrate on the development of concrete methods and to run large-scale evaluations

of the experiments. The node-based architecture allows job scheduling on grid farms – AMUSE experiments were successfully run on the four different grid systems (LiDOng[1], LSF[2], SLURM[3] and Sun Grid Engine[4]). Because of the already integrated algorithms and plugins, it is, e.g., possible to

- concentrate only on feature extraction and implement new features. They can be simply integrated into the classification experiments, and the impact of these features on the classification quality can be measured.
- Another possibility is to develop new feature aggregation methods, using the already available features (Chroma Toolbox [155], MIR Toolbox [117], jAudio [138], NNLS Chroma [135], and Yale [147] are available either as AMUSE plugins or integrated libraries).
- The integration of further classification algorithms from WEKA [76] and RapidMiner [147] is very straightforward, because these Java libraries are directly integrated into AMUSE. If it is desired to implement other classification methods, they only have to support the AMUSE input/output formats and are not limited to any programming language.
- The XML configuration of the evolutionary feature selection allows an intuitive setup of many different evolutionary algorithms for the optimisation of feature selection and classification window size. For example, it can be switched between different selection methods, self-adaptation parameters, and local search settings.
- The high-level feature set, which is designed in this work, is completely available for extraction with AMUSE. Unfortunately, the extraction of the high-level characteristics with a current AMUSE version is not very simple, and the extractors of several underlying low-level features cannot be distributed freely because of the licence restrictions. However, we plan to improve the tool support in future and also to provide more documentation.

6.2. Directions for future research

We could show in our studies that music classification based on high-level descriptors is comparable or even preferable to the classification based on low-level audio features. Also, evolutionary multi-objective feature selection has proven its ability to select many feature subsets with different trade-off characteristics. However, to increase the robustness and the efficiency of our approach, further developments and studies are still required. In this section, we discuss several ideas for future research.

In our opinion, one of the most relevant directions is to develop MORE ROBUST AUDIO HIGH-LEVEL FEATURES, and to describe their relevance to a category in a more NATURAL WAY:

[1] http://lidong.hrz.tu-dortmund.de/ldw/index.php/Main_Page, date of visit: 15.02.2013.
[2] http://en.wikipedia.org/wiki/Load_Sharing_Facility, date of visit: 15.02.2013.
[3] https://computing.llnl.gov/linux/slurm, date of visit: 15.02.2013.
[4] http://en.wikipedia.org/wiki/Oracle_Grid_Engine, date of visit: 15.02.2013.

- Robustness of features suffers from a strong data variance between the training set and other sets. We have already mentioned that the automatic guitar identification trained on mixtures of guitar, piano, wind, and strings may perform unexpectedly, if it is applied on other mixtures, for example, of guitar, gong, and harp. The same holds for other high-level features: if only popular music pieces are labelled with the Energetic mood, some of the powerful orchestral arrangements would be always categorised as 'non-energetic'. A possible solution is to integrate an interactive semi-supervised approach: first, some unlabelled data clusters are identified, which strongly differ from the available labelled data. Then, an expert opinion is required to estimate the relevance of these data: it is still possible that it corresponds to less relevant outliers. Finally, the ground truth for the extended training set can be provided.
- Many high-level features should or could only be learned from real-world polyphonic recordings. In that case, a high amount of expert labelling efforts is unavoidable. For example, the onsets and occurrences of each instrument must be precisely notated. This holds not only for the instruments: a 'melancholic' music piece can contain a short harmonic variation in major. This task can be addressed in a more artificial, generative way: if enough rules are provided by the music experts, such as 'melancholic music corresponds to minor key and low loudness', it is possible to randomly generate a sufficient number of corresponding audio recordings.
- Another goal is to integrate more natural language descriptions for music categorisation with the help of fuzzy approaches. Even the interpretable decision tree models are often large and confusing for a non-expert. More comprehensible rules would describe the music in a natural way, e.g., 'the personal preferences for car driving: non-vocal segments with a large piano share, a large vocal share in general, a balanced distribution between male and female vocals, from time to time intermediate segments dominated by strings or organ'. Another example of a fuzzy application is automatic playlist adjustment: 'slightly increase the share of dance pop songs at the beginning of the party, change to house at the later hours, and switch rapidly to classical music at the very end of the event'. Fuzzy logic for music classification was explicitly recommended in [139] as approach, which "would significantly improve the quality of ground truth, and would make the evaluation of systems more realistic". Even if it has been already investigated in some works [235, 59], it still remains a less explored domain.
- Robust high-level features may allow new user-centered recommendation applications, which combine high-level features of several personal categories: e.g., if a music listener has created a classical music category and a hard rock category, it is possible to recommend her or him 'simphonic metal' songs, which combine some high-level characteristics from both genres.

Evaluation can be done more thoroughly for the reliable proof of the generalisation ability of classification models:

- A simple approach, which on the other side is very time intensive, is to increase the number of the statistical repetitions for each experiment.
- Nested validation can be integrated. Then, the optimisation can be stopped, after its generalisation performance begins to decrease. We mentioned the early stopping

approach from [124] in Section 3.4.1. Again, this extension is time consuming.

- More different songs from different genres can be distributed across the data sets. Though our experiment and holdout sets do not contain the same songs, the songs for these sets were drawn from the same albums. This dependency can be reduced in future.

The SCALE OF THE EXPERIMENT STUDIES can be enlarged:

- The number of the classification categories can be increased.
- Many further high-level features can be derived by means of sliding feature selection or other approaches. The recognition of instruments and digital effects may provide a significant increase of the classification quality, in particular, for the categorisation into subgenres and user-specific preferences.
- Further classification methods can be selected based on their relevant properties. For example, the k-nearest neighbour classifier has higher model storage demands, but can be valuable because of the high model interpretability in the high-level feature domain. Since we have observed that the combination of several classifiers led to a classification performance increase, it is reasonable to integrate also ensemble methods or meta-classifiers, such as AdaBoost [82].

Multi-objective feature selection and optimisation by many COMBINATIONS OF METRICS from our lists in Section 4.1 depends on concrete preferences of the application scenario:

- First, the evaluation can be extended to different confusion matrix metrics, as done for recall and specificity in [217].
- A further step is to simultaneously optimise the metrics of different groups. Especially, resource metrics and user related metrics are often in conflict with the classification quality.
- If the evaluation is done using four or more conflicting, but also loosely correlated metrics, the many-objective optimisation can be applied (as mentioned in note 4, Section 3.2.4). Then, specific challenges of this application domain should be addressed: for example, it becomes harder to compare the solutions, and exponentially larger populations are required to provide enough trade-off solutions.

Several further METHOD ENHANCEMENTS are possible:

- One of the very challenging tasks is to enable faster classification, which is required to satisfy listener expectations on an automatic classifier or a recommendation system. But it is also motivated by the limited resources of mobile devices and the large growth of personal music collections.
- Multi-class and, in particular, multi-label classification, are well suited for music categorisation. A single song may belong exclusively to one genre, but may appear in several user preference categories or contribute to different moods.
- Semi-supervised learning is reasonable for real-world situations, where the ground truth is not always available, or its definition requires high efforts. It is again well suited to learn individual listener preferences.

- Another group of classification methods, which may further reduce the efforts for the labelling of ground truth, is the classification only by positives.
- The application of evolutionary algorithms for multi-objective feature selection or any other optimisation in music classification can be extended by further tuning of the optimisation methods. For example, memetic algorithms performed quite well in our single-objective studies [223, 15] and can be integrated into the multi-objective approach. Self-adaptation or predator-prey extensions, which are mentioned in Section 3.2.1, may be also applicable.

Summarising this discussion, we believe that music classification, but also MIR in general, would strongly benefit from a further integration of COMPUTATIONAL INTELLIGENCE (CI) techniques. The focus of the major CI research fields in that content could be as follows:

- FUZZY LOGIC may help to handle classification scenarios in a natural way and to increase the comprehensibility of the dependencies between high-level features and the categories.
- EVOLUTIONARY ALGORITHMS, in particular, multi- and many-objective evolutionary algorithms, may facilitate the evaluation and optimisation of the classification performance from different perspectives.
- NEURAL NETWORKS AND SUPPORT VECTOR MACHINES are probably less preferable for the classification of music categories from high-level features, because of their less interpretable models. But they may be essential for the robust recognition of the high-level features themselves.

A. Feature Lists

The following tables list all low-level and high-level features used in this thesis. The column values and variables have the following meaning:

- **Name**: feature name.
- **Ref.**: reference to feature definition.
- W_e: extraction frame size in samples.
- S_e: extraction frame step size in samples.
- F^{**}: number of feature dimensions.
- **ID**: unique AMUSE feature ID.
- **HL**: in the tables with both high-level and low-level features, high-level features are marked with 'H' and low-level with 'L'.
- W_a: structural complexity algorithm frame size in seconds.
- W_e^{SC}: structural complexity extraction frame size in samples.
- S_e^{SC}: structural complexity extraction frame step size in samples.

A.1. Timbre and energy features (low-level)

Name	Ref.	$W_e = S_e$	F^{**}	Tool	ID
TIME DOMAIN					
Linear prediction coefficients	[206]	512	10	jAudio	1
Low energy	[206]	512	1	jAudio	6
Root mean square	[206]	512	1	jAudio	4
RMS peak number in 3 seconds	[115]	66,150	1	Matlab	11
RMS peak number above half of maximum peak in 3 seconds	[115]	66,150	1	Matlab	12
Zero-crossing rate	[206]	512	1	jAudio	0
SPECTRAL DOMAIN					
Average distance between extremal spectral values and its variance	[206]	512	2	Yale	2
Average distance between zero-crossings of the time-domain signal and its variance	[206]	512	2	Yale	3
Normalised energy of harmonic components	[206]	512	1	Yale	7
Onset envelope LPCs for blocks	[51]	52,920	125	Matlab	51
Onset envelope	[51]	52,920	132	Matlab	52
Spectral bandwidth	[206]	512	1	Yale	16
Spectral brightness	[115]	512	1	MIR Toolbox	23

continued on next page

continued from previous page

Name	Ref.	$W_e = S_e$	F^{**}	Tool	ID
Spectral centroid	[206]	512	1	Yale	14
Spectral crest factor	[206]	512	4	Yale	19
Spectral discrepancy	[206]	512	1	Yale	31
Spectral extent	[206]	512	1	Yale	21
Spectral flatness measure	[206]	512	4	Yale	20
Spectral flux	[206]	512	1	jAudio	22
Spectral irregularity	[115]	512	1	MIR Toolbox	15
Spectral kurtosis	[206]	512	1	Yale	18
Spectral skewness	[206]	512	1	Yale	17
Spectral slope	[206]	512	1	Yale	29
Sensory roughness	[115]	1,024	1	MIR Toolbox	24
Sub-band energy ratio	[206]	512	4	Yale	25
Tristimulus	[206]	512	2	Matlab	10
y-axis intercept	[206]	512	1	Yale	30
CEPSTRAL DOMAIN					
CMRARE cepstral modulation features with polynomial order 3	[133]	110,250	8	Matlab	45
CMRARE cepstral modulation features with polynomial order 5	[133]	110,250	12	Matlab	46
CMRARE cepstral modulation features with polynomial order 10	[133]	110,250	22	Matlab	47
Delta MFCCs	[115]	512	13	MIR Toolbox	48
Mel frequency cepstral coefficients	[206]	512	13	jAudio	28
Mel frequency cepstral coefficients	[206]	512	20	Matlab	38
Mel frequency cepstral coefficients	[206]	512	13	MIR Toolbox	39
Onset envelope MFCCs	[51]	52,920	16	Matlab	49
Onset envelope MFCCs for blocks	[51]	52,920	80	Matlab	50
PHASE DOMAIN					
Angles in phase domain	[206]	512	1	Yale	32
Distances in phase domain	[206]	512	1	Yale	33
ERB AND BARK DOMAINS					
Bark scale magnitudes	[115]	512	23	MIR Toolbox	40
Root mean square for ERB bands	[115]	512	10	MIR Toolbox	61
Spectral centroid for ERB bands	[115]	512	10	MIR Toolbox	62
Zero-crossing rate for ERB bands	[115]	512	10	MIR Toolbox	60

A.2. Chroma and harmony features (low-level and high-level)

Name	Ref.	$W_e = S_e$	F^{**}	Tool	ID	HL
CHROMA AND RELATED CHARACTERISTICS						
Amplitude, position and width of the 1st spectral peak	[206]	512	3	Yale	211	L
Amplitude, position and width of the 2nd spectral peak	[206]	512	3	Yale	212	L
Amplitude, position and width of the 3rd spectral peak	[206]	512	3	Yale	213	L
Amplitude, position and width of the 4th spectral peak	[206]	512	3	Yale	214	L
Amplitude, position and width of the 5th spectral peak	[206]	512	3	Yale	215	L
Bass chroma	[135]	2,048	12	NNLS Chroma	251	L
Chroma	[115]	512/4,096	12	MIR Toolbox	206	L

continued on next page

continued from previous page						
Name	**Ref.**	$W_e = S_e$	F^{**}	**Tool**	**ID**	**HL**
Chroma	[135]	2,048	12	NNLS Chroma	250	L
Chroma and normalised chroma	[206]	512	24	Yale	204	L
Chroma DCT-reduced log pitch	[155]	4,410	12	Chroma Toolb.	219	L
Chroma energy normalised statistics	[154]	4,410	12	Chroma Toolb.	218	L
Chroma maximum	[206]	512	1	Yale	205	H
Chroma tone with the maximum strength	[206]	512	1	Yale	207	H
Fundamental frequency	[206]	512	1	Matlab	200	L
Inharmonicity	[115]	512	1	MIR Toolbox	201	L
Semitone spectrum	[135]	2,048	85	NNLS Chroma	252	L
HARMONY						
Consonance	[135]	2,048	1	NNLS Chroma	255	H
Harmonic change	[135]	2,048	1	NNLS Chroma	254	H
Harmonic change detection function	[115]	512/4,096	1	MIR Toolbox	217	H
Interval strengths from the 10 highest semitone values	Sect. 2.2.3.2	2,048	12	AMUSE	260	H
Interval strengths from the semitone spectrum above 3/4 of its maximum value	Sect. 2.2.3.2	2,048	12	AMUSE	261	H
Key and its clarity	[115]	512/4,096	2	MIR Toolbox	202	H
Local tuning	[135]	8,192	1	NNLS Chroma	253	H
Major/minor alignment	[115]	512/4,096	1	MIR Toolbox	203	H
Strengths of CRP cooccurrences	Sect. 2.2.3.2	4,410	66	Matlab	220	H
Strengths of major keys	[115]	512/4,096	12	MIR Toolbox	209	H
Strengths of minor keys	[115]	512/4,096	12	MIR Toolbox	210	H
Tonal centroid vector	[115]	512/4,096	6	MIR Toolbox	216	H
CHORD ANALYSIS						
Number of different chords in 10 s	Sect. 2.2.3.2	220,500	1	AMUSE	257	H
Number of chord changes in 10 s	Sect. 2.2.3.2	220,500	1	AMUSE	258	H
Shares of the most frequent 20, 40, and 60 per cent of chords with regard to their duration	Sect. 2.2.3.2	220,500	3	AMUSE	259	H

A.3. Temporal characteristics (low-level and high-level)

Name	**Ref.**	$W_e = S_e$	F^{**}	**Tool**	**ID**	**HL**
TEMPORAL AND CORRELATION CHARACTERISTICS						
Duration of music piece	[206]	-1	1	Matlab	400	H
Estimated beat number per minute	[206]	229,376	1	Matlab	421	H
Estimated tatum number per minute	[206]	229,376	1	Matlab	422	H
Estimated onset number per minute	[206]	229,376	1	Matlab	420	H
Tempo based on onset times	[115]	66,150	1	MIR Toolbox	425	H
First periodicity peak	[206]	131,072	1	Yale	405	L
First relative periodicity amplitude peak	[206]	131,072[a]	1	jAudio	402	L
Sum of correlated components	[206]	131,072	1	jAudio	407	L
RHYTHM						
Five peaks of fluctuation curves summed across all bands	[115]	229,376	5	MIR Toolbox	427	L
Characteristics of fluctuation patterns	[206]	32,768	7	Matlab	410	L
Rhythmic clarity	[115]	66,150	1	MIR Toolbox	418	H
STRUCTURE						
Segmentation characteristics	Sect. 2.2.3.3	-1	3	AMUSE	602	H

[a] 256 short RMS frames are estimated for beat histogram.

