Feature Selection for Classification of Music According to Expressed Emotion

Pasi Saari

Master's Thesis

Music, Mind & Technology

December 2009

University of Jyväskylä

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄN YLIOPISTO

Tiedekunta – Faculty Humanities	Laitos – Department Music

Tekijä – Author

Pasi Saari

Työn nimi – Title

Feature Selection for Classification of Music According to Expressed Emotion

Oppiaine – Subject Music, Mind & Technology	Työn laji – Level Master's Thesis
Aika – Month and year December 2009	Sivumäärä – Number of pages 74

Tiivistelmä – Abstract

This thesis suggests a framework for studying wrapper selection in emotion recognition from musical audio. Wrapper selection is a feature selection approach that can be used to improve the performance of a classifier by reducing the dimensionality of the input feature set into a smaller subset that decreases problems related to the curse of dimensionality, redundancy and irrelevancy of the input features. The search is guided by the performance of the classifier which ideally leads to the optimal or semi-optimal feature subset. However, the attainable benefits of the method are limited due to its computational intensiveness and tendency towards overfitting. Especially the problem of overfitting, which relates to the lack of generalizability of the obtained results to unknown test data, is a critical issue in wrapper selection. Therefore special caution should be taken when applying the method and reporting the obtained results.

It will be argued that the previous research on wrapper selection in the field of Music Information Retrieval (MIR) has reported supposedly misled and over-optimistic results due to inadequate methodology indicative of overfitting. The suggested framework in this thesis concentrates on addressing this problem by exploiting a cross-indexing algorithm that has been reported to yield unbiased estimates of the performance of wrapper selection. Applying the algorithm leads to realization of a model parameter selection problem where the optimal dimensionality of the feature space for a given classifier is searched. A modification to the cross-indexing algorithm was suggested to inherently favor small feature sets in an attempt to increase the generalizability and interpretability of the obtained models. Although the framework is used in this thesis for emotion recognition from music, it can as well be exploited in other MIR-related classification tasks. Cross-indexing has not been used in music classification previously.

The analysis was conducted with Naive Bayes, k-Nearest Neighbors and Support Vector Machine classifiers with forward selection and backward elimination search methods. Comparison of the classifier-search method combinations yielded highest performances for k-NN with backward elimination. The best subset size for k-NN was 4 features which yielded 56.5% accuracy. The most useful features for k-NN were related to harmony and dynamics. Especially mode, i.e. majorness of the excerpts, and key clarity were found to increase the performance of the classifier. Generally, it was seen that classification can truly benefit from wrapper selection in terms of generalizability, classification accuracy and interpretability of the obtained models. Moreover, the proposed modification to the algorithm increased the generalizability and reliability of the results when compared to the original algorithm.

Asiasanat – Keywords musical emotions, musical features, feature selection, wrapper selection, overfitting, cross-indexing

Säilytyspaikka – Depository

Muita tietoja – Additional information

Contents

Chapter 1

Introduction

Automatic recognition of emotions in musical audio has gained increasing attention in the field of Music Information Retrieval (MIR) during the past few years. This is reflected for example in the number of submitted systems in the annual Audio Music Mood Classification (AMC) contest part of the Music Information Retrieval Evaluation eXchange[1] (MIREX). Since the initial launch of the contest in 2007 the number of submitted systems has increased from 9 to 33. The used approaches have had great variance within AMC due to the diversity of information possible to extract from music as well as diversity of methods to infer meaningful knowledge about the analyzed data. Even more variance is seen in the approaches in the field in general – an understanding of the psychological background of emotions or moods in music and the conventions relating to machine learning are still far from being established.

The development in the field has coincided with the growing interest for providing large collections of digital audio for the public via web services such as Spotify[2] and Last.fm[3]. The management of these collections as well as retrieving information from them can be diversified and smoothed by the research in the field of MIR. In this context, expressed emotions and moods are indeed relevant and important characteristics in music.

In addition to promoting the technical development and to meeting its demands, automatic emotion recognition can benefit the understanding of music perception. Despite the research in musicology studying the influence of specific cues in the musical structure on emotional expressions (Gabrielsson and Lindström, 2001), there is no complete analytical knowledge or consensus about what "ingredients", i.e. acoustical features possible to extract from music are required to build the optimal models for emotion recognition, in the limits imposed by the subjectivity of emotion. Machine learning comprises methods developed for

[1] `http://www.music-ir.org/mirex/20xx/index.php/Main_Page`

[2] `http://www.spotify.com/`

[3] `http://www.last.fm/`

detecting these type of relations automatically by taking account the interrelations of the features, but problems arise from either inadequacy of the used features for capturing pieces of perceptually meaningful knowledge (Aucouturier et al., 2007) or from the complexity or technical deficiencies in the feature sets (Guyon and Elisseeff, 2003). This prevents machine learning methods from reaching their full potential to improve the understanding about the underlying phenomena affecting music perception.

A part of the research field have adopted the view that linear models are more useful for understanding emotions in music than classifiers (Eerola et al., 2009). The most adequate linear models transform the feature space used in learning into few dimensions constituting of sets of input features while retaining the predictive information about the target concept. On the other hand, classifiers traditionally exploit the features independently to build the optimal model. The prevalent downside to this approach in MIR has been the large dimensionality of the feature space which leads to badly interpretable models and thus low contribution to the understanding of the phenomenon under study. Another curse relating to the high dimensionality of the input data given to a learning method is that it leads to overfitting which is reflected in the low degree of generalizability of the models in classifying unknown data.

Different dimension reduction methods applied to the input data have been developed to deal with problems related to high dimensionality in machine learning. Wrapper selection (Kohavi and John, 1997) is a method that can be used in finding a subset of input features optimal for a given classifier. Perhaps surprisingly, also wrapper selection itself is highly prone to leading to overfitting when the found subsets are used to build classification models (Reunanen, 2003). The analysis of the previous research in MIR (detailed in the section 2.2.4) shows that the use of wrapper approach has almost constantly led to disputable results indicating overfitting of the data. Fiebrink and Fujinaga (2006) addressed this pitfall in music classification by applying guidelines for building a wrapper selection and classification framework given in two machine learning studies by Reunanen (2003, 2004). Since then, Reunanen (2006, 2007) developed the guidelines further and suggested a cross-indexing algorithm that was shown to yield unbiased estimates of the performance of classification models. Applying the algorithm leads to a framework that is essentially a realization of a model parameter selection problem where the optimal dimensionality of the feature space for a given classifier is searched. Cross-indexing has not been used in music classification previously.

This study has two aims: to develop a framework for studying wrapper selection in music classification applying the cross-indexing algorithm, and to show that classification in MIR, specifically in the recognition of expressed emotions in music, can lead to interpretable models when the number of features used to build the models is reduced dramatically into very few features while maintaining or increasing the explanation capacity and generalizability of the models. What will be seen is that by modifying the cross-indexing algorithm to inherently favor small feature sets the generalizability of the models is increased. The subtlety of the chosen approach is that rather than combining a large number of features to represent few dimensions open to interpretations as in linear modeling, the gained dimensions will be exactly those single features itself whose relations to the phenomenon under study are at least ideally understood.

The study also outlines ways to contribute to understanding of the relation between acoustical features and musical emotions and to examine the performance of the models by using the results to derive knowledge that extend beyond the traditional performance measures. The aim is reached by combining the results of a multitude of models output by the framework and by exploiting a rich representation of the perceived emotional content of the analyzed data.

Chapter 2

Background

2.1 General Background from Machine Learning

This section defines and discusses the background concepts. First, description of feature-based classification is given in the section 2.1.1 to explain the general methodological framework of the present study. The section 2.1.2 discusses the challenges in feature-based classification that are caused by the feature sets and primes the section 2.1.3 which describes the basic concepts of feature selection, eventually concentrating on the used wrapper selection method. Last, the section 2.1.4 concentrates on overfitting, a phenomenon in inductive learning which affects the generalizability of the models obtained with classification, especially if feature selection is used prior to the classification phase.

2.1.1 Feature-based Classification

Feature-based classification as a machine learning approach is defined in this study in the context set by two distinctions (Han and Kamber, 2001). The first distinction relates to the visibility of the target concept to be learned, whether the concept is known to the learner and exploited in the process of learning or if the concept is invisible and therefore not used in the learning process. The first case is called supervised learning while the second case is called unsupervised learning, or clustering, in which subsets of objects are grouped together based on their similarity. The other distinction can be made between classification and prediction. In classification the target concept is categorical and the values that the concept takes are called classes. For example in this study, with the target being the expressed emotion in music, the classes can be for instance 'happy' and 'sad'. In prediction the target concept is represented in continuous scale(s). A scale could be for instance the amount of happiness a music expresses, e.g. measured on a scale from 1 to 9.

In the presented framework, feature-based classification belongs to the category of supervised classification. The basis of feature-based classification is the representation of information that defines the objects, or instances, to be classified. The information in this case is given in the form of *features*. A formalization of the approach is given in (Liu and Motoda, 1998, p. 5): An instance is a pair $(x, f(x))$, where x is an N-dimensional feature vector denoting the input and $f(x)$ is the discrete output, which is one of the pre-defined classes. For the given data and the chosen learning algorithm, the classification consists of three steps (Liu and Motoda, 1998, p. 5):

1. **Learning a classifier, i.e. description of the target concept.** A function h that approximates f is learned given a collection of instances $(X, f(X))$.