A.4. Instruments (high-level)

Name	Ref.	$W_e = 2S_e$	F^{**}	Tool	ID
Guitar C4.5 best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,000
Guitar C4.5 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,009
Guitar C4.5 all non-dominated solutions	Sect. 5.1.1	220,500	7	AMUSE	2,004
Guitar RF best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,001
Guitar RF 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,010
Guitar RF all non-dominated solutions	Sect. 5.1.1	220,500	8	AMUSE	2,005
Guitar NB best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,002
Guitar NB 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,011
Guitar NB all non-dominated solutions	Sect. 5.1.1	220,500	5	AMUSE	2,006
Guitar SVM best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,003
Guitar SVM 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,012
Guitar SVM all non-dominated solutions	Sect. 5.1.1	220,500	25	AMUSE	2,007
Guitar all non-dominated solutions	Sect. 5.1.1	220,500	15	AMUSE	2,008
Piano C4.5 best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,020
Piano C4.5 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,029
Piano C4.5 all non-dominated solutions	Sect. 5.1.1	220,500	6	AMUSE	2,024
Piano RF best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,021
Piano RF 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,030
Piano RF all non-dominated solutions	Sect. 5.1.1	220,500	7	AMUSE	2,025
Piano NB best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,022
Piano NB 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,031
Piano NB all non-dominated solutions	Sect. 5.1.1	220,500	8	AMUSE	2,026
Piano SVM best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,023
Piano SVM 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,032
Piano SVM all non-dominated solutions	Sect. 5.1.1	220,500	24	AMUSE	2,027
Piano all non-dominated solutions	Sect. 5.1.1	220,500	8	AMUSE	2,028
Wind C4.5 best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,040
Wind C4.5 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,049
Wind C4.5 all non-dominated solutions	Sect. 5.1.1	220,500	7	AMUSE	2,044
Wind RF best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,041
Wind RF 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,050
Wind RF all non-dominated solutions	Sect. 5.1.1	220,500	11	AMUSE	2,045
Wind NB best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,042
Wind NB 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,051
Wind NB all non-dominated solutions	Sect. 5.1.1	220,500	4	AMUSE	2,046
Wind SVM best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,043
Wind SVM 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,052
Wind SVM all non-dominated solutions	Sect. 5.1.1	220,500	15	AMUSE	2,047
Wind all non-dominated solutions	Sect. 5.1.1	220,500	14	AMUSE	2,048
Strings C4.5 best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,060
Strings C4.5 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,069
Strings C4.5 all non-dominated solutions	Sect. 5.1.1	220,500	4	AMUSE	2,064
Strings RF best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,061
Strings RF 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,070
Strings RF all non-dominated solutions	Sect. 5.1.1	220,500	11	AMUSE	2,065
Strings NB best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,062
Strings NB 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,071
Strings NB all non-dominated solutions	Sect. 5.1.1	220,500	11	AMUSE	2,066
Strings SVM best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,063

continued on next page

continued from previous page

Name	Ref.	$W_e = 2S_e$	F^{**}	Tool	ID
Strings SVM 2nd best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,072
Strings SVM all non-dominated solutions	Sect. 5.1.1	220,500	20	AMUSE	2,067
Strings all non-dominated solutions	Sect. 5.1.1	220,500	11	AMUSE	2,068

A.5. Moods (high-level)

Name	Ref.	$W_e = 2S_e$	F^{**}	Tool	ID
Aggressive C4.5 best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,000
Aggressive C4.5 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,001
Aggressive RF best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,002
Aggressive RF 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,003
Aggressive NB best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,004
Aggressive NB 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,005
Aggressive SVM best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,006
Aggressive SVM 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,007
Confident C4.5 best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,020
Confident C4.5 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,021
Confident RF best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,022
Confident RF 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,023
Confident NB best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,024
Confident NB 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,025
Confident SVM best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,026
Confident SVM 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,027
Earnest C4.5 best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,040
Earnest C4.5 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,041
Earnest RF best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,042
Earnest RF 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,043
Earnest NB best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,044
Earnest NB 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,045
Earnest SVM best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,046
Earnest SVM 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,047
Energetic C4.5 best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,060
Energetic C4.5 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,061
Energetic RF best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,062
Energetic RF 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,063
Energetic NB best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,064
Energetic NB 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,065
Energetic SVM best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,066
Energetic SVM 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,067
PartyCelebratory C4.5 best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,080
PartyCelebratory C4.5 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,081
PartyCelebratory RF best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,082
PartyCelebratory RF 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,083
PartyCelebratory NB best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,084
PartyCelebratory NB 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,085
PartyCelebratory SVM best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,086
PartyCelebratory SVM 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,087
Reflective C4.5 best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,100
Reflective C4.5 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,101

continued on next page

continued from previous page

Name	Ref.	$W_e = 2S_e$	F^{**}	Tool	ID
Reflective RF best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,102
Reflective RF 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,103
Reflective NB best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,104
Reflective NB 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,105
Reflective SVM best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,106
Reflective SVM 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,107
Sentimental C4.5 best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,120
Sentimental C4.5 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,121
Sentimental RF best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,122
Sentimental RF 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,123
Sentimental NB best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,124
Sentimental NB 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,125
Sentimental SVM best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,126
Sentimental SVM 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,127
Stylish C4.5 best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,140
Stylish C4.5 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,141
Stylish RF best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,142
Stylish RF 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,143
Stylish NB best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,144
Stylish NB 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,145
Stylish SVM best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,146
Stylish SVM 2nd best model (m_{BRE})	Sect. 5.1.2	529,200	1	AMUSE	4,147

A.6. GFKL-2011 (high-level)

Name	Ref.	$W_e = 2S_e$	F^{**}	Tool	ID
Effects distortion C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,020
Effects distortion C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,021
Effects distortion RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,022
Effects distortion RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,023
Effects distortion NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,024
Effects distortion NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,025
Effects distortion SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,026
Effects distortion SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,027
Harmony major C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,180
Harmony major C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,181
Harmony major RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,182
Harmony major RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,183
Harmony major NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,184
Harmony major NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,185
Harmony major SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,186
Harmony major SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,187
Harmony minor C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,200
Harmony minor C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,201
Harmony minor RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,202
Harmony minor RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,203
Harmony minor NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,204
Harmony minor NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,205
Harmony minor SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,206
Harmony minor SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,207
Instrumentation drums C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,220

continued on next page

continued from previous page

Name	Ref.	$W_e = 2S_e$	F^{**}	Tool	ID
Instrumentation drums C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,221
Instrumentation drums RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,222
Instrumentation drums RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,223
Instrumentation drums NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,224
Instrumentation drums NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,225
Instrumentation drums SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,226
Instrumentation drums SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,227
Level of activation high C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,000
Level of activation high C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,001
Level of activation high RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,002
Level of activation high RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,003
Level of activation high NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,004
Level of activation high NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,005
Level of activation high SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,006
Level of activation high SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,007
Melodic range $\leq$ octave C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,260
Melodic range $\leq$ octave C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,261
Melodic range $\leq$ octave RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,262
Melodic range $\leq$ octave RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,263
Melodic range $\leq$ octave NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,264
Melodic range $\leq$ octave NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,265
Melodic range $\leq$ octave SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,266
Melodic range $\leq$ octave SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,267
Melodic range $>$ octave C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,240
Melodic range $>$ octave C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,241
Melodic range $>$ octave RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,242
Melodic range $>$ octave RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,243
Melodic range $>$ octave NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,244
Melodic range $>$ octave NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,245
Melodic range $>$ octave SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,246
Melodic range $>$ octave SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,247
Melodic range linearly C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,280
Melodic range linearly C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,281
Melodic range linearly RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,282
Melodic range linearly RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,283
Melodic range linearly NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,284
Melodic range linearly NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,285
Melodic range linearly SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,286
Melodic range linearly SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,287
Melodic range volatile C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,300
Melodic range volatile C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,301
Melodic range volatile RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,302
Melodic range volatile RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,303
Melodic range volatile NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,304
Melodic range volatile NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,305
Melodic range volatile SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,306
Melodic range volatile SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,307
Singing solo clear C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,040
Singing solo clear C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,041
Singing solo clear RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,042
Singing solo clear RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,043
Singing solo clear NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,044

continued on next page

continued from previous page

Name	**Ref.**	$W_e = 2S_e$	F^{**}	**Tool**	**ID**
Singing solo clear NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,045
Singing solo clear SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,046
Singing solo clear SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,047
Singing solo man C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,140
Singing solo man C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,141
Singing solo man RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,142
Singing solo man RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,143
Singing solo man NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,144
Singing solo man NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,145
Singing solo man SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,146
Singing solo man SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,147
Singing solo polyphonic C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,160
Singing solo polyphonic C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,161
Singing solo polyphonic RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,162
Singing solo polyphonic RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,163
Singing solo polyphonic NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,164
Singing solo polyphonic NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,165
Singing solo polyphonic SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,166
Singing solo polyphonic SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,167
Singing solo rough C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,080
Singing solo rough C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,081
Singing solo rough RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,082
Singing solo rough RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,083
Singing solo rough NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,084
Singing solo rough NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,085
Singing solo rough SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,086
Singing solo rough SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,087
Singing solo unison C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,100
Singing solo unison C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,101
Singing solo unison RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,102
Singing solo unison RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,103
Singing solo unison NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,104
Singing solo unison NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,105
Singing solo unison SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,106
Singing solo unison SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,107
Singing solo woman C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,120
Singing solo woman C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,121
Singing solo woman RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,122
Singing solo woman RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,123
Singing solo woman NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,124
Singing solo woman NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,125
Singing solo woman SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,126
Singing solo woman SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,127
Singing voice medium C4.5 best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,060
Singing voice medium C4.5 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,061
Singing voice medium RF best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,062
Singing voice medium RF 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,063
Singing voice medium NB best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,064
Singing voice medium NB 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,065
Singing voice medium SVM best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,066
Singing voice medium SVM 2nd best model (m_{BRE})	Sect. 5.1.3	529,200	1	AMUSE	6,067

A.7. Structural complexity characteristics and features for their estimation

Name	Ref.	W_e	F^{**}	Tool	ID
CHORD COMPLEXITY ($W_a \in \{10;20\}, W_e^{SC} = 2S_e^{SC} = 2,116,800$ (96 s))					
Number of different chords in 10 s	Sect. 2.2.3.2	220,500	1	AMUSE	257
Number of chord changes in 10 s	Sect. 2.2.3.2	220,500	1	AMUSE	258
Shares of the most frequent 20, 40, and 60 per cent of chords with regard to their duration	Sect. 2.2.3.2	220,500	3	AMUSE	259
CHROMA COMPLEXITY ($W_a \in \{2;4;8\}, W_e^{SC} = 2S_e^{SC} = 529,200$ (24 s))					
Bass chroma	[135]	2,048	12	NNLS Chroma	251
Chroma	[135]	2,048	12	NNLS Chroma	250
CHROMA RELATED COMPLEXITY ($W_a \in \{2;4;8\}, W_e^{SC} = 2S_e^{SC} = 529,200$ (24 s))					
Amplitude, position and width of the 1st spectral peak	[206]	512	3	Yale	211
Amplitude, position and width of the 2nd spectral peak	[206]	512	3	Yale	212
Amplitude, position and width of the 3rd spectral peak	[206]	512	3	Yale	213
Amplitude, position and width of the 4th spectral peak	[206]	512	3	Yale	214
Amplitude, position and width of the 5th spectral peak	[206]	512	3	Yale	215
Chroma maximum	[206]	512	1	Yale	205
Chroma tone with the maximum strength	[206]	512	1	Yale	207
Fundamental frequency	[206]	512	1	Matlab	200
Inharmonicity	[115]	512	1	MIR Toolbox	201
Harmonic change detection function	[115]	512/4,096	1	MIR Toolbox	217
HARMONY COMPLEXITY ($W_a \in \{2;4;8\}, W_e^{SC} = 2S_e^{SC} = 529,200$ (24 s))					
Chroma maximum	[206]	512	1	Yale	205
Chroma tone with the maximum strength	[206]	512	1	Yale	207
Consonance	[135]	2,048	1	NNLS Chroma	255
Harmonic change	[135]	2,048	1	NNLS Chroma	254
Harmonic change detection function	[115]	512/4,096	1	MIR Toolbox	217
Interval strengths estimated from 10 highest semitone values	Sect. 2.2.3.2	2,048	12	AMUSE	260
Interval strengths estimated from the semitone spectrum above 3/4 of the maximum value	Sect. 2.2.3.2	2,048	12	AMUSE	261
Key and its clarity	[115]	512/4,096	2	MIR Toolbox	202
Local tuning	[135]	8,192	1	NNLS Chroma	253
Major/minor alignment	[115]	512/4,096	1	MIR Toolbox	203
Number of different chords in 10 s	Sect. 2.2.3.2	220,500	1	AMUSE	257
Number of chord changes in 10 s	Sect. 2.2.3.2	220,500	1	AMUSE	258
Shares of the most frequent 20, 40, and 60 per cent of chords with regard to their duration	Sect. 2.2.3.2	220,500	3	AMUSE	259
Strengths of major keys	[115]	512/4,096	12	MIR Toolbox	209
Strengths of minor keys	[115]	512/4,096	12	MIR Toolbox	210
Tonal centroid vector	[115]	512/4,096	6	MIR Toolbox	216
INSTRUMENTS COMPLEXITY ($W_a \in \{10;20\}, W_e^{SC} = 2S_e^{SC} = 2,116,800$ (96 s))					
Guitar RF best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,001
Guitar SVM best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,003
Piano RF best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,021

continued on next page

continued from previous page

Name	**Ref.**	W_e	F^{**}	**Tool**	**ID**
Piano SVM best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,023
Wind RF best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,041
Wind SVM best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,043
Strings RF best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,061
Strings SVM best model (m_{RE})	Sect. 5.1.1	220,500	1	AMUSE	2,063
TEMPO AND RHYTHM COMPLEXITY ($W_a \in \{2;4;8\}, W_e^{SC} = 2S_e^{SC} = 529,200$ (24 s))					
Duration of music piece	[206]	-1	1	Matlab	400
Estimated beat number per minute	[206]	229,376	1	Matlab	421
Estimated tatum number per minute	[206]	229,376	1	Matlab	422
Estimated onset number per minute	[206]	229,376	1	Matlab	420
Tempo based on onset times	[115]	66,150	1	MIR Toolbox	425
Five peaks of fluctuation curves summed across all bands	[115]	229,376	5	MIR Toolbox	427
Characteristics of fluctuation patterns	[206]	32,768	7	Matlab	410
Rhythmic clarity	[115]	66,150	1	MIR Toolbox	418
TIMBRE COMPLEXITY ($W_a \in \{2;4;8\}, W_e^{SC} = 2S_e^{SC} = 529,200$ (24 s))					
Low energy	[206]	512	1	jAudio	6
Root mean square	[206]	512	1	jAudio	4
RMS peak number in 3 seconds	[115]	66,150	1	Matlab	11
RMS peak number above half of maximum peak in 3 seconds	[115]	66,150	1	Matlab	12
Zero-crossing rate	[206]	512	1	jAudio	0
Average distance between extremal spectral values and its variance	[206]	512	2	Yale	2
Average distance between zero-crossings of the time-domain signal and its variance	[206]	512	2	Yale	3
Normalised energy of harmonic components	[206]	512	1	Yale	7
Spectral bandwidth	[206]	512	1	Yale	16
Spectral brightness	[115]	512	1	MIR Toolbox	23
Spectral centroid	[206]	512	1	Yale	14
Spectral crest factor	[206]	512	4	Yale	19
Spectral discrepancy	[206]	512	1	Yale	31
Spectral extent	[206]	512	1	Yale	21
Spectral flatness measure	[206]	512	4	Yale	20
Spectral flux	[206]	512	1	jAudio	22
Spectral irregularity	[115]	512	1	MIR Toolbox	15
Spectral kurtosis	[206]	512	1	Yale	18
Spectral skewness	[206]	512	1	Yale	17
Spectral slope	[206]	512	1	Yale	29
Sensory roughness	[115]	1,024	1	MIR Toolbox	24
Sub-band energy ratio	[206]	512	4	Yale	25
Tristimulus	[206]	512	2	Matlab	10
y-axis intercept	[206]	512	1	Yale	30
Delta MFCCs	[115]	512	13	MIR Toolbox	48
Mel frequency cepstral coefficients	[206]	512	13	MIR Toolbox	39
Angles in phase domain	[206]	512	1	Yale	32
Distances in phase domain	[206]	512	1	Yale	33

B. Song Lists

The following tables describe our song database, where the column names correspond to:

- **Interpret**: artist, band, or composer (for classic).
- **Album**: album name.
- **Genre**: genre (exclusive). 120 albums are distributed across the six genres: 45 Pop/Rock, 15 Classic, 15 Electronic, 15 Jazz, 15 R'n'B, 15 Rap.
- **Styles**: albums, which were marked by AMG experts as belonging to one or several styles used in our experiments.
- **N**: album track number.
- **Song**: song name.

B.1. Genre and style album distribution

Interpret	Album	Genre	Styles
2Pac	Me Against The World	Rap	
2raumwohnung	In Wirklich	Electronic	
AC/DC	Back In Black	Pop/Rock	HeavyMetal
ATB	Dedicated	Electronic	ClubDance
Abba	Gold	Pop/Rock	
Aim	Cold Water Music	Electronic	
Alan Parsons Project, The	Tales Of Mystery And Imagination-Edgar Allan Poe	Pop/Rock	ProgRock
Amos, Tori	The Beekeeper	Pop/Rock	
Anastacia	Anastacia	Pop/Rock	
Armstrong, Louis	All-Time Greatest Hits	Jazz	
Arrested Development	3 Years 5 Months And 2 Days In The Life Of	Rap	
Ashanti	Ashanti	R'n'B	
BAP	Für Usszeschnigge	Pop/Rock	
Bach, Johann Sebastian	Italienisches Konzert etc - Alfred Brendel	Classic	
Bach, Johann Sebastian	Organ Works	Classic	
Baker, Chet	Jazz Masters 32	Jazz	
Barclay James Harvest	Best Of Barclay James Harvest	Pop/Rock	ProgRock
Basie, Count	Portrait	Jazz	
Beastie Boys	Licensed To Ill	Rap	
Beethoven, Ludwig van	Piano Sonatas-Maria-Joao Pires	Classic	
Benoit, David	Urban Daydreams	Jazz	
Berlioz, Hector	Symphonie Fantastique - Orchester de RTL	Classic	

continued on next page

continued from previous page

Interpret	**Album**	**Genre**	**Styles**
Boney M	The Magic Of Boney M	R'n'B	ClubDance
Braxton, Toni	Toni Braxton	R'n'B	
Brecker Brothers The	Out Of The Loop	R'n'B	
Brecker, Michael	Tales From The Hudson	Jazz	
Burgh, Chris de	Spanish Train And Other Stories	Pop/Rock	ProgRock
Busta Rhymes	When Disaster Strikes	Rap	
Carey, Mariah	Daydream	R'n'B	ClubDance
Charles, Ray	Ray Soundtrack	R'n'B	
Chemical Brothers, The	We Are The Night	Electronic	ClubDance
Chicago	17	Pop/Rock	
Chopin, Frederic	Horowitz Plays Chopin	Classic	
Chopin, Frederic	Waltzes-Vladimir Ashkenazy	Classic	
Coldplay	X And Y	Pop/Rock	
Collins, Phil	Both Sides	Pop/Rock	
Coltrane, John	The Very Best Of John Coltrane	Jazz	
Cooke, Sam	Sam Cooke	R'n'B	
Coolio	The Return Of The Gangsta	Rap	
Corrs, The	In Blue	Pop/Rock	
Cosmic Gate	Rhythm And Drums	Electronic	
Cypress Hill	Skull And Bones	Rap	
Davis, Miles	Kind Of Blue	Jazz	
Depeche Mode	The Singles	Pop/Rock	ClubDance
Destiny's Child	Destiny's Child	R'n'B	ClubDance
Diamond, Neil	Serenade	Pop/Rock	
Dire Straits	Love Over Gold	Pop/Rock	
Disturbed	Ten Thousand Fists	Pop/Rock	
Dr. Dre	2001	Rap	
Dream Theater	Images And Words	Pop/Rock	ProgRock, HeavyMetal
Ellington, Duke With Charles Mingus And Max Roach	Money Jungle	Jazz	
Eminem	The Eminem Show	Rap	
Eurythmics	Peace	Pop/Rock	
Faithless	Sunday 8pm Special Edition	Electronic	ClubDance
Fatboy Slim	Palookaville	Electronic	ClubDance
Foo Fighters	One By One	Pop/Rock	
Foxy Brown	Chyna Doll	Rap	
Franklin, Aretha	Collections	R'n'B	
Furtado, Nelly	Loose	Pop/Rock	
Gaye, Marvin	Midnight Love	R'n'B	
Genesis	We Can't Dance	Pop/Rock	ProgRock
Glen, Marla	This Is Marla Glen	R'n'B	
Gnarls Barkley	St. Elsewhere	Rap	
Grandmaster Mele-Mel And Scorpio	Right Now	Rap	
Grönemeyer, Herbert	4630 Bochum	Pop/Rock	
Groove Armada	LoveBox	Electronic	ClubDance
Haendel, Georg Friedrich	Organ Concertos Op. 4 - Rudolph Ewerhart	Classic	
Hancock, Herbie - Brecker, Michael - Hargrove, Roy	Directions In Music	Jazz	
Haydn, Joseph	Piano Sonatas - Carmen Piazzini	Classic	
In Flames	Clayman	Pop/Rock	HeavyMetal

continued on next page

continued from previous page

Interpret	**Album**	**Genre**	**Styles**
Irish Music	Green Green Grass	Pop/Rock	
Jacques Loussier Trio	Vivaldi - The Four Seasons	Jazz	
Jarre, Jean-Michel	Images	Electronic	
Joel, Billy	2000 Years The Millennium Concert	Pop/Rock	
John, Elton	Made In England	Pop/Rock	
Jones, Norah	Feels Like Home	Pop/Rock	
Kruder And Dorfmeister	The K And D Sessions	Electronic	
Madonna	Confessions On A Dance Floor	Pop/Rock	ClubDance
Madsen	Goodbye Logik	Pop/Rock	
Mann, Herbie	Just Wailin'	Jazz	
Massive Attack	Blue Lines	Electronic	ClubDance
Mendelssohn And Schubert	Symph No 4 Italian-Symph No 8 Unfinished-Giuseppe Sinopoli	Classic	
Miller, Glenn	Portrait	Jazz	
Mozart, Wolfgang Amadeus	Frühe Salzburger Meistersinfonien-Kölner Kammerorchester	Classic	
Muse	Showbiz	Pop/Rock	
Mussorgsky And Ravel	Bilder einer Ausstellung - Bolero - Berliner Philarmoniker	Classic	
Nightwish	Century Child	Pop/Rock	HeavyMetal
Nils Landgren Funk Unit	Fonk Da World	Pop/Rock	
Nirvana	Nevermind	Pop/Rock	
Orff, Carl	Carmina Burana	Classic	
Outkast	Stankonia	Rap	
Parker, Charlie	Portrait	Jazz	
Prodigy	The Fat Of The Land	Electronic	ClubDance
Queen	Greatest Hits	Pop/Rock	HeavyMetal
Rihanna	Music Of The Sun	R'n'B	
Rollins, Sonny	Portrait	Jazz	
Ross, Diana	Blue	R'n'B	
Roxette	Room Service	Pop/Rock	
Schumann, Robert	Concert Pieces With Orchestra - Sinfonieorchester des Südwestfunks	Classic	
Scooter	Back To The Heavyweight Jam	Electronic	
Sex Pistols	Never Mind The Bollocks	Pop/Rock	
Sibelius, Jean	Symphonien Nos. 5 And 6	Classic	
Smetana, Bedrich	The Moldau-Wiener Philarmoniker	Classic	
Smolski, Victor	Majesty And Passion	Pop/Rock	
Snoop Doggy Dogg	Doggystyle	Rap	
Snow Patrol	Eyes Open	Pop/Rock	
Soulfly	Conquer	Pop/Rock	HeavyMetal
Steely Dan	Pretzel Logic	Pop/Rock	
Stern, Leni	Like One	Jazz	
Stewart, Al	Year Of The Cat	Pop/Rock	ProgRock
Sylver	Chances	Electronic	
Therion	Secret Of The Runes	Pop/Rock	HeavyMetal
Timbaland	Presents Shock Value	R'n'B	
Toto	The Seventh One	Pop/Rock	
Usher	Confessions	R'n'B	
Van Dyk, Paul	Zurdo	Electronic	
Wayne, Jeff	War Of The Worlds	Pop/Rock	ProgRock
Wir sind Helden	Von hier an blind	Pop/Rock	

continued on next page

continued from previous page

Interpret	Album	Genre	Styles
X-Cutioners	X-Pressions	Rap	
Xzibit	Weapons Of Mass Destruction	Rap	

B.2. Genre and style training sets

Interpret	Album	N	Song
CLASSIC - POSITIVE SONGS			
Chopin, Frederic	Waltzes-Vladimir Ashkenazy	08	As-dur op. 64 No.3
Haydn, Joseph	Piano Sonatas - Carmen Piazzini	12	Sonata No.4 in E major Hob.XVI13-II. Menuet. Trio
Mussorgsky And Ravel	Bilder einer Ausstellung - Bolero - Berliner Philarmoniker	04	Ravel Rapsodie Espagnole-3. Habanera
Orff, Carl	Carmina Burana	07	Floret silva
Sibelius, Jean	Symphonien Nos. 5 And 6	01	Symphonie No.5 Es-dur op.82 - 1. Tempo molto moderato - Largamente
Beethoven, Ludwig van	Piano Sonatas-Maria-Joao Pires	10	Sonata No.23 in F minor op. 57 Appassionata-Allegro assai
Haendel, Georg Friedrich	Organ Concertos Op. 4 - Rudolph Ewerhart	01	Opus 4 No.1-Larghetto e staccato
Mendelssohn And Schubert	Symph No 4 Italian-Symph No 8 Unfinished-Giuseppe Sinopoli	02	Schubert Symphony No. 8 Unfinished-II. Andante con moto
Schumann, Robert	Concert Pieces With Orchestra - Sinfonieorchester des Südwestfunks	05	Konzertstück for Four Horns op.86-Romanze
Smetana, Bedrich	The Moldau-Wiener Philarmoniker	06	The Bartered Bride-Furiant
CLASSIC - NEGATIVE SONGS			
Charles, Ray	Ray Soundtrack	04	Drown In My Own Tears
Foxy, Brown	Chyna Doll	09	Bonnie And Clyde Part II
Franklin, Aretha	Collections	04	Mockingbird
Grönemeyer, Herbert	4630 Bochum	10	Mambo
Madonna	Confessions On A Dance Floor	04	Future Lovers
ATB	Dedicated	01	Dedicated
Amos, Tori	The Beekeeper	16	Hoochie Woman
Brecker Brothers The	Out Of The Loop	03	Scrunch
Jarre, Jean-Michel	Images	17	Rendez-Vous 2
Rollins, Sonny	Portrait	07	No Moe
POP - POSITIVE SONGS			
Alan Parsons Project, The	Tales of Mystery And Imagination-Edgar Allan Poe	06	The Fall Of The House Of Usher-I Prelude
Diamond, Neil	Serenade	07	Reggae Strut
Disturbed	Ten Thousand Fists	14	Avarice
Furtado, Nelly	Loose	08	Glow
Joel, Billy	2000 Years The Millennium Concert	14	Goodnight Saigon
Dream Theater	Images And Words	01	Pull Me Under
Eurythmics	Peace	03	Power To The Meek
Snow Patrol	Eyes Open	01	You're All I Have
Stewart, Al	Year Of The Cat	08	One Stage Before
Wayne, Jeff	War Of The Worlds	10	Brave New World
POP - NEGATIVE SONGS			