2. **Classifying data using the classifier.**

3. **Measuring the classifier's performance.** Classifier's performance can be measured for example by predictive accuracy, learning speed or complexity of the learned model.

To obtain realistic performance measures, the data used in learning should be separate from the data used in classification. This is achieved by splitting the available data into train and test sets. An option is to use *cross-validation* in estimating the performance. Cross-validation involves splitting the data randomly several times, defined by the chosen number of folds n, and averaging the estimates to get the final measure. In cross-validation the steps described above are taken n times, each time splitting the data randomly into a train and test sets consisting of proportions of $\frac{n-1}{n}$ and $\frac{1}{n}$ out of the available data respectively. The final performance measure is the average over the n folds.

There is a large variety of learning algorithms used in classification. The main approaches include decision tree induction whose idea is to construct a structured list of features based on their statistical properties in the separation of the classes, Bayesian classification, which is based on class probabilities, backpropagation, which is a learning algorithm for neural networks modeling the neural activity in human brain, and instance-based learning relying on the similarities between the instances. The algorithms used in this study are described and discussed in the section 3.1.2.1 on page 37.

2.1.2 Challenges of Feature Sets

Usually in machine learning it is customary to define a feature set to be large as possible to cover all aspects of the phenomenon under study. This practice is well-justified but the large set of predictors can lead to problems. This has aroused interest in the field towards studying the general properties of data used in predictive analysis as well as representing the data in a more compact manner while retaining their predictive potential. The most crucial problem in a large set of data is the *curse of dimensionality* explained next.

2.1.2.1 Curse of Dimensionality

The term curse of dimensionality was initially coined by Richard Bellman (1961). Bellman considered the problems in his work due to the curse of dimensionality so difficult that it led him to state the frequently quoted words:

> “[I]n view of all that we said in the foregoing sections, the many obstacles we appear to have surmounted, what casts the pall over our victory celebration? It is the curse of dimensionality, a malediction that has plagued the scientist from the earliest days." (Bellman, 1961)

Curse of dimensionality describes the problem that is caused by adding extra dimensions (features) to a space. The curse of dimensionality can be demonstrated by comparing two feature spaces as observations of the same phenomenon. If both spaces hold 100 evenly scattered instances scaled on a unit interval such that the other space is formed by two independent features while the other is formed by 10 independent features, they create respectively 2-dimensional and 10-dimensional spaces, i.e. a square and a 10-dimensional hypercube. If all but one feature is held constant, in the square there are 10 observations of the phenomenon - 0.1 unit distances apart - which makes it possible to analyze the effect of changing the value of a particular feature, whereas in the hypercube the distance is 0.63 units along a feature-axis[1]. This means that the effect cannot be observed; the amount of instances required in the hypercube grows exponentially with the number of dimensions. Moreover, the average Euclidean distances between two closest neighbors in these spaces are approximately 0.14 and 2.00 units[2] which arouses doubt whether the effects constituting the observed phenomenon could be reliably measured in the hypercube at all.

The problem is at its biggest with learning methods that give each feature an equal amount of importance, for example with instance-based and Bayesian learners, and lesser with learners that can handle features non-uniformly, giving them differing weights.

[1] The distances are counted by $\frac{1}{\sqrt[n]{100}}$, where n denotes the number of dimensions.

[2] $\simeq \sqrt{n\left(\frac{1}{\sqrt[n]{100}}\right)^2}$.

2.1.2.2 Feature Redundancy

Adding extra dimensions to the feature space also increases the possibilities of the occurrences of two phenomena - feature redundancy and feature irrelevance. Redundancy of a feature can be measured by the mutual information between the feature and another feature in the set. Mutual information between two features is informally determined by the amount of information that one feature contains about the other. This is equivalent to the information contained in one feature minus the information contained in the feature when the other feature is known. Formally, in relation to processes containing information, a general definition is given in Shannon and Weaver (1949):

$$I\left(X,Y\right)=H\left(X\right)+H\left(Y\right)-H\left(X,Y\right)=H\left(X\right)-H\left(X\,|\,Y\right)=H\left(Y\right)-H\left(X\,|\,Y\right),$$

where $H\left(\cdot\right)$ is entropy and $H\left(\cdot|\cdot\right)$ is conditional entropy.

An indication of feature redundancy is correlation. This has an effect on the dimensionality of the data since correlations reduce the independency of the dimensions, thus presumably reducing also the curse of dimensionality. However, in classification feature redundancy has usually negative effects. The reason is that two redundant features can be thought as realizations of a same phenomenon. Thus, for a classifier treating each feature with the same importance, the effect of a particular phenomenon measured by two features can be exaggerated, leading to misleading results.

However, although feature redundancy is an important characteristic in data and can contribute to problems in classification, it is not necessarily a phenomenon that needs to be avoided. Guyon and Elisseeff (2003, p.1163-1164) used artificial data to demonstrate cases where redundancy is in fact beneficial. They showed that adding presumably redundant features can lead to noise reduction and better class separation.

2.1.2.3 Feature Irrelevance

Also feature irrelevance can mislead the learning by causing noise in the data and obscuring possibly relevant effects. Feature irrelevance and feature relevance has been a topic of interest because knowing that a feature is irrelevant would mean that it is not needed for the learning algorithm. However, it has been noted that even the definition of feature irrelevance or feature relevance is problematic.

John et al. (1994) analyzed several general definitions of relevance and revealed that all of them led to unrealistic results. Therefore they suggested that two types of relevance must be defined – *weak* and *strong.* These definitions were given in terms of a *Bayes classifier*, the optimal classifier for a given problem, as described by Duda and Hart (2001). Informally, a feature X_i is *strongly relevant* if its removal alone causes deterioration in the performance of a Bayes classifier, a feature X_i is *weakly relevant* if it is not strongly relevant and if there exists a subset S of features such that the performance of a Bayes classifier built on S is worse than on $S \bigcup \{X_i\}$ and a feature is irrelevant if it is not strongly or weakly relevant (John et al., 1994).

Although these definitions might be sound, their usefulness in practice is limited due to the Bayes classifier used as a basis for the definitions. Building of the Bayes classifier would require the true underlying distributions to be known. Therefore finding the ideally optimal classifier is not possible in the inductive learning, i.e. formation of general conclusions from a number of observations, since it would require an infinite number of observations. Moreover, strong or weak relevance do not determine whether a feature should be used by a given learning algorithm in practice as algorithms can benefit from omission of even strongly relevant features as well as inclusion of irrelevant features (Kohavi and John, 1997). An important notion is also that determining whether a feature is strongly or weakly relevant or irrelevant is *NP*-hard problem since it requires all possible subsets of features to be tested which makes it impossible to solve the problem even within medium-sized feature sets.

To avoid these problems, Caruana and Freitag (1994) distinguished the concept of relevance from usefulness. Defined in relation to a given learning algorithm L, feature set S and instances I, a feature X_i is *useful* to L with respect to S if the accuracy of the hypothesis that L produces using the feature set $S \bigcup \{X_i\}$ is better than the accuracy achieved using S.[3] While feature relevance can be considered a general property of a feature and often only ideally observable and exploitable, usefulness is easily demonstrated only by measuring the performance of a learning algorithm with and without the feature.

It must be emphasized that usefulness or relevance of a feature is always determined in the context of another features. As Guyon and Elisseeff (2003, p.1165) pointed out, usefulness of a feature does not imply that it is useful by itself. This is where the limitations of the above definition lies. Whether a feature is useful or not with respect to a single specific feature set might not be of much value because even a small available feature set contains a large number of possible subsets, presumably giving differing results for usefulness of the feature.

In an attempt to address these problems by computational means, a variety of classifiers have been developed and modifications to basic classifiers have been suggested. Another

[3] This definition is evolved by Blum and Langley (1997).

approach to that aim is to use dimension reduction, namely *feature selection*, which can be used in context of any classifier and therefore proposes a general solution. The discussion above highlighted the complexity of the problems related to the properties of features used in classification. In that regard the topic of this study, feature selection, is far from trivial.

2.1.3 Feature Selection

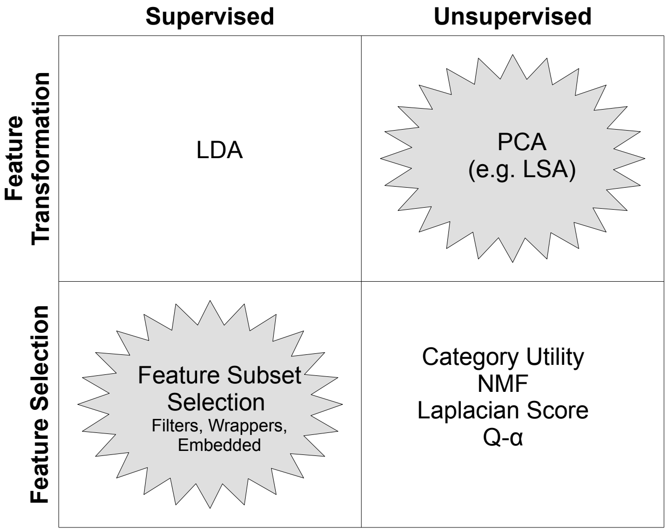
Given the data containing the instances represented by features and the targets, the problem of *feature selection* can be formulated as a task of finding a subset of the original features which maximizes the performance of a given learning algorithm run on the data. By reducing the amount of data used in learning, feature selection can reduce the problems described in the previous section. Consequently, feature selection can reduce the required computational effort in learning and classification, reduce the complexity of the models, making them easier to be interpreted and increase the generalization capabilities of the models.