continued on next page

continued from previous page

Interpret	Album	N	Song
Aim	Cold Water Music	14	Underground Crown Holders Bonus Track
Arrested Development	3 Years 5 Months And 2 Days In The Life Of	10	U
Coltrane, John	The Very Best Of John Coltrane	08	Summertime
Jarre, Jean-Michel	Images	05	Computer Week-End
Van Dyk, Paul	Zurdo	05	Escape
Bach, Johann Sebastian	Organ Works	01	Toccata And Fuge BWV 565 d-moll-1. Toccata
Chemical Brothers The	We Are The Night	07	The Salmon Dance feat. Fatlip
Gnarls Barkley	St. Elsewhere	04	Gone Daddy Gone
Prodigy	The Fat Of The Land	10	Fuel My Fire
Usher	Confessions	10	Truth Hurts
RAP - POSITIVE SONGS			
2Pac	Me Against The World	13	Fuck The World
Dr. Dre	2001	15	Murder Ink
Eminem	The Eminem Show	03	Business
Grandmaster Mele-Mel And Scorpio	Right Now	08	Right Now
X-Cutioners	X-Pressions	12	Solve For X
Busta Rhymes	When Disaster Strikes	12	Rhymes Galore
Coolio	The Return Of The Gangsta	05	Drop Something feat. Brasa
Foxy Brown	Chyna Doll	10	456
Outkast	Stankonia	22	Slum Beautiful
Snoop Doggy Dogg	Doggystyle	07	Lodi Dodi
RAP - NEGATIVE SONGS			
Boney M	The Magic Of Boney M	15	Baby Do You Wanna B
Burgh, Chris de	Spanish Train And Other Stories	04	Patricia The Stripper
Chicago	17	05	Remember The Feeling
Franklin, Aretha	Collections	02	What A Difference A Day Makes
Wayne, Jeff	War Of The Worlds	06	The Red Weed Part 1
AC/DC	Back In Black	04	Given The Dog A Bone
Charles, Ray	Ray Soundtrack	11	Unchain My Heart
Chopin, Frederic	Horowitz Plays Chopin	13	Mazurka in f-Moll Op.59 No. 3
Ellington, Duke With Charles Mingus And Max Roach	Money Jungle	11	Backward Country Boy Blues
Queen	Greatest Hits	21	I Want It All
CLUBDANCE - POSITIVE SONGS			
Boney M	The Magic Of Boney M	05	No Woman No Cry
Chemical Brothers, The	We Are The Night	06	Das Spiegel
Faithless	Sunday 8pm Special Edition	11	Killers Lullaby
Groove Armada	LoveBox	03	Remember
Massive Attack	Blue Lines	02	One Love
Boney M	The Magic Of Boney M	14	Mary's Boy Child-Oh My Lord
Chemical Brothers, The	We Are The Night	12	The Pills Won't Help You Now feat. Midlake
Faithless	Sunday 8pm Special Edition	07	She's My Baby
Groove Armada	LoveBox	09	Easy
Massive Attack	Blue Lines	05	Five Man Army
CLUBDANCE - NEGATIVE SONGS			
Coldplay	X And Y	04	Fix You

continued on next page

continued from previous page

Interpret	Album	N	Song
Sex Pistols	Never Mind The Bollocks	04	Liar
Sibelius, Jean	Symphonien Nos. 5 And 6	08	Symphonie No.6 d-moll op.104-3. Poco vivace
Smetana, Bedrich	The Moldau-Wiener Philarmoniker	07	The Bartered Bride-Skocná
Stewart, Al	Year Of The Cat	07	Broadway Hotel
Armstrong, Louis	All-Time Greatest Hits	03	Sittin' In The Sun
Haendel, Georg Friedrich	Organ Concertos Op. 4 - Rudolph Ewerhart	16	Opus 4 No.5-Larghetto
Haydn, Joseph	Piano Sonatas - Carmen Piazzini	16	Sonata No.38 in D major Hob.XVI33-III. Tempo di Minuet
Jones, Norah	Feels Like Home	12	The Prettiest Thing
X-Cutioners	X-Pressions	15	Beat Treats
HeavyMetal - positive songs			
AC/DC	Back In Black	08	Have A Drink On Me
In Flames	Clayman	02	Pinball Map
Queen	Greatest Hits	13	Play The Game
Soulfly	Conquer	13	Sailing On
Therion	Secret Of The Runes	05	Schwarzalbenheim
AC/DC	Back In Black	06	Back In Black
In Flames	Clayman	06	Clayman
Queen	Greatest Hits	24	Its A Hard Life
Soulfly	Conquer	01	Blood Fire War Hate
Therion	Secret Of The Runes	10	Helheim
HeavyMetal - negative songs			
ATB	Dedicated	01	Dedicated
Corrs, The	In Blue	07	Irresistible
Cypress Hill	Skull And Bones	15	A Man
Mozart, Wolfgang Amadeus	Frühe Salzburger Meistersinfonien- Kölner Kammerorchester	10	Sinfonie D-Dur KV2022 Andantino con moto
Sibelius, Jean	Symphonien Nos. 5 And 6	08	Symphonie No.6 d-moll op.104 - 3. Poco vivace
Beastie Boys	Licensed To Ill	01	Rhymin' And Stealin'
Coolio	The Return Of The Gangsta	05	Drop Something feat. Brasa
Dire Straits	Love Over Gold	01	Telegraph Road
Eminem	The Eminem Show	09	Drips
Irish Music	Green Green Grass	12	Galway Town
ProgRock - positive songs			
Alan Parsons Project, The	Tales Of Mystery And Imagination- Edgar Allan Poe	06	The Fall Of The House Of Usher-I Prelude
Barclay James Harvest	Best Of Barclay James Harvest	03	Berlin
Burgh, Chris de	Spanish Train And Other Stories	08	Old Friend
Genesis	We Can't Dance	08	Living Forever
Wayne, Jeff	War Of The Worlds	01	The Eve Of The War
Alan Parsons Project, The	Tales Of Mystery And Imagination- Edgar Allan Poe	05	The System Of Doctor Tarr And Professor Fether
Barclay James Harvest	Best Of Barclay James Harvest	13	John Lennon's Guitar
Burgh, Chris de	Spanish Train And Other Stories	04	Patricia The Stripper
Genesis	We Can't Dance	11	Since I Lost You
Wayne, Jeff	War Of The Worlds	02	Horsell Common And The Heat Ray
ProgRock - negative songs			
Chicago	17	06	Along Comes A Woman
Cooke, Sam	Sam Cooke	05	Having A Party

continued on next page

continued from previous page

Interpret	Album	N	Song
Franklin, Aretha	Collections	04	Mockingbird
Jarre, Jean-Michel	Images	14	Moon Machine
Sibelius, Jean	Symphonien Nos. 5 And 6	04	Symphonie No.5 Es-dur op.82-3. Allegretto molto-Misterioso - Un pochettino largamente - Largamente assai
2Pac	Me Against The World	12	Old School
Bach, Johann Sebastian	Italienisches Konzert etc - Alfred Brendel	04	Chorale Prelude Ich ruf zu dir Herr Jesu Christ BWV 639 Arr. Busoni
Busta Rhymes	When Disaster Strikes	07	Turn It Up
Eurythmics	Peace	06	Peace Is Just A Word
John, Elton	Made In England	03	House

B.3. Genre and style optimisation and holdout sets

Interpret	Album	N	Song
Optimisation set OS120			
2Pac	Me Against The World	07	Heavy In The Game
2raumwohnung	In Wirklich	05	Freie Liebe
AC/DC	Back In Black	05	Let Me Put My Love Into You
ATB	Dedicated	04	You're Not Alone
Abba	Gold	04	Mamma Mia
Aim	Cold Water Music	13	Another Summer Bonus Track
Alan Parsons Project, The	Tales Of Mystery And Imagination-Edgar Allan Poe	09	The Fall Of The House Of Usher - IV Pavane
Amos, Tori	The Beekeeper	18	Marys Of The Sea
Anastacia	Anastacia	12	Maybe Today
Armstrong, Louis	All-Time Greatest Hits	06	It Takes Two To Tango
Arrested Development	3 Years 5 Months And 2 Days In The Life Of	07	Raining Revolution
Ashanti	Ashanti	09	Baby
BAP	Für Usszeschnigge	16	Waschsalon
Bach, Johann Sebastian	Italienisches Konzert etc - Alfred Brendel	06	Chromatic Fantasia And Fugue in D minor BWV 903
Bach, Johann Sebastian	Organ Works	07	Praeludium And Fuge BWV552 es-dur-1. Praeludium
Baker, Chet	Jazz Masters 32	10	Mean To Me
Barclay James Harvest	Best Of Barclay James Harvest	04	Child Of The Universe
Basie, Count	Portrait	09	One O'Clock Jump
Beastie Boys	Licensed To Ill	13	Time To Get Ill
Beethoven, Ludwig van	Piano Sonatas-Maria-Joao Pires	04	Sonata No.8 in C minor Op. 13 Pathetique-Grave-Allegro di molto e con brio
Benoit, David	Urban Daydreams	10	As If I Could Reach Rainbows
Berlioz, Hector	Symphonie Fantastique - Orchester de RTL	08	Damnation de Faust 3. Menuet des Follets
Boney M	The Magic Of Boney M	19	Sunny Remix By Mousse T
Braxton, Toni	Toni Braxton	02	Breathe Again
Brecker Brothers The	Out Of The Loop	05	African Skies
Brecker, Michael	Tales From The Hudson	02	Midnight Voyage

continued on next page

continued from previous page

Interpret	**Album**	**N**	**Song**
Burgh, Chris de	Spanish Train And Other Stories	03	This Song For You
Busta Rhymes	When Disaster Strikes	11	We Could Take It Outside
Carey, Mariah	Daydream	08	Long Ago
Charles, Ray	Ray Soundtrack	06	Mary Ann
Chemical Brothers The	We Are The Night	11	Harpoons
Chicago	17	01	Stay The Night
Chopin, Frederic	Horowitz Plays Chopin	14	Mazurka in f-Moll Op.7 No. 3
Chopin, Frederic	Waltzes-Vladimir Ashkenazy	09	As-dur op. 69 No.1
Coldplay	X And Y	07	Speed Of Sound
Collins, Phil	Both Sides	03	Everyday
Coltrane, John	The Very Best Of John Coltrane	11	Body And Soul
Cooke, Sam	Sam Cooke	11	Touch The Hem Of His Garment
Coolio	The Return Of The Gangsta	10	One More Night
Corrs, The	In Blue	06	Radio
Cosmic Gate	Rhythm And Drums	04	The Drums Video Mix
Cypress Hill	Skull And Bones	03	Rap Superstar
Davis, Miles	Kind Of Blue	03	Blue In Green
Depeche Mode	The Singles	29	Walking In My Shoes
Destiny's Child	Destiny's Child	02	No No No Part 2 feat. Wyclef Jean
Diamond, Neil	Serenade	02	Rosemary's Wine
Dire Straits	Love Over Gold	05	It Never Rains
Disturbed	Ten Thousand Fists	05	Stricken
Dr. Dre	2001	13	Bitch Niggaz
Dream Theater	Images And Words	05	Metropolis Part I The Miracle And The Sleeper
Ellington, Duke With Charles Mingus And Max Roach	Money Jungle	13	Switch Blade Alternate Take
Eminem	The Eminem Show	14	Hailies Song
Eurythmics	Peace	04	Beautiful Child
Faithless	Sunday 8pm Special Edition	10	Sunday 8pm
Fatboy Slim	Palookaville	05	Put It Back Together
Foo Fighters	One By One	06	Tired Of You
Foxy Brown	Chyna Doll	14	BWA
Franklin, Aretha	Collections	06	God Bless The Child
Furtado, Nelly	Loose	05	Showtime
Gaye, Marvin	Midnight Love	02	Sexual Healing
Genesis	We Can't Dance	02	Jesus He Knows Me
Glen Marla	This Is Marla Glen	08	Feet On The Ground
Gnarls Barkley	St. Elsewhere	01	Go-Go Gadget Gospel
Grandmaster Mele-Mel And Scorpio	Right Now	04	Mama
Grönemeyer, Herbert	4630 Bochum	03	Flugzeuge im Bauch
Groove Armada	LoveBox	11	But I Feel Good
Haendel, Georg Friedrich	Organ Concertos Op. 4 - Rudolph Ewerhart	21	Opus 4 No.6-Larghetto
Hancock, Herbie - Brecker, Michael - Hargrove, Roy	Directions In Music	04	Misstery
Haydn, Joseph	Piano Sonatas - Carmen Piazzini	17	Sonata No.32 in C-sharp minor Hob.XVI36-I. Moderato
In Flames	Clayman	09	Swim

continued on next page

continued from previous page

Interpret	**Album**	**N**	**Song**
Irish Music	Green Green Grass	11	Red Is The Rose
Jacques Loussier Trio	Vivaldi - The Four Seasons	06	Summer-Concerto No. 2 in G Minor-Presto
Jarre, Jean-Michel	Images	07	Ethnicolor 1
Joel, Billy	2000 Years The Millennium Concert	15	We Didn't Start The Fire
John, Elton	Made In England	05	Pain
Jones, Norah	Feels Like Home	05	In The Morning
Kruder And Dorfmeister	The K And D Sessions	15	Kruder And Dorfmeister Boogie Woogie
Madonna	Confessions On A Dance Floor	02	Get Together
Madsen	Goodbye Logik	02	Ein Sturm
Mann, Herbie	Just Wailin'	05	Jumpin With Symphony Sid
Massive Attack	Blue Lines	03	Blue Lines
Mendelssohn And Schubert	Symph No 4 Italian-Symph No 8 Unfinished-Giuseppe Sinopoli	04	Mendelssohn Symphony No. 4 Italian-II. Andante con moto
Miller, Glenn	Portrait	03	Chattanooga Choo Choo
Mozart, Wolfgang Amadeus	Frühe Salzburger Meistersinfonien-Kölner Kammerorchester	05	Sinfonie A-Dur KV2011Allegro moderato
Muse	Showbiz	02	Muscle Museum
Mussorgsky And Ravel	Bilder einer Ausstellung - Bolero - Berliner Philarmoniker	15	Mussorgsky Pictures At An Exhibition - 10. Samuel Goldenberg und Schmuyle
Nightwish	Century Child	05	Slaying The Dreamer
Nils Landgren Funk Unit	Fonk Da World	05	Anytime Anywhere
Nirvana	Nevermind	09	Lounge Act
Orff, Carl	Carmina Burana	20	Circa mea pectora
Outkast	Stankonia	10	I'll Call Before I Come
Parker, Charlie	Portrait	07	Victory Ball
Prodigy	The Fat Of The Land	05	Serial Thrilla
Queen	Greatest Hits	25	Breakthru
Rihanna	Music Of The Sun	12	Now I Know
Rollins, Sonny	Portrait	12	Newks Fadeaway
Ross, Diana	Blue	16	T'aint' Nobodys' Bizness If I Do
Roxette	Room Service	07	Bringing Me Down To My Knees
Schumann, Robert	Concert Pieces With Orchestra - Sinfonieorchester des Südwestfunks	07	Symphony No.1 op 38 Spring - II. Larghetto
Scooter	Back To The Heavyweight Jam	05	Fuck The Millenium
Sex Pistols	Never Mind The Bollocks	06	Problems
Sibelius, Jean	Symphonien Nos. 5 And 6	06	Symphonie No.6 d-moll op.104-1. Allegro molto moderato
Smetana, Bedrich	The Moldau-Wiener Philarmoniker	02	Má vlast-Vltava-Die Moldau
Smolski, Victor	Majesty And Passion	12	Concert for 2 Violins With Orchestra Chapter 3
Snoop Doggy Dogg	Doggystyle	10	Who Am I
Snow Patrol	Eyes Open	04	Shut Your Eyes
Soulfly	Conquer	08	Doom
Steely Dan	Pretzel Logic	03	Any Major Dude Will Tell You
Stern, Leni	Like One	01	Bubbles
Stewart, Al	Year Of The Cat	01	Lord Grenville
Sylver	Chances	02	Skin
Therion	Secret Of The Runes	03	Asgård