Feature selection forms a branch in the field of dimension reduction, whose idea in general is to represent a feature space with less dimensions containing as much relevant information about the phenomena under study as possible. Dimension reduction can be performed through various types of methods. A comprehensive description of the field is given for example in (Cunningham, 2008), where the set of dimension reduction techniques was partioned in terms of two categorizations – each dimension reduction method applies either to supervised or unsupervised learning[4] and reduces the dimensionality of the feature space by either feature selection or *feature transformation*. Figure 2.1 illustrates the categories.

The most popular approaches of dimension reduction fall into the categories of supervised feature selection and unsupervised feature transformation.[5] Feature transformation either reduces the dimensionality of the feature space by combining the original features or generates completely new features. The former is called *feature extraction* and the latter *feature generation*. An example of feature extraction is the popular Principal Component analysis (PCA). PCA forms linear combinations of the features, so that the correlation between the components is minimized while the covered variance of the feature space is maximized. Feature

[4] This categorization does not strictly determine the type of learning algorithm (whether it is supervised or not) eventually used with the feature sets, it rather refers to the visibility of the precise target function to the dimension reduction method. Exceptions to this are the wrapper and embedded methods described later.

[5] The other approaches include Linear Discriminant Analysis (LDA), which is a supervised technique related to PCA, Category Utility for clustering purposes (Devaney and Ram, 1997), Nonnegative Matrix Factorization (NMF) that in fact can be used in both supervised and unsupervised extraction and selection (Lee and Seung, 1999), Laplacian score (supervised and unsupervised) (He et al., 2006) and the Q-α algorithm for clustering (Wolf and Shashua, 2005).



	Supervised	Unsupervised
Feature Transformation	LDA	PCA (e.g. LSA)
Feature Selection	Feature Subset Selection Filters, Wrappers, Embedded	Category Utility NMF Laplacian Score Q-α

Figure 2.1: Dimension reduction approaches (table adopted from (Cunningham, 2008)).

selection methods, on the other hand, select a subset of features to be used in their original form.

There are justifications for using either one of these two opposites. PCA can be used more in explanatory purposes, to understand the relations in the data. The models built on only few components can be interpreted easily. However, as the method does not exploit the information about the target concept, the explanatory potential of the reduced dimensions is by no means guaranteed. For this reason, supervised approaches are generally preferred for classification purposes.

Supervised feature selection benefit from the exploitation of the target concept and from the fact that it leads to reduced amount of information used in classification. In the predictive analysis of musical audio an important benefit of feature selection is also the lesser amount of information that must be collected for classification; musical features are in fact extracted through various types of feature extraction from audio and require much computational effort (see the section 2.2.1).

Perhaps the most straight-forward feature selection technique is *feature ranking,* which forms a ranked list of features according to their predictive potential[6]. From the list the top n features can then be used in the final subset. The major drawback of feature ranking, due to the problems described in the last section, is that feature relevance and redundancy cannot be evaluated independently from the other features. Therefore the subset of n top-ranked features can contain a high amount of redundancy as well as small amount of variable complementarity which can cause low prediction potential. To avoid the described drawback,

[6] This technique does not require search method since all features are evaluated separately.

feature selection is commonly implemented with subset evaluation, which enables evaluating features in the context of other features.

Feature selection with subset evaluation requires defining how to search the space of feature subsets (*search method*) and what measure to use when evaluating a feature subset (*evaluation criterion*) as well as the *initial feature set* and a *termination condition*. The process starts with the initial set of features. The initial set can be empty set, set of all features or any subset defined by the user. The set is then evaluated with a chosen criterion. Then, using the search method, the initial set is modified and a new subset is formed and evaluated. The procedure is repeated iteratively, each time modifying the feature set and evaluating it until the termination condition is met or the search space, i.e. all possible solutions with the used search method, has been explored. A widely used condition is to stop if the performance does not improve in few consecutive iterations.

2.1.3.1 Evaluation Criteria

Two main approaches can be distinguished based on the type of evaluation criterion used in the search: *filter* and *wrapper*.[7] In the filter approach, a chosen information theoretic measure is used as an evaluation criterion. For example mutual information criterion can be used to reduce redundancy in the feature set or a more complex criteria such as Correlation-based Feature Subset selection criterion (CFS) (Hall, 1998) can be used. CFS takes both redundancy and predictive potential of individual variables in the subset into account. Filter approach can be thought as a pre-processing step before the learning phase and it is independent from the choice of the learner.

In the wrapper approach each feature subset is evaluated with the learning algorithm ultimately to be used when classifying the data. The approach was first introduced in John et al. (1994), where it was also tested briefly. John et al. (1994) claimed that a subset selection algorithm must take into account the biases of the learning algorithm in order to select a subset with highest possible prediction accuracy with the learning algorithm on unseen data. More extensive evaluation on data from various fields together with definitive formulation of the approach was given in (Kohavi and John, 1997). Wrapper performed favorably on some datasets in comparisons to a filter selection algorithm and induction with no selection.

The figure 2.2 illustrates the process. First the data are split into train and test sets. The train set is used in the feature selection while keeping the test set only for the final evaluation of the performance of the induction algorithm. Then, the search is conducted using a chosen search method and by evaluating each candidate subset with respect to the performance of

[7] The third approach is to embed the feature selection in the used learning algorithm.

the learning algorithm. The performance is assessed usually either through cross-validation or using a validation set that is separate from the train and test sets. After the terminating condition is met the learning phase is conducted on the train set represented by the selected feature subset. Last, the output model is used in evaluating the test set. Note that the filter approach can be implemented in a similar framework to the one illustrated in the figure, but replacing the performance assessment and learning / induction with the general evaluation criterion.

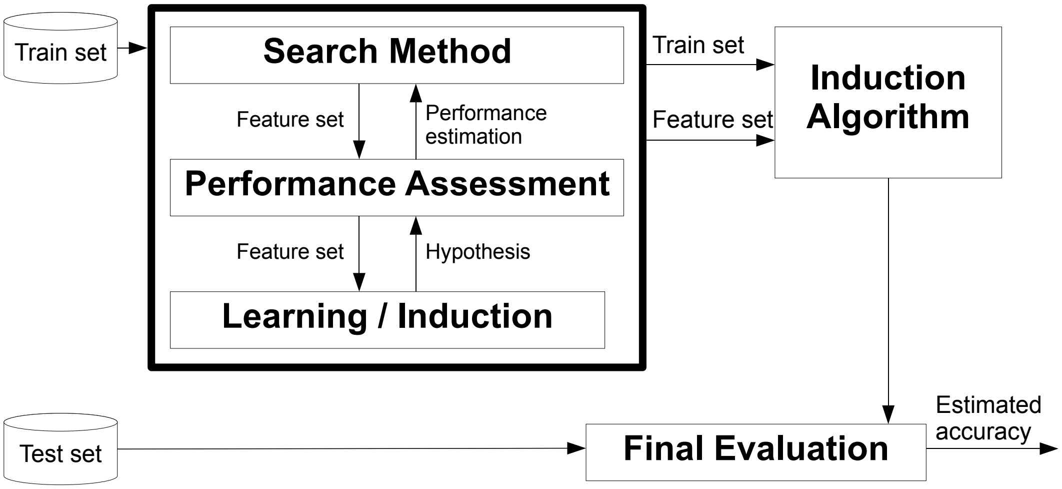


Figure 2.2: Wrapper selection (chart adopted and slightly modified from (Kohavi and John, 1997)).

Both filter and wrapper approaches have their benefits. Filters in general require less computational effort than wrappers as the latter requires the learning algorithm to be run multiple times in the selection process. Another benefit of filters is that they are not dependent on the learning algorithm. Therefore the obtained feature set can be thought of as a universal semi-optimal set that can be compatible with any learning method. On the other hand, the fact that the wrapper approach is dependent on the used learning algorithm can be viewed as its advantage: the approach makes it ideally possible to find the optimal subset of features for a given learning algorithm, resulting into higher performance than what may be possible with the filter approach.

2.1.3.2 Search Methods

Finding the optimal subset of features requires searching the whole feature space. This involves as many evaluations as there are possible subsets. If the feature set contains N features, the number of possible subsets is 2^N. This makes the problem NP-hard and an *exhaustive* search method, which involves searching through all possible subsets becomes prohibitively expensive as the number of features increases. Therefore, *heuristic* search methods have been developed in order to reduce the number of subsets that are evaluated. Many popular heuristic algorithms used in feature selection exploit *greedy selection heuristics,* which always makes the choice that looks best at the moment (Cormen et al., 2001, p.370). This means that the greedy heuristic makes locally optimal choices and so it does not guarantee that the search will lead to the globally optimal subset. Two of the most famously used greedy selection algorithms are forward selection (FS) and backward elimination (BE). They are called *stepwise algorithms* since they involve addition or removal of only one feature at each modification of the feature subset:

Forward selection: The search starts with an empty set of features. At each iteration the next candidate subsets are formed by expanding the current subset with each feature that is not yet in the subset, one at a time. The feature whose addition results in the best performance is added in the subset. The iterations are repeated until the termination condition is met or the full set is reached.

Backward elimination: The search starts with the full feature set. At each iteration the next candidate subsets are formed by eliminating each feature, one at a time, from the current subset. The feature whose elimination resulted in the highest performance improvement is eliminated permanently. The search stops when the termination condition is met or only one feature remains in the set.