continued on next page

continued from previous page

Interpret	Album	N	Song
Timbaland	Presents Shock Value	08	Boardmeeting
Toto	The Seventh One	06	Stay Away
Usher	Confessions	07	Caught Up
Van Dyk, Paul	Zurdo	11	Otro Dia
Wayne, Jeff	War Of The Worlds	04	Forever Autumn
Wir sind Helden	Von hier an blind	04	Zuhälter
X-Cutioners	X-Pressions	19	Poetry In Motion
Xzibit	Weapons Of Mass Destruction	07	Judgement Day
Test set TS120			
2Pac	Me Against The World	02	If I Die 2Nite
2raumwohnung	In Wirklich	01	Da Sind Wir
ACDC	Back In Black	03	What Do You Do For Money Honey
ATB	Dedicated	09	I See It
Abba	Gold	11	Chiquitita
Aim	Cold Water Music	04	Sail
Alan Parsons Project, The	Tales Of Mystery And Imagination-Edgar Allan Poe	11	To One In Paradise
Amos, Tori	The Beekeeper	17	Goodbye Pisces
Anastacia	Anastacia	08	Pretty Little Dum Dum
Armstrong, Louis	All-Time Greatest Hits	12	La Vie En Rose
Arrested Development	3 Years 5 Months And 2 Days In The Life Of	13	Dawn Of The Dreads
Ashanti	Ashanti	16	Dreams
BAP	Für Usszeschnigge	06	Frau ich freu mich
Bach, Johann Sebastian	Italienisches Konzert etc - Alfred Brendel	08	Fantasia And Fugue in A minor BWV 904
Bach, Johann Sebastian	Organ Works	14	Kommst du nun Jesu vom Himmel herunter BWV 650
Baker, Chet	Jazz Masters 32	05	How Deep Is The Ocean
Barclay James Harvest	Best Of Barclay James Harvest	11	Welcome To The Show
Basie, Count	Portrait	13	Wild Bill Boogie
Beastie Boys	Licensed To Ill	09	Paul Revere
Beethoven, Ludwig van	Piano Sonatas - Maria-Joao Pires	08	Sonata No.17 in D minor Op. 31 No.2 The Tempest-Adagio
Benoit, David	Urban Daydreams	05	Snow Dancing
Berlioz, Hector	Symphonie Fantastique - Orchester de RTL	03	Symphonie Fantastique 3. Scene aux champs
Boney M	The Magic Of Boney M	08	Painter Man
Braxton, Toni	Toni Braxton	01	Another Sad Love Song
Brecker Brothers, The	Out Of The Loop	04	Secret Heart
Brecker, Michael	Tales From The Hudson	04	Beau Rivage
Burgh, Chris de	Spanish Train And Other Stories	09	The Tower
Busta Rhymes	When Disaster Strikes	05	So Hardcore
Carey, Mariah	Daydream	01	Fantasy
Charles, Ray	Ray Soundtrack	08	What'd I Say Live
Chemical Brothers, The	We Are The Night	08	Burst Generator
Chicago	17	03	Hard Habit To Break
Chopin, Frederic	Horowitz Plays Chopin	10	Mazurka in e-Moll Op.41 No. 2
Chopin, Frederic	Waltzes-Vladimir Ashkenazy	19	Es-Dur Op. Posth
Coldplay	X And Y	01	Square One
Collins, Phil	Both Sides	09	There's A Place For Us
Coltrane, John	The Very Best Of John Coltrane	06	My Favorite Things

continued on next page

continued from previous page

Interpret	**Album**	**N**	**Song**
Cooke, Sam	Sam Cooke	14	Nothing Can Change This Love
Coolio	The Return Of The Gangsta	07	Make Money feat. Gangsta-Lu
Corrs, The	In Blue	15	Rebel Heart Instrumental
Cosmic Gate	Rhythm And Drums	09	Wicked
Cypress Hill	Skull And Bones	07	Highlife
Davis, Miles	Kind Of Blue	06	Flamenco Sketches Alternate Take
Depeche Mode	The Singles	18	Stripped
Destiny's Child	Destiny's Child	08	Show Me The Way
Diamond, Neil	Serenade	06	Yes I Will
Dire Straits	Love Over Gold	04	Love Over Gold
Disturbed	Ten Thousand Fists	06	I'm Alive
Dr. Dre	2001	22	The Message
Dream Theater	Images And Words	02	Another Day
Ellington, Duke With Charles Mingus And Max Roach	Money Jungle	02	Fleurette Africaine
Eminem	The Eminem Show	12	Sing For The Moment
Eurythmics	Peace	08	I Want It All
Faithless	Sunday 8pm Special Edition	05	Take The Long Way Home
Fatboy Slim	Palookaville	07	Push And Shove
Foo Fighters	One By One	13	Sister Europe Bonus Track
Foxy Brown	Chyna Doll	15	Tramp
Franklin, Aretha	Collections	03	Misty
Furtado, Nelly	Loose	03	Do It
Gaye, Marvin	Midnight Love	05	Turn On Some Music
Genesis	We Can't Dance	03	Driving The Last Spike
Glen, Marla	This Is Marla Glen	06	Control
Gnarls Barkley	St. Elsewhere	03	St. Elsewhere
Grandmaster Mele-Mel And Scorpio	Right Now	10	When You Lose A Child
Grönemeyer, Herbert	4630 Bochum	02	Maenner
Groove Armada	LoveBox	08	Tuning In Rewritten
Haendel, Georg Friedrich	Organ Concertos Op. 4 - Rudolph Ewerhart	12	Opus 4 No.3-Allegro Gavotte
Hancock, Herbie - Brecker, Michael - Hargrove, Roy	Directions In Music	01	The Sorcerer
Haydn, Joseph	Piano Sonatas - Carmen Piazzini	07	Sonata No.9 in C major Hob. XVI7-III. Finale. Allegro
In Flames	Clayman	08	Brush The Dust Away
Irish Music	Green Green Grass	03	The Foggy Dew
Jacques Loussier Trio	Vivaldi - The Four Seasons	05	Summer-Concerto No. 2 in G Minor-Adagio
Jarre, Jean-Michel	Images	08	London Kid
Joel, Billy	2000 Years The Millennium Concert	02	Big Shot
John, Elton	Made In England	08	Please
Jones, Norah	Feels Like Home	03	Those Sweet Words
Kruder And Dorfmeister	The K And D Sessions	09	Rainer Trueby Trio Donaueschingen Peter Kruders Donaudampfschiff-fahrtsgesellschaftskapitänskajütenremix

continued on next page

continued from previous page

Interpret	**Album**	**N**	**Song**
Madonna	Confessions On A Dance Floor	11	Push
Madsen	Goodbye Logik	01	Du Schreibst Geschichte
Mann, Herbie	Just Wailin'	04	Gospel Truth
Massive Attack	Blue Lines	08	Lately
Mendelssohn	Symph No 4 Italian-Symph No 8	06	Mendelssohn Symphony No. 4
And Schubert	Unfinished-Giuseppe Sinopoli		Italian-IV. Saltarello Presto
Miller, Glenn	Portrait	07	In The Mood
Mozart,	Frühe Salzburger Meistersinfonien -	06	Sinfonie A-Dur KV2012 Andante
Wolfgang Amadeus	Kölner Kammerorchester		
Muse	Showbiz	05	Cave
Mussorgsky And Ravel	Bilder einer Ausstellung -	03	Ravel Rapsodie Espagnole -
	Bolero - Berliner Philarmoniker		2. Malaguena
Nightwish	Century Child	08	Feel For You
Nils Landgren Funk Unit	Fonk Da World	13	Calvados
Nirvana	Nevermind	08	Drain You
Orff, Carl	Carmina Burana	17	Amor volat undique
Outkast	Stankonia	02	Gasoline Dreams
Parker, Charlie	Portrait	11	An Oscar For Treadwell
Prodigy	The Fat Of The Land	02	Breathe
Queen	Greatest Hits	10	Somebody To Love
Rihanna	Music Of The Sun	04	You Don't Love Me No No No
Rollins, Sonny	Portrait	16	On A Slow Boat To China
Ross, Diana	Blue	05	Smile
Roxette	Room Service	08	Make My Head Go Pop
Schumann, Robert	Concert Pieces With Orchestra -	08	Symphony No.1 op 38
	Sinfonieorchester des Südwestfunks		Spring-III. Scherzo
Scooter	Back To The Heavyweight Jam	11	Kashmir
Sex Pistols	Never Mind The Bollocks	02	Bodies
Sibelius, Jean	Symphonien Nos. 5 And 6	07	Symphonie No.6 d-moll op.104-2.
			Allegretto moderato
Smetana, Bedrich	The Moldau-Wiener Philarmoniker	04	The Bartered Bride-Ouvertuere
Smolski, Victor	Majesty And Passion	16	Longing Dedicated to My Family
Snoop Doggy Dogg	Doggystyle	11	For All My Niggaz And Bitches
Snow Patrol	Eyes Open	09	Headlights On Dark Roads
Soulfly	Conquer	06	Rough
Steely Dan	Pretzel Logic	10	Charlie Freak
Stern, Leni	Like One	05	Lights Out
Stewart, Al	Year Of The Cat	10	On The Border Live Bonus Track
Sylver	Chances	05	In Your Eyes
Therion	Secret Of The Runes	13	Summernight City Bonus Track
Timbaland	Presents Shock Value	14	Time
Toto	The Seventh One	03	Anna
Usher	Confessions	15	Do It To Me
Van, Dyk Paul	Zurdo	10	Animacion
Wayne, Jeff	War Of The Worlds	05	Thunder Child
Wir sind Helden	Von hier an blind	10	Gekommen um zu bleiben
X-Cutioners	X-Pressions	05	Raidas Theme
Xzibit	Weapons Of Mass Destruction	13	Tough Guy

Bibliography

[1] P. Ahrendt. *Music Genre Classification Systems – A Computational Approach*. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, 2006. 12, 38, 44

[2] P. Ahrendt, C. Goutte, and J. Larsen. Co-occurrence models in music genre classification. In V. Calhoun, T. Adali, J. Larsen, D. Miller, and S. Douglas, editors, *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 247–252, 2005. 38

[3] E. Alexandre, L. Cuadra, M. Rosa, and F. López-Ferreras. Feature selection for sound classification in hearing aids through restricted search driven by genetic algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(8):2249–2256, 2007. 69, 70

[4] E. Alpaydin. *Introduction to Machine Learning*. The MIT Press, Cambridge London, 2010. 37, 44, 82, 85

[5] E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1-2):237–260, 1998. 60

[6] W. Apel. *The Notation of Polyphonic Music, 900–1600*. Mediaeval Academy of America, Cambridge, 1961. 14, 15

[7] A. F. Arabi and G. Lu. Enhanced polyphonic music genre classification using high level features. In *Proceedings of the IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 101–106, 2009. 22, 23

[8] J.-J. Aucouturier and F. Pachet. Music similarity measures: What's the use? In *Proceedings of the 3rd International Symposium on Music Information Retrieval (ISMIR)*, pages 157–163, 2002. 38

[9] T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996. 58

[10] J. Bader, K. Deb, and E. Zitzler. Faster hypervolume-based search using Monte Carlo sampling. In M. Ehrgott, B. Naujoks, T. J. Stewart, and J. Wallenius, editors, *Proceedings of the 19th International Conference on Multiple Criteria Decision Making (MCDM), 2008*, volume 634 of *Lecture Notes in Economics and Mathematical Systems*, pages 313–326. Springer Berlin Heidelberg, 2010. 63

[11] A. Berenzweig, D. P. W. Ellis, and S. Lawrence. Anchor space for classification and similarity measurement of music. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 29–32. IEEE, 2003. 13, 64

[12] A. Berenzweig, B. Logan, D. P. W. Ellis, and B. P. W. Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 28(2):63–76, 2004. 12, 45

[13] T. Bertin-Mahieux, D. Eck, F. Maillet, and P. Lamere. Autotagger: A model for predicting social tags from acoustic features on large music databases. *Journal of New Music Research*, 37(2):115–135, 2008. 17

[14] B. Bischl, O. Mersmann, H. Trautmann, and C. Weihs. Resampling methods for meta-model validation with recommendations for evolutionary computation. *Evolutionary Computation*, 20(2):249–275, 2012. 80, 82

[15] B. Bischl, I. Vatolkin, and M. Preuß. Selecting small audio feature sets in music classification by means of asymmetric mutation. In R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, editors, *Proceedings of the 11th International Conference on Parallel Problem Solving From Nature (PPSN)*, volume 6238 of *Lecture Notes in Computer Science*, pages 314–323. Springer, 2010. 9, 42, 43, 70, 91, 129, 133

[16] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006. 38, 44

[17] H. Blume, B. Bischl, M. Botteck, C. Igel, R. Martin, G. Rötter, G. Rudolph, W. Theimer, I. Vatolkin, and C. Weihs. Huge music archives on mobile devices. *IEEE Signal Processing Magazine*, 28(4):24–39, 2011. 10, 13

[18] H. Blume, M. Haller, M. Botteck, and W. Theimer. Perceptual feature based music classification - a DSP perspective for a new type of application. In W. A. Najjar and H. Blume, editors, *Proceedings of the 8th International Conference on Systems, Architectures, Modeling and Simulation (IC-SAMOS)*, pages 92–99. IEEE, 2008. 68, 77

[19] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. 48

[20] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting*. Springer, 2002. 38

[21] J. C. Brown, O. Houix, and S. McAdams. Feature dependence in the automatic identification of musical woodwind instruments. *Journal of the Acoustical Soceity of America*, 109(3):1064–1072, 2001. 38

[22] I. Bruha. Pre- and post-processing in machine learning and data mining. In *Machine Learning and its Applications: Advanced Lectures*, pages 258–266. Springer, London, 2001. 36

[23] J. J. Burred and A. Lerch. A hierarchical approach to automatic musical genre classification. In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx)*, pages 8–11, 2003. 68

[24] M. Caetano and X. Rodet. Independent manipulation of high-level spectral envelope shape features for sound morphing by means of evolutionary computation. In *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx)*, 2010. 21

[25] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008. 11, 16

[26] Ò. Celma. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play In the Digital Music Space*. Springer, 2010. 12, 79

[27] Ò. Celma and X. Serra. FOAFing the music: Bridging the semantic gap in music recommendation. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):250–256, 2008. 22

[28] W. Chai. *Automated Analysis of Musical Structure*. PhD thesis, School of Architecture and Planning, Massachusetts Institute Of Technology, 2005. 95

[29] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. The MIT Press, Cambridge, 2010. 44

[30] M. Chmulík, R. Jarina, and M. Kuba. On objective feature selection for affective sounds discrimination. In *Proceedings of the 54th International Symposium on Electronics in Marine (ELMAR)*, pages 199–202, 2012. 69, 70

[31] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, 2007. 60

[32] N. Collins. Computational analysis of musical influence: A musicological case study using MIR tools. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 177–182, 2010. 13

[33] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. 27

[34] D. Cope. *Experiments in Musical Intelligence*. A-R Editions, Inc., Madison, 1996. 11