It is very likely that both methods fail at finding the globally optimal subset since the number or evaluated subsets is $O\left(N^2\right) \lll O\left(2^N\right)$. Therefore, modifications to the greedy stepwise heuristic have been developed such as sequential floating selection (SFS) (Pudil et al., 1994) that can be implemented with either forward selection or backward elimination. The sequential floating forward selection (SFFS) consists of applying backward steps at each iteration. This phase called *backtracking* continues until the performance of the subset cannot be improved. If backtracking does not result in improvement after one feature elimination the phase is omitted. By inverting the search directions, the same procedure applies to the sequential floating backward selection (SFBS).

The SFS can be controlled by specifying the maximum number of backtracking steps per iteration. A small number of allowed backtracking steps reduces the required computation time but increases the possibility of getting stuck in a local minimum whereas a large number expands the search space, thus presumably increasing the performance of the found subset.

In addition to exhaustive and heuristic search methods, methods using *random subset generation* have been developed, for example the popular genetic search algorithm (Vafai and Jong, 1992). Common to the random generation methods is that at each iteration, the current subset is modified by a set or random feature removals and additions. Although the search space with these methods is $O\left(2^N\right)$, in practice the space is reduced by defining the maximum number of iterations.

2.1.4 Overfitting and Cross-Indexing

Overfitting relates to a problem that occurs when a learning algorithm fits the training data too well, considering peculiarities in the data such as noise or possible outliers as important characteristics of the phenomenon under analysis. Overfitting creates high variance in the models trained on different data for the purpose of predicting the same phenomenon, causing the smallest changes in the training data to have big effects on the model performance (Cunningham, 2000). Jensen and Cohen (2000) pointed out several reasons for avoiding overfitting:

- Overfitting produces incorrect models that indicate falsely that some features are relevant.
- Overfitted models need more space to store and more computational resources than models without unnecessary components.
- The models need larger feature sets than necessary, which increases the cost of building and storing the input data.
- Overfitted models are hard to interpret.
- Overfitted models produce high accuracy estimates on the data that they have been trained on. However, performances on new data are worse with overfitted models than with models that are not overfitted.

While the problem is well-known in classification and regression models, the effect grows in feature selection, especially in the wrapper approach, where the feature set is adapted to the bias of the learning algorithm.

Kohavi and John (1997) pointed out that in wrapper selection overfitting is caused by the fact that there are so many feature subsets that it is likely that one subset leads to high predictive accuracy on the hold-out data used in selection. This was demonstrated with randomly created data with binary class. While the maximal prediction performance is 50%

with the data, forward selection yielded 75% accuracy due to overfitting. A notion that data with larger sample size lowered the prediction accuracy towards the expected value indicated that the effect of overfitting is related to the sample size. The same problem can be seen in real-world data; Cunningham (2000) concluded a study by stating that overfitting becomes a serious problem in wrapper selection if the training data is not "sufficiently large to provide good coverage of the problem domain". Therefore the problem is very closely related to the curse of dimensionality.

Another cause of overfitting in feature selection is the size of the search space. This means that search algorithms exploring more subsets are more prone to overfitting than simpler search algorithms. Reunanen (2003) reviewed some previous studies that had celebrated the supremacy of complex searches over simple ones. When comparing the performance of SFFS and FS in wrapper approach, his study showed that SFFS produced better performance estimates on the training data but lower performance on the test data, implying the tendency of SFFS to overfit the training data. Loughrey and Cunningham (2005) studied overfitting in wrapper-based subset selection with search algorithms using random generation (simulated annealing and genetic algorithm). They proposed a method called early-stopping to reduce the search space. The study was based on the notions in Reunanen (2003).

While the problem of overfitting in wrapper selection can be reduced by proper choices of used learning algorithms and search methods, preferring as large data sets as possible, the issue should be taken into account also in the analysis frameworks and reporting of the results. Kohavi and John (1997) emphasized that in wrapper selection, the performance of the models on a test set unseen to the search process must be reported, not the (cross-validation) performance on the train set. However, guidelines on reporting the results addresses the problem only in retrospect and does not take into account the fact that the selection itself may have been led by unrealistic evaluation criteria and thus yielded a subset far from optimal.

To address the problem straightly, Juha Reunanen devoted a series of papers to developing guidelines to be exploited in the design of the analysis framework for wrapper selection (Reunanen, 2003, 2004, 2006, 2007). By following the guidelines the performance of wrapper selection with different search methods and learning algorithms can be compared. Reunanen pointed out that after running several analyzes, comparison between different methods or between different results obtained with the same method becomes a model selection problem.[8]

[8] In prediction of continuous target function with regression, several information criteria such as Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) can be used in model selection (Burnham and Anderson, 2004). In classification these types of criteria cannot be exploited in a straight-forward way.

The initial method for overcoming overfitting, as suggested by Kohavi and John (1997), was to run a feature selection algorithm on data obtaining the best subset found by the algorithm and to report the result of a model built on that subset and tested on an data unseen to the selection process. A suggestion was also made in the paper to use a so-called *external loop of cross-validation* that involved splitting the data into folds and running the feature selection process several times, each time using the fold unseen to the selection as a test set. The final performance is then the average of the maximum performances obtained in each run.

Reunanen (2004) studied an approach of determining the optimal subset size based on the accuracy estimates obtained in the wrapper selection process. In the analysis FS algorithm was tested with the external loop of cross-validation on data from different fields. The cross-validation accuracy estimates used to guide the selection as well as classification performances on independent test sets were recorded for all subset sizes. The results showed that the optimal subset sizes in terms of cross-validation accuracies used in the selection were not optimal in terms of classification error on test data, even when there existed a clear peaking phenomenon in the CV-accuracies. Therefore Reunanen (2004) suggested that the optimal subset size should be determined by performance on independent test data, preferably by averaging the performance estimates obtained in the external cross-validation loop separately for each subset size.

Reunanen (2006, p. 201-202) based his argument on the proof given in the article by Jensen and Cohen (2000) and stated that even the external cross-validation loop leads to overfitting. Determining the optimal subset size based on the maximum of the averaged classification accuracies on independent test sets yields unbiased estimate of the optimal subset size, but positively biased estimate of the performance attainable with the optimal subset size. It was pointed out that generally the value that is used to pick a certain model from a large set of candidates should not be used as an estimate of the performance. As a solution, Reunanen proposed *cross-indexing algorithms* A and B to be used in the model selection.

The cross-indexing algorithms are based on external cross-validation loop run with K folds.[9] The difference to the previous approach is in the computation of the optimal subset size estimates and their performance estimates. Both algorithm$s$ yield $k \in [1, 2, \ldots, K]$ estimates for the optimal subset size and their performances rather than only one. In algorithm A, each of the K estimates for the subset size is computed by taking the maximum of aver-

[9] In cross-indexing the loop is called the *cross-indexing loop*. Although it involves running feature selection several times, the results are not averaged in the same manner as in external cross-validation.

aged accuracies of $K-1$ runs, excluding the kth run, and defining the performance estimate of that subset size as the accuracy of the kth run. The algorithm B is analogous to the algorithm A, with the exception that the kth estimate for the subset sizes is the maximum of only kth run and the performance of that subset size is the averaged accuracy of $K-1$ runs, excluding the kth run. For both algorithms, the K estimates are averaged in order to obtain a single estimate of the optimal subset size and its performance. Reunanen (2006) tested these algorithms against the traditional external cross-validation loop and found that they reduced the bias significantly with data from various fields. This was due to the fact that with both algorithms, the optimal subset size was determined by different values than for the performance.

Reunanen (2007) discussed his previous work on cross-indexing and supposed that both proposed algorithms still yielded bias. Because of the number of runs averaged in computing each subset size estimate versus the number of runs averaged for each performance estimate, the algorithm A yields good estimate for the subset size but noisy estimate for its performance whereas the algorithm B yields noisy estimate for the subset size but accurate estimate for the selected size. As a solution he proposed a *generalized* $(N, K-N)$*-fold cross-indexing algorithm*[10] . In the generalized algorithm the parameter N is used to control the number of runs averaged in estimating the optimal subset size[11], reserving the $K-N$ other runs for the estimation of the gain of the performance estimate for the obtained subset size. The parameter N can be then considered as a trade-off between leaning towards accurate estimation of the optimal subset and towards accurate estimation of the performance attainable with the obtained subset size. Based on the preliminary results with the algorithm, Reunanen (2007) stated that the choice of $1 < N < K-1$ is a good trade-off between these two. At the two extremes, $N = K-1$ yields the cross-indexing algorithm A and $N = 1$ yields algorithm B.

The second part of the paper (Reunanen, 2007) concentrated on selecting the optimal model architecture among several candidates, which included selecting the optimal subset size and estimating its performance as described above, but also selecting the optimal hyperparameters that define the properties of the models. This involved a search through the space of possible hyperparameter vectors, all conducted via exploiting the cross-indexing algorithm. The proposed model selection approach was among the top-ranked in a summarized model selection competition. However, this type of more extensive approach is beyond the scope of this study whose emphasis is on feature selection.

[10] The generalized cross-indexing algorithm is detailed later in the section 3.1.4.

[11] $1 < N < K$

2.2 Content-based Analysis of Emotions in MIR

Lu et al. (2006) identified four issues that need to be considered in order to build a good computational model for mood detection from acoustic signals. The first issue relates to the fact that emotional expression and perception is subjective, depending on factors such as culture, education and personal experience. However, relying on previous research, Lu et al. (2006) stated that emotions are inherently expressed in certain musical structures or patterns, which is reflected on the major agreements between subjects on the elicited emotions in music between subjects. The agreement is the basis for the hypothesis that it is possible to build models for the recognition of affective content in music. However, subjectivity possibly sets boundaries for the hypothetically maximal performance of the models. The issue of subjectivity will be touched upon in the section 2.2.1 in relation to the aforementioned hypothesis and in the section 2.2.2 in relation to gathering ground truth for computational analysis. The second issue relates to how emotions are conceptualized. This is discussed in the section 2.2.2 which introduces different emotion models and reviews the research on musical emotions. The third issue relates to the acoustical features that are used as an input for the emotion recognition system. The features are extracted from the acoustical signal in a process called musical feature extraction which is explored in the section 2.2.1. The fourth issue relates to the actual framework used for studying recognition of affective content in music. The previous sections explored the background of the issue from the viewpoint of the general research on machine learning and discussed the problems that stem from the utilization of its techniques. The section 2.2.3 reviews the state of the research on emotion and mood recognition in MIR and the section 2.2.4 reviews how the problems of overfitting in the wrapper selection approach have been taken into account in the field.

2.2.1 Musical Features

The computational approach to classification of musical audio according to high-level perceptual categories such as emotion or genre is highly dependent on the representation of the input data given to the learning method. Digital audio itself cannot be used as an input since its dimensionality (ranging from 22500 to 48000 per second) is prohibitive for machine learning and it does not represent any relevant information explicitly. Feature extraction from musical audio, or audio content analysis, refers to the process that aims at representing audio data in reduced amount of dimensions, with perceptually relevant musical features.

Musical features represent information about musical cues that can be perceived in human brain. These cues relate for example to dynamics, timbre, rhythm, pitch, harmony and structure. Gabrielsson and Lindström (2001) reviewed the empirical research concerning the importance of musical cues as factors on perceived emotional expressions in music and gave clear evidence that the relation exists. However, musical structure cannot explain the whole mechanism that influences the perception of music. Meyer (1956, p.3) examined musical meanings "which result from the response to relationships inherent in the musical progress" while agreed that meanings can also be communicated by the "relationships between the musical organization and the extra-musical world of concepts, actions, characters and situations". These referential meanings can be attributed for example to culturally specific learning or individual experiences. Due to these aspects, any learning algorithm that uses exclusively musical features as input to predict perceptual concepts need to deal with fundamental limitations

Musical feature extraction is a particularly new research field with its origins in speech recognition (Hunt et al., 1980). Driven by the need to apply the techniques of speech recognition in real-world multimedia domains, Scheirer and Slaney (1997) studied 12 features related to timbre and articulation as well as one novel feature related to rhythm in speech/music discrimination task. Tzanetakis and Cook (2002) developed a highly advanced and influential framework for the purposes of musical genre classification. The authors extracted features relating to timbre, pitch and rhythm. The timbral feature vector included 19 features based on the studies by Hunt et al. (1980) and Scheirer and Slaney (1997) and was augmented with Mel-frequency Cepstral Coefficients (Logan, 2000). Pitch- and rhythm-related feature vectors comprised of 6 and 5 features, respectively.

Musical features are typically extracted in a frame-based approach to account for the evolving nature of music (Wold et al., 1996; Tzanetakis and Cook, 2002). The frame-based approach consists of first dividing an audio file into overlapping windows and then extracting features from each window separately. For timbre-related features the frame length is typically small ranging from 23 to 50 ms with 50% or 0% overlapping (Tzanetakis and Cook, 2002; Lu et al., 2006; Aucouturier and Pachet, 2004). Some features related for example to rhythm require longer frames from 1 to 2 seconds and if a feature can be considered as static within the analyzed excerpt, a feature can be computed over the whole file without splitting into frames (Tzanetakis and Cook, 2002). The feature values computed in the frames can be then summarized in order to obtain an overall description of the features for example by taking statistical means and variances of the features such as in (Tzanetakis and Cook, 2002). This type of summarization is referred to as the bag-of-frames approach in analogy with the bag-of-words representation used in text classification (Aucouturier et al., 2007). The bag-of-frames remains the main technique in music classification even though the study by

Aucouturier et al. indicated that the technique may not be able to sufficiently take account the level of the cognitive processes involved in music perception.

The most typical musical features are summarized next:

Category	Feature	Description
Dynamics	RMS	Root-Mean-Square energy of a signal estimating the perceptual loudness.
	Low energy ratio	The percentage of frames showing less-than-average RMS energy (Tzanetakis and Cook, 2002).
	Event density	Summarization of the information about note onsets represented as the number of onsets per second (Klapuri, 1999).
	Attack time	Estimation of the temporal duration of the attack phase of the onsets (Collins, 2006).
	Attack slope	Estimation of the slopes of the attack phase. There is evidence that attack slope is the key factor in the perception of the attack phase (Gordon, 1987).
Rhythm	Fluctuation	Information on the strength and density of beats played within the respective frequency bands (Pampalk et al., 2002). The periodicities in the bands are weighted according to the psychoacoustic model of fluctuation strength (Fastl, 1982). The spectra can then be summed to obtain an overall representation.
	Tempo	Estimation of the periodicity of the onsets in frequency subbands (Alonso et al., 2003).
	Pulse clarity	The strength of the beats estimated by a tempo detection function (Lartillot et al., 2008).
Pitch	Pitch	Pitches are extracted by a multipitch detection technique such as in (Tolonen and Karjalainen, 2000) and their frequencies are returned.
	Chromagram	Broad description of the localization of pitch energy along pitch classes (wrapped) (Eerola et al., 2009) or pitches (unwrapped) (Pauws, 2004).
Harmony	Key clarity	Strength of the most probable key. Key strength can be represented as the probability associated with each possible key candidate. For example Gómez (2006) computed the key strength by taking the cross-correlation between the extracted chromagram and the key profiles by Krumhansl (1990).
	Mode	Modality, i.e. a measure of how major or minor music is. Modality can be derived from the key strengths (Gómez, 2006).
	HCDF	The Harmonic Change Detection Function computed as the flux of the tonal centroid (Harte et al., 2006).

Category	**Feature**	**Description**
	Roughness	Roughness can be computed as the sensory dissonance related to the beating phenomenon whenever a pair of sinusoids are close in frequency. Sethares (1998) proposed a method for estimating roughness by taking the average of all pairs of peaks in the spectrum while Leman (2000) proposed a Synchronization Index Model (SIM) that is based on auditory processing.
	Inharmonicity	The degree in which the harmonic partials deviate from the multiples of the fundamental frequency. Inharmonicity can be estimated for example by the algorithm developed by Rauhala et al. (2007) (for single tones). The extraction of the feature from polyphonic music is extremely challenging.
Timbre	Brightness	High-frequency energy computed as the relative proportion of sound energy above a certain cutoff frequency (Juslin, 2000).
	Rolloff	Brightness-related feature computed as the frequency below which a certain proportion of the sound energy is concentrated (Tzanetakis and Cook, 2002).
	Spectral centroid	The statistical mean of the spectral distribution, optionally averaged over centroids computed in separate frequency sub-bands (Tzanetakis and Cook, 2002).
	Spread	Statistical standard deviation of the spectral distribution, optionally averaged over spreads computed in separate frequency sub-bands (Lidy and Rauber, 2006).
	Skewness	The coefficient of the skewness of the spectral distribution, optionally averaged over skewness measures computed in separate frequency sub-bands (Lidy and Rauber, 2006).
	Spectral entropy	Entropy of the spectral distribution, optionally averaged over spectral entropies computed in separate frequency sub-bands (Toh et al., 2005).
	Flatness	Statistical measure indicating whether the spectral distribution is smooth or spiky, optionally averaged over flatness measures computed in separate frequency sub-bands (Izmirli, 2000).
	Zerocross	Time-domain measure of noisiness of a signal consisting of the number of times the signal changes sign (Tzanetakis and Cook, 2002).
	Spectral flux	Measure of the amount of local spectral change computed as the distance between the spectrum of each successive frames (Tzanetakis and Cook, 2002).
	Regularity	The degree of variation of the successive peaks of the spectrum computed as the log of the spectral deviation of component amplitudes from a global spectral envelope derived from a running mean of the amplitudes of three adjacent harmonics (Krimphoff et al., 1994) or as the square sum of the difference in amplitude between adjoining harmonic partials (Jensen, 1999).

Category	Feature	Description
	MfCC	Mel-frequency Cepstral Coefficients (Logan, 2000) computed in several steps including grouping and smoothing of the magnitude spectrum according to the perceptually motivated mel-frequency scale and taking the Discrete Cosine Transform (DFT) of the smoothed sectrum. Also time-domain derivatives of MfCCs (dMfCC) can be used.
Structure	Repetition	Repetitiveness of a feature such as spectrum or chromagram can be estimated through the computation of novelty curve from the similarity matrix that contains similarities between all possible pairs of frames of the given feature (Cooper and Foote., 2001).

2.2.2 Musical Emotions

Two main approaches for conceptualizing emotions – categorical and dimensional – have emerged in psychology research. The categorical approach (Ekman, 1992) distinguishes a small number of *basic* emotions that most clearly differ from one another such as anger, fear, sadness, happiness, surprise and disgust. These basic emotions are supposed to have developed in the course of evolution for dealing with fundamental life-tasks. This makes them innate and universal. The approach also postulates that other non-basic emotions can be considered as combinations of the basic emotions.