[35] D. Cope. *Computer Models of Musical Creativity*. The MIT Press, Cambridge, 2005. 12, 13

[36] G. W. Corder and D. I. Foreman. *Nonparametric Statistics for Non-Statisticians*. Wiley, New Jersey, 2009. 82, 83, 84

[37] F. G. Cozman, I. Cohen, and M. C. Cirelo. Semi-supervised learning of mixture models. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 99–106, 2003. 44

[38] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, Chicester, 2001. 59, 60

[39] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. M. Guervós, and H.-P. Schwefel, editors, *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature (PPSN)*, volume 1917 of *Lecture Notes in Computer Science*, pages 849–858, Berlin, Heidelberg, 2000. Springer. 62

[40] T. Deinert, I. Vatolkin, and G. Rudolph. Regression-based tempo recognition from chroma and energy accents for slow audio recordings. In *Proceedings of the AES*

42nd International Conference on Semantic Audio (AES), pages 60–68, 2011. 10, 79, 123

[41] E. Diner. Music signal-based classification methods for personal categories. Master's thesis, Institute for Neural Computation, Ruhr-Universität Bochum, 2008. 38

[42] C. Dittmar, K. F. Hildebrand, D. Gaertner, M. Winges, F. Müller, and P. Aichroth. Audio forensics meets music information retrieval – a toolbox for inspection of music plagiarism. In *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1249–1253, 2012. 12

[43] M. Dorigo and T. Stützle. *Ant Colony Optimization*. The MIT Press, Cambridge, 2004. 58

[44] J. S. Downie. Music information retrieval. *Annual Review of Information Science and Technology*, 37(1):295–340, 2003. 11, 13

[45] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, New York, 2001. 44, 46, 82

[46] R. C. Eberhart, Y. Shi, and J. Kennedy. *Swarm intelligence*. Morgan Kaufmann, San Francisco, 2001. 58

[47] D. Eck, T. Bertin-Mahieux, and P. Lamere. Autotagging music using supervised machine learning. In S. Dixon, D. Bainbridge, and R. Typke, editors, *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 367–368. Austrian Computer Society, 2007. 23

[48] T. Eerola and R. Ferrer. Instrument library (MUMS) revised. *Music Perception*, 25(3):253–255, 2008. 87

[49] B. Efron and R. Tibshirani. Improvements on cross-validation: The 0.632 + bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997. 80

[50] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer, 2007. 58

[51] M. Eichhoff and C. Weihs. Musical instrument recognition by high-level features. In W. Gaul, A. Geyer-Schulz, L. Schmidt-Thieme, and J. Kunze, editors, *Proceedings of the 34th Annual Conference of the Gesellschaft für Klassifikation, 2010*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 373–381. Springer, 2012. 21, 68, 88, 135, 136

[52] C. Emmanouilidis. *Evolutionary Multi-Objective Feature Selection and its Application to Industrial Machinery Fault Diagnosis*. PhD thesis, School of Computing, Engineering and Technology, University of Sunderland, 2001. 67

[53] C. Emmanouilidis, A. Hunter, and J. MacIntyre. A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, volume 1, pages 309–316, 2000. 63, 66, 67

[54] M. Emmerich, N. Beume, and B. Naujoks. An EMO algorithm using the hypervolume measure as selection criterion. In C. A. Coello Coello, A. H. Aguirre, and E. Zitzler, editors, *Proceedings of the 3rd Conference on Evolutionary Multi-Criterion Op-*

timization (EMO), volume 3410 of *Lecture Notes in Computer Science*, pages 62–76, 2005. 62

[55] A. Eronen. *Signal Processing Methods for Audio Classification and Music Content Analysis*. PhD thesis, Department of Signal Processing, Tampere University of Technology, 2009. 40

[56] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi. Audio-based context recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):321–329, 2006. 17, 39

[57] S. Essid, G. Richard, and B. David. Musical instrument recognition by pairwise classification strategies. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1401–1412, 2006. 37, 68, 69, 70

[58] S. Ewert, M. Müller, and R. B. Dannenberg. Towards reliable partial music alignments using multiple synchronization strategies. In M. Detyniecki, A. García-Serrano, and A. Nürnberger, editors, *Proceedings of the 7th International Workshop on Adaptive Multimedia Retrieval (AMR), 2009*, volume 6535 of *Lecture Notes in Computer Science*, pages 35–48, 2011. 13

[59] F. Fernández, F. Chavez, R. Alcalá, and F. Herrera. Musical genre classification by means of fuzzy rule-based systems: A preliminary approach. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 2571–2577. IEEE, 2011. 131

[60] R. Fiebrink and I. Fujinaga. Feature selection pitfalls and music classification. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pages 340–341, 2006. 68, 69, 81

[61] R. Fiebrink, C. McKay, and I. Fujinaga. Combining D2K and JGAP for efficient feature weighting for classification tasks in music information retrieval. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 510–513, 2005. 69

[62] H. Fletcher and W. A. Munson. Loudness, its definition, measurement and calculation. *Journal of the Acoustical Society of America*, 5(2):82–108, 1933. 25

[63] A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer. Playlist generation using start and end songs. In J. P. Bello, E. Chew, and D. Turnbull, editors, *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, pages 173–178, 2008. 13

[64] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, 1966. 57

[65] C. B. Fowler. The museum of music: A history of mechanical instruments. *Music Educators Journal*, 54(2):45–49, 1967. 14

[66] I. Fujinaga. Exemplar-based learning in adaptive optical music recognition system. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 55–56, 1996. 66, 69

[67] I. Fujinaga. Machine recognition of timbre using steady-state tone of acoustic musical instruments. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 207–210, 1998. 69

[68] T. Fujishima. Realtime chord recognition of musical sound: a system using common lisp music. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 464–467, 1999. 27

[69] A. Gaspar-Cunha, F. Mendes, J. Duarte, A. Vieira, B. Ribeiro, A. Ribeiro, and J. C. das Neves. Multi-objective evolutionary algorithms for feature selection: Application in bankruptcy prediction. In K. Deb, A. Bhattacharya, N. Chakraborti, P. Chakroborty, S. Das, J. Dutta, S. K. Gupta, A. Jain, V. Aggarwal, J. Branke, S. J. Louis, and K. C. Tan, editors, *Proceedings of the 8th International Conference on Simulated Evolution and Learning (SEAL)*, volume 6457 of *Lecture Notes in Computer Science*, pages 319–328, Berlin Heidelberg, 2010. Springer. 68

[70] R. O. Gjerdingen and D. Perrott. Scanning the dial: The rapid recognition of music genres. *Journal of New Music Research*, 37(2):93–100, 2008. 23, 43, 64

[71] E. Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, 2006. 28

[72] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Music genre database and musical instrument sound database. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, pages 229–230, 2003. 87

[73] M. Grimaldi, P. Cunningham, and A. Kokaram. An evaluation of alternative feature selection strategies and ensemble techniques for classifying music. In *Proceedings of the Workshop on Multimedia Discovery and Mining held within the 14th European Conference on Machine Learning (ECML)/the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, 2003. 68

[74] C. Guerra-Salcedo, S. Chen, D. Whitley, and S. Smith. Fast and accurate feature selection using hybrid genetic strategies. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 177–184, 1999. 67

[75] I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, editors. *Feature Extraction. Foundations and Applications*, volume 207 of *Studies in Fuzziness and Soft Computing*. Springer, Berlin Heidelberg, 2006. 55, 60, 61

[76] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11:10–18, 2009. 130

[77] V. Hamacher, J. Chalupper, J. Eggers, E. Fischer, U. Kornagel, H. Puder, and U. Rass. Signal processing in high-end hearing aids: state of the art, challenges, and future trends. *EURASIP Journal on Advances in Signal Processing*, 2005(18):2915–2929, 2005. 13

[78] T. M. Hamdani, J.-M. Won, A. M. Alimi, and F. Karray. Multi-objective feature selection with NSGA II. In B. Beliczynski, A. Dzielinski, M. Iwanowski, and B. Ribeiro, editors, *Proceedings of the 8th international conference on Adaptive and*

Natural Computing Algorithms (ICANNGA), Part I, volume 4431 of *Lecture Notes in Computer Science*, pages 240–247. Springer, Berlin Heidelberg, 2007. 68

[79] Y. Han and C. Raphael. Informed source separation of orchestra and soloist. In J. S. Downie and R. C. Veltkamp, editors, *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 315–320, 2010. 13

[80] J. Handl and J. Knowles. Feature subset selection in unsupervised learning via multiobjective optimization. *International Journal of Computational Intelligence Research*, 2(3):217–238, 2006. 68

[81] W. M. Hartmann. Pitch, periodicity, and auditory organization. *Journal of the Acoustical Society of America*, 100(6):3491–3502, 1996. 24

[82] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2009. 45, 48, 54, 132

[83] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975. 57

[84] M. Hollander and D. A. Wolfe. *Nonparametric Statistical Methods*. Wiley-Interscience, New York, 1999. 82, 84, 120

[85] H. Homburg, I. Mierswa, B. Möller, K. Morik, and M. Wurst. A benchmark dataset for audio classification and clustering. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 528–531, 2005. 93

[86] T. S. Huang, C. K. Dagli, S. Rajaram, E. Y. Chang, M. I. Mandel, G. E. Poliner, and D. P. W. Ellis. Active learning for interactive multimedia retrieval. *Proceedings of the IEEE*, 96(4):648–667, 2008. 78

[87] K. A. Schouhamer Immink. The CD story. *Journal of the Audio Engineering Society*, 46:458–465, 1998. 26

[88] D. C. Ince, editor. *Mechanical Intelligence: Collected Works of A. M. Turing*. North-Holland Publishing Co., Amsterdam, 1992. 58

[89] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 2419–2426, 2008. 63

[90] Ö. Izmirli. Audio key finding using low-dimensional spaces. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pages 127–132, 2006. 11

[91] M. Jelasity, M. Preuß, and A. E. Eiben. Operator learning for a problem class in a distributed peer-to-peer environment. In J. J. Merelo Guervós, P. Adamidis, H.-G. Beyer, J. L. Fernández-Villacañas Martín, and H.-P. Schwefel, editors, *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature (PPSN)*, volume 2439 of *Lecture Notes in Computer Science*, pages 172–183. Springer, 2002. 63

[92] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2002. 37

[93] A. Jordan, D. Scheftelowitsch, J. Lahni, J. Hartwecker, M. Kuchem, M. Walter-Huber, N. Vortmeier, T. Delbrügger, Ü. Güler, I. Vatolkin, and M. Preuß. BeatTheBeat: Music-based procedural content generation in a mobile game. In *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG)*, pages 320–327, 2012. 10, 13

[94] J. Joyce. Pandora and the music genome project. *Scientific Computing*, 23(10):40–41, 2006. 17

[95] H. Kameoka, T. Nishimoto, and S. Sagayama. A multipitch analyzer based on harmonic temporal structured clustering. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):982–994, 2007. 44

[96] G. V. Karkavitsas and G. A. Tsihrintzis. Automatic music genre classification using hybrid genetic algorithms. In G. A. Tsihrintzis, M. Virvou, L. C. Jain, and R. J. Howlett, editors, *Proceedings of the 4th International Conference on Intelligent Interactive Multimedia Systems (IIMSS)*, volume 11 of *Smart Innovation, Systems and Technologies*, pages 323–335, Berlin Heidelberg, 2011. Springer. 69

[97] M. Kassler. Toward music information retrieval. *Perspectives of New Music*, 4:59–67, 1966. 11

[98] J. D. Kelly Jr. and L. Davis. Hybridizing the genetic algorithm and the k-nearest neighbors classification algorithm. In R. K. Belew and L. B. Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithm and their Applications (ICGA)*, pages 377–383. Morgan Kaufmann, 1991. 67

[99] Y.-S. Kim, W. N. Street, and F. Menczer. Evolutionary model selection in unsupervised learning. *Intelligent Data Analysis*, 6(6):531–556, 2002. 68

[100] A. Klapuri. Introduction to music transcription. In A. Klapuri and M. Davy, editors, *Signal Processing Methods for Music Transcription*, pages 3–20. Springer, New York, 2006. 24

[101] A. Klapuri and M. Davy, editors. *Signal Processing Methods for Music Transcription*. Springer, New York, 2006. 12, 13, 27

[102] P. Knees, M. Schedl, T. Pohle, and G. Widmer. Exploring music collections in virtual landscapes. *IEEE Multimedia*, 14(3):46–54, 2007. 44

[103] T. A. Knight. Analysis of trumpet tone quality using machine learning and audio feature selection. Master's thesis, Department of Electrical and Computer Engineering, McGill University, 2012. 69, 70

[104] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997. 53, 60, 61

[105] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, Cambridge, 1992. 58

[106] O. Kramer and T. Hein. Stochastic feature selection in support vector machine based instrument recognition. In B. Mertsching, M. Hund, and Z. Aziz, editors, *Proceedings of the 32nd Annual German Conference on Advances in Artificial Intelligence (KI)*, volume 5803 of *Lecture Notes in Computer Science*, pages 727–734, Berlin Heidelberg, 2009. Springer. 69, 70

[107] N. Krasnogor. Memetic algorithms. In G. Rozenberg, T. Bäck, and J. N. Kok, editors, *Handbook of Natural Computing, Volume 2*, pages 905–936. Springer, Berlin Heidelberg, 2012. 58

[108] P. Krishnamoorthy and S. Kumar. Hierarchical audio content classification system using an optimal feature selection algorithm. *Multimedia Tools and Applications*, 54(2):415–444, 2011. 68

[109] C. L. Krumhansl. Plink: "thin slices" of music. *Music Perception: An Interdisciplinary Journal*, 27(5):337–354, 2010. 23, 43

[110] M. Kudo and J. Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33(1):25–41, 2000. 60, 67, 68, 70

[111] T. N. Lal, O. Chapelle, J. Weston, and A. Elisseeff. Embedded methods. In I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, editors, *Feature Extraction. Foundations and Applications*, volume 207 of *Studies in Fuzziness and Soft Computing*, pages 137–165. Springer, Berlin Heidelberg, 2006. 55

[112] T. Lambrou, P. Kudumakis, R. Speller, M. Sandler, and A. Linney. Classification of audio signals using statistical features on time and wavelet transform domains. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3621–3624, 1998. 38

[113] T. Lange, M. L. Braun, V. Roth, and J. M. Buhmann. Stability-based model selection. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15 (NIPS)*, pages 617–624. The MIT Press, 2002. 78

[114] P. Langley. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*, pages 140–144. AAAI Press, 1994. 55

[115] O. Lartillot. *MIRtoolbox 1.4 User's Manual*. Finnish Centre of Excellence in Interdisciplinary Music Research and Swiss Center for Affective Sciences, 2012. Online resource. 30, 135, 136, 137, 143, 144

[116] O. Lartillot, T. Eerola, P. Toiviainen, and J. Fornari. Multi-feature modeling of pulse clarity: Design, validation, and optimization. In J. P. Bello, E. Chew, and D. Turnbull, editors, *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, pages 521–526, 2008. 33

[117] O. Lartillot and P. Toiviainen. MIR in Matlab (II): A toolbox for musical feature extraction from audio. In S. Dixon, D. Bainbridge, and R. Typke, editors, *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 127–130, 2007. 30, 39, 130

[118] M. Laumanns, G. Rudolph, and H.-P. Schwefel. A spatial predator-prey approach to multi-objective optimization: A preliminary study. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature (PPSN)*, volume 1498 of *Lecture Notes in Computer Science*, pages 241–249, 1998. 58

[119] T. Li and M. Ogihara. Music artist style identification by semi-supervised learning from both lyrics and content. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, pages 364–367, New York, 2004. ACM Press. 44

[120] T. Li, M. Ogihara, and G. Tzanetakis, editors. *Music Data Mining*. CRC Press/Taylor & Francis, Boca Raton, 2012. 12, 17, 36, 44

[121] T. Lidy, A. Rauber, A. Pertusa, and J. M. Iñesta. Improving genre classification by combination of audio and symbolic descriptors using a transcription system. In S. Dixon, D. Bainbridge, and R. Typke, editors, *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 61–66. Austrian Computer Society, 2007. 23

[122] J. Liu and X. Hu. User-centered music information retrieval evaluation. In *Proceedings of the Joint Conference on Digital Libraries (JCDL) Workshop: Music Information Retrieval for the Masses*, 2010. 79

[123] A. Livshin and X. Rodet. The significance of the non-harmonic "noise" versus the harmonic series for musical instrument recognition. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pages 95–100, 2006. 42, 68

[124] J. Loughrey and P. Cunningham. Overfitting in wrapper-based feature subset selection: The harder you try the worse it gets. In M. Bramer, F. Coenen, and T. Allen, editors, *Proceedings of AI-2004, the 24th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 33–43. Springer, 2004. 67, 80, 91, 132

[125] H. Lukashevich. Towards quantitative measures of evaluating song segmentation. In J. P. Bello, E. Chew, and D. Turnbull, editors, *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, pages 375–380, 2008. 79

[126] J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580, 1975. 38

[127] T. Mäkinen, S. Kiranyaz, J. Pulkkinen, and M. Gabbouj. Evolutionary feature generation for content-based audio classification and retrieval. In *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1474–1478, 2012. 37, 70

[128] M. I. Mandel and D. Ellis. Song-level features and support vector machines for music classification. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 594–599, 2005. 38

[129] L. Manniche. *Music and Musicians in Ancient Egypt*. British Museum Press, London, 1991. 14

[130] T. Marill and D. M. Green. On the effectiveness of receptors in recognition systems. *IEEE Transactions on Information Theory*, 9(1):11–17, 1963. 55

[131] B. M. Marlin. *Missing Data Problems in Machine Learning*. PhD thesis, Department of Computer Science, University of Toronto, 2008. 36

[132] C. M. Marques, I. R. Guilherme, R. Y. M. Nakamura, and J. P. Papa. New trends in musical genre classification using optimum-path forest. In A. Klapuri and C. Leider, editors, *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 699–704. University of Miami, 2011. 69

[133] R. Martin and A. M. Nagathil. Cepstral modulation ratio regression (CMRARE) parameters for audio signal analysis and classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 321–324. IEEE, 2009. 31, 136

[134] V. Mattern, I. Vatolkin, and G. Rudolph. A case study about the effort to classify music intervals by chroma and spectrum analysis. In B. Lausen, D. van den Poel, and A. Ultsch, editors, *Proceedings of the 35th Annual Conference of the German Classification Society (GfKl), 2011*, to appear. 10

[135] M. Mauch and S. Dixon. Approximate note transcription for the improved identification of difficult chords. In J. S. Downie and R. C. Veltkamp, editors, *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 135–140, 2010. 30, 31, 32, 130, 136, 137, 143

[136] M. Mauch and M. Levy. Structural change on multiple time scales as a correlate of musical complexity. In A. Klapuri and C. Leider, editors, *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 489–494. University of Miami, 2011. 41

[137] R. Mayer, A. Rauber, P. J. Ponce de León, C. Pérez-Sancho, and J. M. Iñesta. Feature selection in a cartesian ensemble of feature subspace classifiers for music categorisation. In *Proceedings of the 3rd International Workshop on Machine Learning and Music (MML)*, pages 53–56. ACM, 2010. 68

[138] C. McKay. *Automatic Music Classification with jMIR*. PhD thesis, Department of Music Research, Schulich School of Music, McGill University, 2010. 21, 23, 24, 30, 64, 130

[139] C. McKay and I. Fujinaga. Musical genre classification: Is it worth pursuing and how can it be improved? In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pages 101–106, 2006. 131

[140] M. F. McKinney, D. Moelants, M. E. P. Davies, and A. Klapuri. Evaluation of audio beat tracking and music tempo extraction algorithms. *Journal of New Music Research*, 36(1):1–16, 2007. 79

[141] A. Meng, P. Ahrendt, J. Larsen, and L. K. Hansen. Temporal feature integration for music genre classification. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1654–1664, 2007. 38

[142] C. Meyer. Kreuzverhörtest. Der C't-Leser-Hörtest: MP3 gegen CD. *C't*, 6:92–94, 2000. 16

[143] S. Meyer-Nieberg and H.-G. Beyer. Self-adaptation in evolutionary algorithms. In F. G. Lobo, C. F. Lima, and Z. Michalewicz, editors, *Parameter Setting in Evolutionary Algorithms*, volume 54 of *Studies in Computational Intelligence*, pages 47–75. Springer, Berlin Heidelberg, 2007. 58

[144] U. Michels. *dtv-Atlas Musik: Band 1: Systematischer Teil. Musikgeschichte von den Anfängen bis zur Renaissance*. Deutscher Taschenbuch Verlag, München, 1977. 31

[145] I. Mierswa. Controlling overfitting with multi-objective support vector machines. In H. Lipson, editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 1830–1837. ACM, 2007. 78

[146] I. Mierswa and K. Morik. Automatic feature extraction for classifying audio data. *Machine Learning Journal*, 58(2-3):127–149, 2005. 18, 28, 29, 31, 37, 46, 69, 70, 101

[147] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. YALE: Rapid prototyping for complex data mining tasks. In T. Eliassi-Rad, L. H. Ungar, M. Craven, and D. Gunopulos, editors, *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 935–940. ACM, 2006. 30, 89, 130

[148] G. Milner. *Perfecting Sound Forever: The Story of Recorded Music*. Granta Books, London, 2010. 14, 16

[149] E. R. Miranda and J. A. Biles. *Evolutionary Computer Music*. Springer, New York, 2007. 13

[150] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Singapore, 1997. 44, 80, 85

[151] B. C. J. Moore and B. R. Glasberg. A revision of Zwicker's loudness model. *Acta Acustica united with Acustica*, 82(2):335–345, 1996. 31

[152] F. Mörchen. *Time Series Knowledge Mining*. PhD thesis, Department of Mathematics and Computer Science, University of Marburg, 2006. 38

[153] F. Mörchen, A. Ultsch, M. Thies, and I. Löhken. Modeling timbre distance with temporal statistics from polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):81–90, 2006. 13, 21, 38, 44, 70

[154] M. Müller. *Information Retrieval for Music and Motion*. Springer, Berlin Heidelberg, 2007. 12, 137

[155] M. Müller and S. Ewert. Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In A. Klapuri and C. Leider, editors, *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, pages 215–220. University of Miami, 2011. 29, 30, 32, 130, 137

[156] R. Munkong and B.-H. Juang. Auditory perception and cognition. *IEEE Signal Processing Magazine*, 25(3):98–117, 2008. 24

[157] A. Nagathil, I. Vatolkin, and W. Theimer. Comparison of partition-based audio features for music classification. In *Proceedings the 9th ITG Fachtagung Sprachkommunikation*, 2010. 9, 46, 129

[158] S. Nayak and A. Bhutani. Music genre classification using GA-induced minimal feature-set. In *Proceedings of the 3rd National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pages 33–36, 2011. 69, 70

[159] M. Niitsuma, H. Takaesu, H. Demachi, M. Oono, and H. Saito. Development of an automatic music selection system based on runner's step frequency. In J. P. Bello, E. Chew, and D. Turnbull, editors, *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, pages 193–198, 2008. 17

[160] I.-S. Oh, J.-S. Lee, and B.-R. Moon. Local search-embedded genetic algorithms for feature selection. In *Proceedings of the 16th International Conference on Pattern Recognition (ICPR)*, volume 2, pages 148–151, 2002. 67

[161] I.-S. Oh, J.-S. Lee, and B.-R. Moon. Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1424–1437, 2004. 67

[162] J. Olajec, C. Erkut, and R. Jarina. GA-based feature selection for synchronous and asynchronous applause detection. In *Proceedings of the Finnish Signal Processing Symposium (FINSIG)*, 2007. 69, 70

[163] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. A methodology for feature selection using multiobjective genetic algorithms for handwritten digit string recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(6):903–929, 2003. 67

[164] N. Orio. Music retrieval: A tutorial and review. *Foundations and Trends in Information Retrieval*, 1(1):1–90, 2006. 11

[165] F. Pachet and D. Cazaly. A taxonomy of musical genres. In J.-J. Mariani and D. Harman, editors, *Proceedings of the 6th International Conference on Computer-Assisted Information Retrieval (Recherche d'Information et ses Applications, RIAO)*, pages 1238–1245, 2000. 12, 118

[166] F. Pachet and A. Zils. Evolving automatically high-level music descriptors from acoustic signals. In *Proceedings of the 1st International Symposium on Computer Music Modeling and Retrieval (CMMR)*, volume 2771 of *Lecture Notes in Computer Science*, pages 42–53. Springer, 2003. 70

[167] E. Pampalk. *Computational Models of Music Similarity and their Application in Music Information Retrieval*. PhD thesis, Department of Computer Science, Vienna University of Technology, 2006. 33

[168] T. H. Park. *Introduction to Digital Signal Processing: Computer Musically Speaking*. World Scientific, Singapore, 2010. 40

[169] J. Paulus and A. Klapuri. Music structure analysis using a probabilistic fitness measure and an integrated musicological model. In J. P. Bello, E. Chew, and D. Turnbull, editors, *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, pages 369–374, 2008. 33, 41, 42

[170] J. Paulus, M. Müller, and A. Klapuri. Audio-based music structure analysis. In J. S. Downie and R. C. Veltkamp, editors, *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 625–636, 2010. 41

[171] G. Peeters. Chroma-based estimation of musical key from audio-signal analysis. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pages 115–120, 2006. 27

[172] C. J. Plack, R. R. Fay, A. J. Oxenham, and A. N. Popper, editors. *Pitch: Neural Coding and Perception*. Springer, New York, 2005. 24

[173] T. Pohle, P. Knees, K. Seyerlehner, and G. Widmer. A high-level audio feature for music retrieval and sorting. In *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx)*, 2010. 21

[174] S. T. Pope, F. Holm, and A. Kouznetsov. Feature extraction and database design for music software. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 596–603, 2004. 21

[175] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(10):1119–1125, 1994. 55

[176] W. F. Punch, E. D. Goodman, M. Pei, L. Chia-Shun, P. D. Hovland, and R. J. Enbody. Further research on feature selection and classification using genetic algorithms. In S. Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms (ICGA)*, pages 557–564. Morgan Kaufmann, 1993. 67, 69

[177] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, 1993. 47, 48

[178] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Upper Saddle River, 1993. 27

[179] C. C. O. Ramos, J. P. Papa, A. N. Souza, G. Chiachia, and A. X. Falcao. What is the importance of selecting features for non-technical losses identification? In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1045–1048, 2011. 69

[180] R. B. Rao, G. Fung, and R. Rosales. On the dangers of cross-validation. An experimental evaluation. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pages 588–596. SIAM, 2008. 81

[181] Z. W. Raś and A. A. Wieczorkowska. *Advances in Music Information Retrieval*, volume 274 of *Studies in Computational Intelligence*. Springer, Berlin Heidelberg, 2010. 12

[182] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. In W. L. Buntine, M. Grobelnik, D. Mladenic, and J. Shawe-Taylor, editors, *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), Part II*, volume 5782 of *Lecture Notes in Computer Science*, pages 254–269, Berlin Heidelberg, 2009. Springer. 65

[183] J. Reunanen. Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3:1371–1382, 2003. 81

[184] J. Reunanen. Search strategies. In I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, editors, *Feature Extraction. Foundations and Applications*, volume 207 of *Studies in Fuzziness and Soft Computing*, pages 119–136. Springer, Berlin Heidelberg, 2006. 55

[185] T. D. Rossing, F. R. Moore, and P. A. Wheeler. *The Science of Sound*. Addison Wesley, San Francisco, 2002. 24, 25, 31

[186] G. Rötter, I. Vatolkin, and C. Weihs. Computational prediction of high-level descriptors of music personal categories. In B. Lausen, D. van den Poel, and A. Ultsch, editors, *Proceedings of the 35th Annual Conference of the German Classification Society (GfKl), 2011*, to appear. 10, 24, 87, 95, 97, 129

[187] G. Rozenberg, T. Bäck, and J. N. Kok, editors. *Handbook of Natural Computing*. Springer, Berlin Heidelberg, 2012. 58

[188] H. Rump, S. Miyabe, E. Tsunoo, N. Ono, and S. Sagayama. Autoregressive MFCC models for genre classification improved by harmonic-percussion separation. In J. S. Downie and R. C. Veltkamp, editors, *Proceedings of the 11th International Society on Music Information Retrieval Conference (ISMIR)*, pages 87–92, 2010. 38

[189] P. Saari, T. Eerola, and O. Lartillot. Generalizability and simplicity as criteria in feature selection: Application to mood classification in music. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1802–1812, 2011. 68, 69

[190] C. Sachs. *Rhythm and Tempo: A Study in Music History*. Norton, New York, 1953. 33

[191] G. Schuller, M. Gruhne, and T. Friedrich. Fast audio feature extraction from compressed audio data. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1262–1271, 2011. 22

[192] H.-P. Schwefel. Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik. Master's thesis, Hermann Föttinger-Institut für Strömungsmechanik, TU Berlin, 1964. 57

[193] J. Seppänen, A. J. Eronen, and J. Hiipakka. Joint beat and tatum tracking from music signals. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pages 23–28, 2006. 33

[194] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948. 47

[195] W. W. Siedlecki and J. Sklansky. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10(5):335–347, 1989. 66

[196] C. N. Silla Jr., A. L. Koerich, and C. A. A. Kaestner. A feature selection approach for automatic music genre classification. *International Journal of Semantic Computing*, 3(2):183–208, 2009. 69, 70

[197] I. R. Sinclair. *Introducing Digital Audio*. PC Publishing, Tonbridge, 1992. 15

[198] M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. In A. Sattar and B. H. Kang, editors, *Proceedings of the 19th Australian Joint Conference on Artificial Intelligence (AI)*, volume 4304 of *Lecture Notes in Computer Science*, pages 1015–1021. Springer, Berlin Heidelberg, 2006. 75

[199] W. A. Stahel. *Statistische Datenanalyse*. Vieweg, Braunschweig Wiesbaden, 1995. 83

[200] B. Sturm. A survey of evaluation in music genre recognition. In *Proceedings of the 10th International Workshop on Adaptive Multimedia Retrieval (AMR)*, 2012. 12