By contrast, the dimensional approach (originally formulated by Wundt (1897)) supposes that emotions can be represented in two or three independent bipolar dimensions such as valence, arousal and tension, each dimension relating to separate neurophysiological systems. The set of theories is augmented by affective state or mood models. The approach is mainly based on the prevalence of these moods in every-day life. Therefore there have been attempts to gain theoretical background by relating the moods to the models of emotions. For example the prototype approach referred in Sloboda and Juslin (2001) suggests a tree-like structure in which that affective states can be categorized in terms of resemblance to a particular basic emotion.

Russell (1980) proposed a circular structure of affective states placed on the dimensions of activation and valence. It was suggested that the structure imposes a similarity measure between emotions, such that two emotions close in the circle are similar. Conversely, emotions across the circle from one another correlate inversely and may be considered as bipolar. However, there has been disagreement over the exact dimensions in the structure. Thayer (1989) suggested that energetic arousal and tense arousal should be the basis for the two-dimensional model and that the plane could be divided into four quadrants that represent the different general mood states.

While the aforementioned models have been used in studying emotions in music, there have been arguments against applying the models from the non-musical research to the music psychology research postulating that the models cannot account for musical emotions. Zentner et al. (2008) conducted four studies which aimed at identifying what emotions are related to musical experiences, how music-related emotions differ from emotions experienced in every-day life and how perceived musical emotions differ form felt musical emotions. The final goal was to derive a structural model that could be used specifically for studying music-induced emotions. The material was obtained from settings where participants rated and evaluated emotions in music excerpts. Based on the occurrences of affective terms used to describe emotions, difference in the occurrences of affective terms indicated that the variability of emotional experiences in every-day life differ significantly from the variability of emotions experienced in the context of music. Therefore the need for a model for musical emotions is well justified. Also the ratings of perceived emotions differed significantly from ratings of felt emotions in music and emotion ratings differed significantly as a function of musical genre. Specifically, emotions were more often perceived than felt in the context of music. For example negative emotions (e.g. fear, sadness, anger-irritation) were perceived as expressive properties of music but rarely felt in response to music. Also other studies have reported same type of findings (Zentner et al., 2000; Juslin and Laukka, 2004). The main output of the four studies by Zentner et al. was the structural model called GEMS (Geneva Emotional Music Scale) which is essentially a prototype model for musically induced emotions. The lowest level of the model contains 40 emotion labels that are often used to describe emotional experiences in relation to music. The middle level obtained from factor analysis of the 40 labels contains 9 first-order factors, namely wonder, transcendence, tenderness, nostalgia, peacefulness, power, joyful activation, tension and sadness. The nine factors are further factorized into three dimensions of sublimity, vitality and unease.

Similarly, Leman et al. (2005) conducted factorial analysis of 15 bipolar adjective pairs gathered from literature and trial experiments and identified three underlying affective dimensions. Hu et al. (2007) took another approach for obtaining a set of mood labels used for describing musical emotions or moods. They analyzed the tags in the Last.fm web service. Using the top 100 tags ranked in terms of the number of occurrences in the USPOP genre they identified 19 tags that were related to mood. A cluster analysis of the tags yielded three clusters relating to happiness, aggressiveness and calmness. Furthermore, Hu and Downie (2007) analyzed mood metadata in several web services including the AllMusic.com[12] and

[12] www.allmusic.com

clustered the most popular moods into 5 categories.

An important step towards musical emotion/mood classification or prediction is the collection of the ground truth set, i.e. a set of music excerpts that are annotated in terms of the selected emotion/mood taxonomy. Li and Ogihara (2003) noted that low performance of their classification system can be partially attributed to the mood taxonomy since the ground truth contained numerous cases where the labeler had found it difficult to make decision between the mood label of the excerpt. The number of labels in the experiment was 13. Skowronek et al. (2006) tried to overcome the issue by using four labelers. They analyzed the assessments in terms of consistency between subjects as well as importance and easiness of the mood labels according to the labelers. This resulted in a reduction of the mood label space into 8 labels that were both above the "acceptable consistency" and important/easy.

Eerola and Vuoskoski (2009) collected two ground truth sets of 360 and 110 excerpts rated according to the intensity of the expressions of each basic emotion and according to the expressed emotion in the 3-dimensional space of valence, energy and tension. The ratings of the emotion models were given in scales by a number of participants. The research setting allowed for a detailed analysis of the consistencies and possible similarities of the models. Although the categorical and dimensional models are commonly thought as inherently different, the results indicated that the models are highly compatible and yield similarly consistent results. Moreover, certain confusions were found within the models. In the 3-dimensional model tension and valence correlated negatively in a highly significant degree which indicated that these two dimensions are not independent in the context of perceived emotions in music and that a 2-dimensional model could be used instead. In the basic emotion model anger and fear were confused to a degree that indicated that these concepts might not be easily distinguishable in the musical domain.

In addition, exploring the metadata in the web services as described above provides an interesting option for collecting ground truth for classification.

2.2.3 Emotion and Mood Recognition

Varying approaches have been implemented for the task of emotion and mood recognition. Variation is partly attributed to the differing models for describing the emotion and mood taxonomies. Reviewing the research in the field reveals that the mood models have been used predominantly over the emotion models in MIR. This has been reflected in the annual Audio Music Mood Classification task organized by MIREX – the ground truth in the task is based on five mood clusters (Hu and Downie, 2007). To enable straight comparison of the submitted systems the classification in the task is conducted on a collectively agreed large

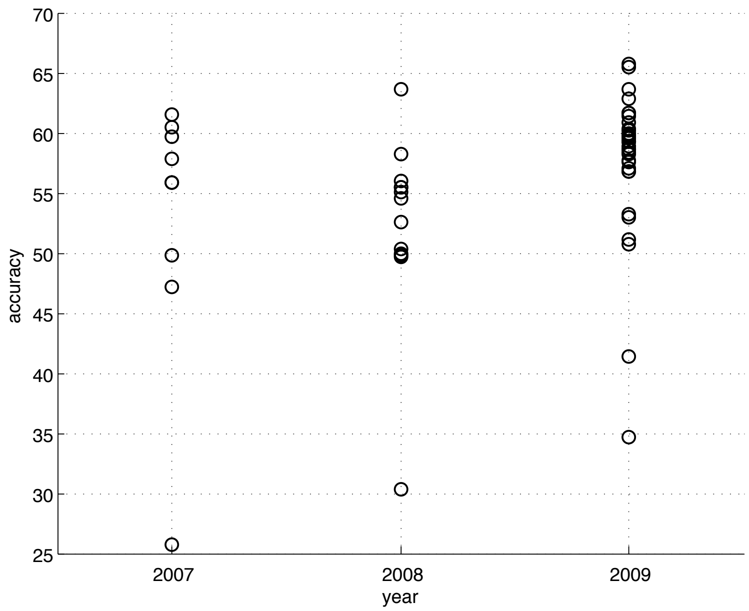


Figure 2.3: Classification accuracies of the submitted models in the annual MIREX Audio Music Mood Classification task.

ground truth set and valid performance measures are obtained with 3-fold cross-validation (Hu et al., 2008). The results of each years task are shown in the MIREX Wiki page[13]. The figure 2.3 summarizing the classification accuracies indicates a slightly increasing trend in the performance of the systems, especially in the performance of the best performing ones, and a clear increase in the number of submitted systems. The 65% accuracy in music classification can be considered rather high considering the estimated class-ceiling performance of around 70% in music classification based on timbre similarity (Aucouturier and Pachet, 2004).

An overview of the first year's systems was provided by Hu et al. (2008). It was notable that six of the nine systems adopted support vector machine as a classifier. There was large variation in the used feature sets, all systems however used spectral features. In fact, the highest performing system by Tzanetakis (2007) used spectral features exclusively. In the year 2008 the system with the highest performance by Peeters (2008) used MfCC, spectral and chromagram features. Prior to classification, the dimensionality of the feature set was reduced in two stages. First feature ranking was used to retain the top 40 features and then linear discriminant analysis was applied to the reduced feature set. The resulting set was classified with a modified Gaussian Mixture Model (GMM) classifier. In the year 2009 the winning system by Cao and Li (2009) used again support vector machine. This time the feature set included features such as MfCCs and rhythm pattern features. As a pre-processing

[13] `http://www.music-ir.org/mirex/20xx/index.php/Audio_Music_Mood_Classification`

stage, the data was transformed according to Gaussian Mixture Models derived derived from a large separate dataset.

Comparing the results obtained in the AMC evaluations with previous research into mood recognition is problematic due to fundamental differences in the ground truth, performance evaluation and reporting. The figure 2.3 showed that the average high performance of the systems is around 60% in a collectively controlled framework. Nevertheless, the previous studies have reported drastically higher performances when the researchers have been able to build the framework from the outset. In one of the first studies of mood recognition, Li and Ogihara (2003) used 13 moods as separate labels and extracted 30 features suggested by Tzanetakis and Cook (2002) relating to timbre, rhythm and pitch. They used support vector machine as classifier which resulted in accuracies from 50% to 80% depending on the mood category. The mood were also clustered manually into 6 categories but the results for these clusters were not reported. Wieczorkowska et al. (2005) analyzed the same collection using the prompted 6 clusters, k-NN classifier with optimized k-values and 29 features relating to timbre and chords. The results showed correct classification rates ranging from 64% to 96%. The results of both studies were high taking account the rather large collection of 499 excerpts but their generalizability can be questioned since the analysis was conducted only for a single 50%-50% split into train and test sets.