[201] P. Tagg. Analyzing popular music: Theory, method and practice. *Popular Music*, 2:37–65, 1982. 24

[202] J. Takagi, Y. Ohishi, A. Kimura, M. Sugiyama, M. Yamada, and H. Kameoka. Automatic audio tag classification via semi-supervised canonical density estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2232–2235. IEEE, 2011. 44

[203] D. M. J. Tax. *One-Class Classification*. PhD thesis, Pattern Recognition Group, Delft University of Technology, 2001. 46

[204] D. Temperley. *Music and Probability*. The MIT Press, Cambridge, 2007. 12

[205] W. Theimer, I. Vatolkin, M. Botteck, and M. Buchmann. Content-based similarity search and visualization for personal music categories. In *Proceedings of the 6th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 9–16, 2008. 9, 12, 13, 46, 74, 81, 101, 129

[206] W. Theimer, I. Vatolkin, and A. Eronen. Definitions of audio features for music content description. Technical Report TR08-2-001, Faculty of Computer Science, Technische Universität Dortmund, 2008. 30, 135, 136, 137, 143, 144

[207] K. Torkkola. Information-theoretic methods. In I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, editors, *Feature Extraction*, volume 207 of *Studies in Fuzziness and Soft Computing*, pages 167–185. Springer, Berlin Heidelberg, 2006. 53

[208] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas. Multi-label classification of music by emotion. *EURASIP Journal on Audio, Speech, and Music Processing*, 2011(4), 2011. 45

[209] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007. 45

[210] P. D. Turney. Types of cost in inductive concept learning. In *Proceedings of the Workshop on Cost-Sensitive Learning at the 17th International Conference on Machine Learning (WCSL at ICML)*, pages 15–21, 2000. 77

[211] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002. 17, 30, 38

[212] M. Unser. Sampling – 50 years after Shannon. *Proceedings of the IEEE*, 88(4):569–587, 2000. 25

[213] E. M. v. Hornbostel and C. Sachs. Systematik der Musikinstrumente. Ein Versuch. *Zeitschrift für Ethnologie*, 46(4-5):553–590, 1914. 24

[214] I. Vatolkin. Multi-objective evaluation of music classification. In W. A. Gaul, A. Geyer-Schulz, L. Schmidt-Thieme, and J. Kunze, editors, *Proceedings of the 34th Annual Conference of the German Classification Society (GfKl), 2010*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 401–410. Springer, Berlin Heidelberg, 2012. 9, 46, 71, 72, 129

[215] I. Vatolkin, B. Bischl, G. Rudolph, and C. Weihs. Statistical comparison of classifiers for multi-objective feature selection in instrument recognition. In M. Spiliopoulou and L. Schmidt-Thieme, editors, *Proceedings of the 36th Annual Conference of the German Classification Society (GfKl), 2012*, to appear. 10, 90, 112, 129

[216] I. Vatolkin, A. Nagathil, W. Theimer, and R. Martin. Performance of specific vs. generic feature sets in polyphonic music instrument recognition. In R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, editors, *Proceedings of the 7th International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, volume 7811 of *Lecture Notes in Computer Science*, pages 587–599, 2013. 7, 10, 40, 42, 54, 70, 129

[217] I. Vatolkin, M. Preuß, and G. Rudolph. Multi-objective feature selection in music genre and style recognition tasks. In N. Krasnogor and P. L. Lanzi, editors, *Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference (GECCO)*, pages 411–418. ACM Press, 2011. 7, 9, 42, 43, 46, 54, 63, 66, 70, 72, 76, 89, 91, 123, 129, 132

[218] I. Vatolkin, M. Preuss, and G. Rudolph. Training set reduction based on 2-gram feature statistics for music genre recognition. Technical report, TR13-2-001, Faculty of Computer Science, Technische Universität Dortmund. Presented at the 2012 Workshop on Knowledge Discovery, Data Mining and Machine Learning (KDML), 2013. 10, 42, 43, 70, 91, 93, 129

[219] I. Vatolkin, M. Preuß, G. Rudolph, M. Eichhoff, and C. Weihs. Multi-objective evolutionary feature selection for instrument recognition in polyphonic audio mixtures. *Soft Computing*, 16(12):2027–2047, 2012. 7, 10, 40, 42, 54, 63, 70, 87, 89, 91, 122, 123, 129

[220] I. Vatolkin and W. Theimer. Optimization of feature processing chain in music classification by evolutionary strategies. In G. Rudolph, T. Jansen, S. M. Lucas, C. Poloni, and N. Beume, editors, *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature (PPSN)*, volume 5199 of *Lecture Notes in Computer Science*, pages 1150–1159. Springer, 2008. 9, 46, 54, 70, 81, 129

[221] I. Vatolkin, W. Theimer, and M. Botteck. AMUSE (Advanced MUSic Explorer) - a multitool framework for music data analysis. In J. S. Downie and R. C. Veltkamp, editors, *Proceedings of the 11th International Society on Music Information Retrieval Conference (ISMIR)*, pages 33–38, 2010. 9, 18, 30, 93, 129

[222] I. Vatolkin, W. Theimer, and M. Botteck. Partition based feature processing for improved music classification. In W. A. Gaul, A. Geyer-Schulz, L. Schmidt-Thieme, and J. Kunze, editors, *Proceedings of the 34th Annual Conference of the German Classification Society (GfKl), 2010*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 411–419, Berlin Heidelberg, 2012. Springer. 9, 18, 40, 41, 42, 77, 129

[223] I. Vatolkin, W. Theimer, and G. Rudolph. Design and comparison of different evolution strategies for feature selection and consolidation in music classification. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 174–181, 2009. 9, 43, 54, 70, 74, 107, 129, 133

[224] A. L. Wang. An industrial strength audio search algorithm. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, pages 7–13, 2003. 12

[225] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009. 79

[226] C. Weihs, K. Friedrichs, M. Eichhoff, and I. Vatolkin. Software in music information retrieval (MIR). In W. A. Gaul, A. Geyer-Schulz, L. Schmidt-Thieme, and J. Kunze, editors, *Proceedings of the 34th Annual Conference of the German Classification Society (GfKl), 2010*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 421–432, Berlin Heidelberg, 2012. Springer. 10

[227] C. Weihs, U. Ligges, F. Mörchen, and D. Müllensiefen. Classification in music research. *Advances in Data Analysis and Classification*, 1(3):255–291, 2007. 38, 44

[228] R. J. Weiss and J. P. Bello. Unsupervised discovery of temporal structure in music. *IEEE Journal on Selected Topics in Signal Processing*, 5(6):1240–1251, 2011. 44

[229] K. West and S. Cox. Incorporating cultural representations of features into audio music similarity estimation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):625–637, 2010. 45

[230] G. Widmer, D. Rocchesso, V. Välimäki, C. Erkut, F. Gouyon, D. Pressnitzer, H. Penttinen, P. Polotti, and G. Volpe. Sound and music computing: Research trends and some key issues. *Journal of New Music Research*, 36(3):169–184, 2007. 22

[231] A. Wieczorkowska, P. Synak, R. A. Lewis, and Z. W. Raś. Extracting emotions from music data. In M.-S. Hacid, N. V. Murray, Z. W. Raś, and S. Tsumoto, editors, *Proceedings of the 15th International Symposium on Methodologies for Intelligent Systems (ISMIS)*, volume 3488 of *Lecture Notes in Computer Science*, pages 456–465. Springer, 2005. 38

[232] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, San Francisco, 2005. 54

[233] D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996. 46, 82

[234] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997. 58

[235] Y.-H. Yang, C.-C. Liu, and H. H. Chen. Music emotion classification: a fuzzy approach. In K. Nahrstedt, M. Turk, Y. Rui, W. Klas, and K. Mayer-Patel, editors, *Proceedings of the 14th ACM International Conference on Multimedia*, pages 81–84. ACM, 2006. 131

[236] K. Yoshii. *Studies on Hybrid Music Recommendation Using Timbral and Rhythmic Features*. PhD thesis, Graduate School of Informatics, Kyoto University, 2008. 22

[237] H. Yu and S. Kim. SVM tutorial - classification, regression and ranking. In G. Rozenberg, T. Bäck, and J. N. Kok, editors, *Handbook of Natural Computing, Volume 1*, pages 479–506. Springer, Berlin Heidelberg, 2012. 50

[238] J. Zhu, X. Xue, and H. Lu. Musical genre classification by instrumental features. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 580–583, 2004. 22, 23

[239] Z. Zhu, S. Jia, and Z. Ji. Towards a memetic feature selection paradigm. *IEEE Computational Intelligence Magazine*, 5(2):41–53, 2010. 67

[240] Z. Zhu, Y.-S. Ong, and J.-L. Kuo. Feature selection using single/multi-objective memetic frameworks. In C.-K. Goh, Y.-S. Ong, and K. C. Tan, editors, *Multi-Objective Memetic Algorithms*, volume 171 of *Studies in Computational Intelligence*, pages 111–131. Springer, Berlin Heidelberg, 2009. 53, 68

[241] E. Zitzler. Evolutionary multiobjective optimization. In G. Rozenberg, T. Bäck, and J. N. Kok, editors, *Handbook of Natural Computing, Volume 2*, pages 871–904. Springer, Berlin Heidelberg, 2012. 58

[242] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, volume 1498 of *Lecture Notes in Computer Science*, pages 292–304. Springer, 1998. 60

List of Symbols

α Type I error in hypothesis testing
α_i^{SVM} Lagrange multipliers in SVM classification
α_F Balance parameter for F-measure
β Type II error in hypothesis testing
γ Probability of bit flip in symmetric mutation
λ EA offspring population size
μ EA population size
$\overline{\mathcal{S}}_{fin}^{H}$ Mean final dominated hypervolume on holdout set
$\overline{\mathcal{S}}_{init}^{H}$ Mean initial dominated hypervolume on holdout set
$\overline{\mathbf{a}}$ Mean value of $\mathbf{a}$
ϕ High-level feature group share factor
$\Phi(\mathbf{X}^*, \theta)$ Feature subset after FS application
τ_α Test statistic rejection value
$\mathbf{w}$ Weights in SVM optimisation
θ Indices of selected features after FS
$\varphi^{SVM}(\mathbf{x})$ SVM mapping to higher dimensional domain
$\widetilde{\mathbf{a}}$ Standard deviation of $\mathbf{a}$
ξ Experimental feature relevance
$\xi^{SVM}(\mathbf{x}_i)$ Slack variable for classification instance i in SVM classification
B_f Number of spectrum frequency bins
b_r Bit range, or depth
C Number of classification categories
c_ρ Spearman's rho rank coefficient
$CFT_{\mathbf{s}}(f^*)$ Amplitudes of continuous Fourier spectrum
D Length of discrete time signal
$d_{JS}(\mathbf{wp}, \mathbf{ws})$ Jenson-Shannon divergence between vectors **wp** and **ws**
$d_{KL}(\mathbf{wp}, \mathbf{ws})$ Kullback-Leibler divergence between vectors **wp** and **ws**
$DFT_{\mathbf{s}}(f)$ Amplitudes of discrete Fourier spectrum
e^{PD} Phase space delay
F Number of processed feature dimensions
f Discrete frequency
F^* Sum of all feature dimensions after $\mathcal{FE}$
f^* Continuous frequency
F_N^* Number of (multidimensional) features after $\mathcal{FE}$
$F^{**}(i)$ Number of raw feature dimensions of feature i
f_s Sampling frequency

FN False negative number
FP False positive number
if_r Initial feature rate
m^{PD} Phase space dimensionality
m^{RF} Number of candidate variables for nodes in random forest
m_c Correlation coefficient
m_{ACC} Accuracy
m_{AE} Absolute error
m_{BRE} Balanced relative error
m_F F-measure
m_{GEOM} Geometric mean
m_{L+} Positive likelihood
m_{L-} Negative likelihood
m_{MSE} Mean squared error
m_{PREC} Precision
m_{REC} Recall
m_{RE} Relative error
m_{SPEC} Specificity
m_Y Youden's index
n Number of folds in n-fold cross-validation
N_f^{SC} Number of analysis frames for structural complexity estimation
O Number of evaluation metrics, or objectives
P_{ND} Size of non-dominated front
$PCP(p_w)$ Pitch class profile of pitch class p_w
$R(\cdot)$ Rank of input variable
S_a Step size of algorithm analysis frame in seconds
S_c Step size of classification frame in seconds
S_e Step size of extraction frame in samples
T Number of classification frames
t Discrete time
T' Number of song classification frames
t^* Continuous time
T^H Time dimensionality of harmonised feature matrix $\mathbf{X}^H$
$T^{**}(i)$ Number of extraction windows for estimation of a raw feature i
T_S^U Mann-Whitney U-test statistic
T_S^W Wilcoxon signed rank test statistic
TN True negative number
TP True positive number
U Constraint number in MOP
W_a Length of algorithm analysis frame in seconds
w_a Sine wave amplitude
W_c Length of classification frame in seconds
W_e Length of extraction frame in samples

w_f	Sine wave frequency
w_l	Sine wave length, or period
y_L	Labelled instance class relationship
y_P	Predicted instance class relationship
$\mathbf{m}$	Evaluation metric vector
$\mathbf{n}$	Time and value series index vector
$\mathbf{p}$	Semitone spectrum bins
$\mathbf{p}^o_{\mathcal{T}}$	Optimal algorithm parameters
$\mathbf{p}_w$	Wrapped semitone spectrum bins, or pitch classes
$\mathbf{p}_{\mathcal{T}}$	Algorithm parameters
$\mathbf{q}$	Bit vector in evolutionary FS
$\mathbf{r}$	Reference point for $\mathcal{S}$ estimation
$\mathbf{s}(t)$	Discrete time signal
$\mathbf{s}(t^*)$	Continuous time signal
$\mathbf{u}, \mathbf{v}$	Sample vectors for statistical comparison
$\mathbf{X}'$	Song processed feature matrix
$\mathbf{X}^*$	Raw (original) feature matrix
$\mathbf{X}$	Processed feature matrix
$\mathbf{X}^H$	Harmonised feature matrix before further $\mathcal{FP}$ steps
$\mathbf{y}_L$	Labelled category relationships
$\mathbf{y}_P$	Predicted category relationships
$\mathcal{A}_C$	Critical area for H0 rejection
$\mathcal{CT}$	Classification training task
$\mathcal{C}$	Classification task
$\mathcal{FE}$	Feature extraction task
$\mathcal{FP}$	Feature processing task
$\mathcal{M}_C$	Classification model
$\mathcal{O}_M$	Multi-objective optimisation task
$\mathcal{O}_S$	Single-objective optimisation task
$\mathcal{Q}^{DT}$	Decision tree query
$\mathcal{TS}$	Time series
$\mathcal{T}$	Music classification chain tasks
$\mathcal{VS}$	Value series
$\mathcal{V}$	Classification validation task
$\mathcal{X}$	Decision space
$\mathcal{Z}$	Objective space
$\mathcal{P}_f$	Pareto front
$\mathcal{S}(\cdot)$	Hypervolume, or $\mathcal{S}$-metric
H0	Null hypothesis
H1	Alternative hypothesis
p	p-value

Index