Lu et al. (2006) studied mood detection from classical music. The mood taxonomy was based on the 2-dimensional model by Thayer (1989) divided into four quadrants to be used as classes. 25 features relating to timbre, 5 features relating to rhythm and an intensity feature computed in several subbands were extracted. Using GMM as classifier the authors compared the traditional approach of using the learning algorithm on a full feature set and a hierarchical framework that consisted of first building a model with intensity feature to obtain separation regarding the valence axis and then building four models with either timbre or rhythm feature sets to separate the excerpts with positive and negative valence regarding the arousal axis. The hierarchical framework enabled studying the importance of timbre and rhythm features by giving varying weights for the feature sets in the second stage of the classification process. Classification was conducted 10 times with random 75%-25% splits of the data and the final performance was obtained by averaging the results. The classification accuracy of the traditional approach was 80.6% and the highest accuracy of the hierarchical approach was 86.3%. The hierarchical approach was optimized in by varying the weights given for timbre and rhythm features. The optimal accuracy was obtained by giving more weight for timbre features which indicated that timbre features were more important in mood recognition than rhythm features. Moreover, using timbre features alone resulted in higher accuracy (77.8%) than using rhythm features alone (70.6%).

One of the few studies using the basic emotion model in music classification is by Feng et al. (2003) who studied the effect of tempo (relating to fast vs. slow) and mean and standard deviation of articulation (relating to staccato vs. legato) on classification of modern popular music into basic emotion categories happy, sad, anger and fear (the authors referred to the basic emotions by the term "mood"). The results suggested that tempo and articulation are important features in determining the emotion of music as a neural network classifier yielded a relatively high total recall rate of 66% and total precision rate of 67% when 200 excerpts were used for training and 23 excerpts were used for testing.

Other studies have used the emotion models in prediction problems rather than in classification problems. The detailed ratings of basic emotions in the soundtrack collection gathered and annotated in the study by Eerola and Vuoskoski (2009) enabled Eerola et al. (2009) to use prediction models for analyzing the material. The study concentrated on dimension reduction by applying multivariate regression models for prediction of the ratings. Also the ratings given in terms of the 3-dimensional model were analyzed. Prior to applying the regressors the dimensionality of the feature set was reduced by minimizing redundancy within each feature category (dynamics, rhythm, harmony etc.). This resulted in a set of 29 features whose within-feature correlation was lower than .30. The prediction rates of the Partial Least Squares (PLS) regressor with only two predictors were the highest for predicting the ratings in both emotion models. Prediction of the ratings of basic emotions yielded rates ranging from 58% to 74% depending on the emotion category. The prediction rates for the 3-dimensional model ranged from 70% to 77%. The results of the dimensional model were improvement to previous research such as the study by Yang et al. (2008) that yielded 58.3% prediction rate for arousal and 28.1% for valence with Support Vector Regression.

2.2.4 Overfitting in Wrapper Selection

In the field of machine learning, classification systems enabling wrapper selection as a preprocessing phase has been acknowledged to be prone to leading to overfitting (see the section 2.1.4). Due to overfitting, especially cross-validation accuracies used in the selection process are highly biased performance estimates and they should not be used when evaluating classification methods (Reunanen, 2004). However, the issue has not been given enough consideration in the studies in MIR. In fact, reviewing the few studies in MIR using wrapper selection reveals possibly inadequate frameworks for obtaining performance estimates for the used methods. For example, Silla Jr et al. (2008) used wrapper selection with Genetic Algorithm and compared the performance of the Decision Tree, 3-NN, Naive Bayes, Multi-layer

Perceptron and Support Vector Machine in genre classification. The results were reported to indicate that wrapper selection procedure is effective for Decision Tree, 3-NN and Naive Bayes but the authors did not base their arguments on classification performance on independent test sets. This may have led to positively biased results given that overfitting in wrapper selection with genetic algorithms is an acute problem (Loughrey and Cunningham, 2005).

Similar conclusions can be drawn from an emotion recognition study by Yang et al. (2006). The authors divided the arousal-valence plane into four quadrants as suggested by Thayer (1989) and represented the quadrants as classes. Two classifiers enabling fuzzy logic (Keller, 1985; Tran et al., 1999) were used in emotion recognition from segments of music. The chosen feature selection approach was the wrapper with backward elimination using the cross-validation accuracy (90%-10% splits repeated 50 times) to assess the performance of the method. The results showed increasing performance with both classifiers with the optimal subset size. However, the found effects might have been optimistically biased since the final performance assessment and reporting of the results was based only on cross-validation. This casts doubt upon the validity of the results.

On the other hand, Yaslan and Cataltepe (2006) studied wrapper selection in genre classification using forward selection and backward elimination with ten different classifiers. The dataset was first split randomly into train and test sets constituting of 90% and 10% of the excerpts, respectively. Then, wrapper selection was run with the train set and finally the test set accuracies with different subset sizes were reported. However, since the whole process was run only once, the results might have been somewhat random. Using the external loop of cross-validation would have improved the validity to some extent.

In a single account of overfitting in feature selection in MIR, Fiebrink and Fujinaga (2006) addressed the problem by re-evaluating the claims of the authors' previous study (Fiebrink et al., 2005). In that study feature weighting with genetic algorithm was found to yield performance improvement in timbre recognition from snare-drum attack sounds and beat-box sounds using k-Nearest Neighbors classifier. Performance assessment and evaluation of the results in (Fiebrink et al., 2005) was based solely on cross-validation accuracies obtained in the selection process while the used frame-work in (Fiebrink and Fujinaga, 2006) was based on the guidelines given by Reunanen (2003, 2004) in order to study the effect of overfitting due to feature selection. The framework for the re-evaluation incorporated the external loop of cross-validation and concentrated on wrapper selection with k-Nearest Neighbors and forward selection search strategy with the same datasets as in (Fiebrink et al., 2005). The results showed a significant effect of overfitting as indicated by the high performance on the train set compared to the low performance on the independent test sets in both datasets. Moreover, the performance improvement of the selected feature sets when compared to full sets provided little or no benefit. To gain further evidence on the usability of wrapper selection, the framework was evaluated in a genre classification task. Also Principal Component Analysis was used for dimensionality reduction for comparison purposes. In genre classification, wrapper selection provided significant improvement in terms of test set accuracies when compared to classification without feature selection but PCA yielded similar increase in accuracy in a fraction of computation time. In the conclusions the authors pointed out the importance of a sound framework for studying the benefits of feature selection in MIR problems to gain understanding about the topic under study.

Chapter 3

Analysis

3.1 Method

The purpose of this study was to test the behaviors and benefits of feature selection, namely wrapper selection, on a task of classification of musical audio according to the expressed emotion. The analysis was conducted in a comparative manner to obtain information about different selection methods. For the sake of convenience, the analysis was split into two experiments. In the experiment 1 all selection methods under analysis were used. The most promising ones in the experiment 1 were chosen for the experiment 2 whose aim was to give more reliable information about the performance of these methods.

The overall setup of the analysis is illustrated in the figure 3.1. The choice of the framework was based on cross-indexing (see the section 2.1.4) in order to avoid overfitting. Given the pair of learning method and search method, the *cross-indexing loop* involves running the wrapper selection K times. At each run the selection is done on the train set and evaluated primarily on the test set. The train and test sets are obtained by splitting the primary set in a stratified manner with a random number seed specific to each run.

The comparison of the methods was done via the measures obtained by cross-indexing, namely the obtained optimal subset size and its performance. The secondary measures relate to the averaged values computed to demonstrate the effects of overfitting in the data in the experiment 1 (averaged CV-classification accuracies) and validation of the results in the experiment 2 (averaged classification accuracies on the validation set). In addition to the classification accuracies, the results in the experiment 2 were analyzed in a more detailed level for example to show how the different features contributed to the models (see the section 3.1.5) and to estimate to what degree the misclassification rates could be attributed to the ambiguousness of expressed emotions in terms of human perception (see the section 3.1.6).

The next subsections give a detailed information related to the objects in the figure.

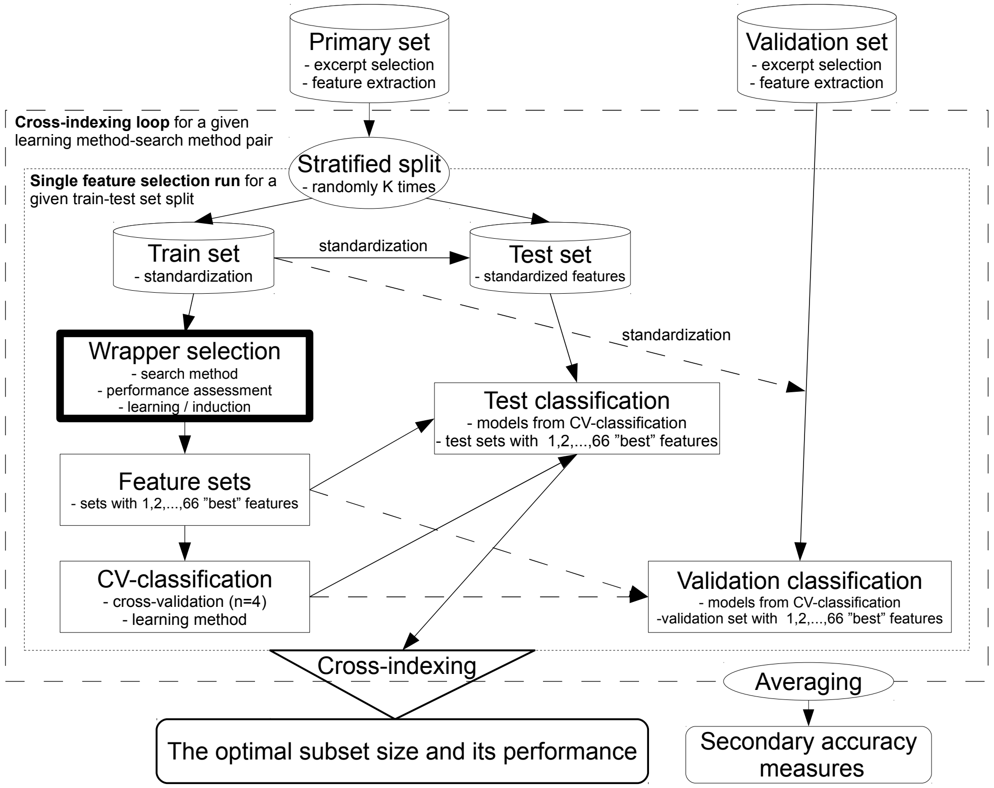


Figure 3.1: A chart describing the analysis framework.

3.1.1 Pre-Processing the Data for Feature Selection and Classification

This section describes how the primary and validation *s*et were obtained and how they were preprocessed for the purposes of the analysis. The sets are summarized in the table 3.1.

Set	**Features**	**Classes**	**Excerpts**
Primary set	66	*anger_fear, happy, sad, tender*	64
Validation set	66	*anger_fear, happy, sad, tender*	160

Table 3.1: Summary of the datasets.

3.1.1.1 The Audio Material and Its Annotation

The musical material analyzed in this study was collected and annotated in an earlier study that was conducted at the Music Department of the University of Jyväskylä (Eerola and Vuoskoski, 2009). The authors used film soundtracks as stimulus material in experiments that involved participants rating the excerpts according to the expressed emotion. The film soundtracks was chosen as the genre of the stimuli because film music in general was presumed to be "composed with an intention to mediate emotions and could serve as a relatively 'neutral' stimulus material in terms of music preference and exposure" (Eerola and Vuoskoski, 2009, p.4). The material was collected by 12 musicology experts.

The participants rated the musical excerpts both by discrete basic emotions – each emotion (*anger, fear, happy, sad* and *tender*) on a scale from 1 to 7 and in a three-dimensional emotion space, each axis (*valence, energy* and *tension*) on a scale from 1 to 9. In the first experiment 360 excerpts with length from 10 to 30 seconds were rated by 12 musicology experts. After analyzing the ratings, 110 of the pieces were chosen as stimuli for the second experiment. The amount of how congruently and high the excerpts were rated in terms of each discrete emotion and dimension was used as a criterion for the selection, favoring strong and distinct emotion content. In the second experiment 116 students – mostly non-musicians – rated the 110 excerpts in a similar manner as in the first experiment.

This study concentrates only on the ratings of basic emotions. In the study of Eerola and Vuoskoski (2009), analysis of correlations between the ratings of basic emotions suggested that emotions *anger* and *fear* might not be easily distinguishable in the context of music. In both of the experiments *anger* and *fear* correlated highly and the second experiment showed correlation $r = .69$ which was significant at the confidence level $p < .001$. This indication led to unification of these emotions into one emotion in the present study.

First, the annotated material was adapted for classification purposes by assigning the highest rated emotion of each musical excerpt as the single representative emotion label. Then the labels *anger* and *fear* were merged by assigning the maximum of these both categories as the mean of the label *anger_fear*.

To sum up, the material in the present study consists of two sets of film soundtrack excerpts which are assigned with emotion labels *anger_fear, happy, sad* and *tender*.

3.1.1.2 Selecting Applicable Excerpts

Before further analysis, the excerpts from the experiment 1 and 2 in the study by Eerola and Vuoskoski (2009) were evaluated according to their applicability for classification purposes. The aim was to omit excerpts whose expressed emotions had been the most ambiguous according to the participants' ratings. Therefore only excerpts with distinct and strong expressed emotion were preserved. The selection was based on a rationale that it would not have been reasonable to classify an excerpt into one emotional category if another emotion would have been nearly as representative. Another option to overcome this issue would have been multi-class classification that is out of the scope of this study.

A selection measure for each excerpt was computed by dividing the mean rating of the highest rated emotion category by the mean rating of the emotion that was rated second highest.[1] Then, for each emotion, a list was formed which contained all excerpts that were assigned to that particular emotion. The list was sorted in a descending order by the selection measure. Those samples that expressed one clear emotion while other emotions were relatively low ended up at the top of the list while the ones with the most confusion in the ratings of the expressed emotions ended up at the bottom. To treat each emotion with equal importance in the evaluation the same amount of excerpts in all emotion categories was preserved. The number of excerpts to be preserved was a trade-off between the number of excerpts and the unambiguousness of the emotions.

The primary set used in the analysis of this study was selected from the set of 110 samples since it was rated by a large group of participants. The chosen number of excerpts belonging to each emotion was 16 which constituted a set of 64 excerpts.

The validation set was selected from the collection of 360 excerpts. Before computing the selection measures the excerpts which overlapped with the primary set were removed. Excerpt selection was then conducted with this reduced set in a similar manner as above except that the number of each emotion to maintain was 40.

3.1.1.3 Feature Extraction

Purpose of audio-related feature extraction in general is to reduce the huge amount of data in audio files into perceptually or analytically meaningful representations that can be subjected to statistical analysis. In this study extraction of musical features was done with *MIRtoolbox*[2] (Lartillot and Toiviainen, 2007), an integrated set of functions dedicated to feature extraction from audio files in *MATLAB*. The main characteristic of the toolbox is its capability to extract a wide range of features relating to dynamics, rhythm, harmony, timbre, pitch, tonality and structure. The versatility is achieved by a modular design where the extraction algorithms are decomposed into stages. This has enabled the exploitation of functions in several public-

[1] The used selection measure is a simple version of the measure used in Eerola and Vuoskoski (2009). Its basis is in intuition that an excerpt that expresses an unambiguous emotion should have been rated high in terms of one emotion and relatively low in terms of all other emotions.

[2] MIRtoolbox is available from `www.jyu.fi/music/coe/material/mirtoolbox`.

domain toolboxes such as *Auditory Toolbox* (Slaney, 1998), *NetLab* (Nabney, 2002) and *SOMtoolbox* (Vesanto, 1999).

The present study exploited the version 1.1.17 of the *MIRtoolbox*. Because of the computationally demanding wrapper selection the number of the extracted features was reduced from a total of about 400 possible features to 66 features, presented in the table 3.2. This reduction was based on pre-existing (unpublished) analysis of a wide collection of audio examples conducted at the Music Department of the University of Jyväskylä (similarly to that in (Eerola et al., 2009)), where feature categories, e.g. timbre and rhythm, were analyzed separately to detect non-redundant features within the categories ($r < .30$). Reduction resulted in a set of 52 features (features $1-52$ in the table 3.2). To obtain a more multifaceted feature set a selection of features relating to mel-frequency cepstral coefficients (features $53 - 66$) was added to the set.

The extraction was done with the frame-based approach where statistical measures such as means and standard deviations were computed over their values in small overlapping analysis windows that covered the whole excerpts (see the section 2.2.1). For most of the features, the used frame length of the analysis window was 46 ms with 50% overlap. For low energy ratio , high-level features that require longer frame length (fluctuation, harmony-related features) as well as tempo and pulse clarity the analysis frame was 2 seconds with 50% overlap whereas the structure-related features were computed with frame length of 100 ms and 50% overlap. The values of frame length and overlap were based on the aforementioned pre-existing analysis. In the case of MfCC features the window was the same as with most other low-level features, i.e. 46 ms with 50% overlap. The commands for extracting the features with MIRtoolbox are detailed in the Appendix A.

3.1.1.4 Stratified Splitting

In order to run the cross-indexing loop several times, at each run the primary set was split in a random stratified manner into train and test sets. Stratified splitting was used to create random equal-sized subsets of samples for analysis while maintaining the relative numbers of excerpts expressing each emotion.[3] At each of the K iterations, different random number seeds were used for shuffling the dataset.

[3] In (Reunanen, 2006, p. 205) a parameter f was used to control the size of the train and test sets – the number of training samples is the total number of samples divided by f and the number of test samples is the number of training samples multiplied by the value $f-1$. Generally, with small to medium sized datasets a value $f = 2$ is preferable and with more data available, bigger f values can be used. In the present study the value $f = 2$ was used, corresponding to splitting into equal-sized train and test sets.

Category	No.	Feature
Dynamics	1	RMS (M)
	2	RMS (SD)
	3	RMS (SL)
	4	Low energy ratio
	5	Attack time (M)
	6	Attack slope (M)
	7	Attack slope (SD)
Rhythm	8	Event density
	9	Fluctuation peak position (M)
	10	Fluctuation peak magnitude (M)
	11	Fluctuation centroid (M)
	12	Tempo (M)
	13	Tempo (SD)
	14	Pulse clarity (M)
	15	Pulse clarity (SD)
Pitch	16	Pitch (M)
	17	Pitch (SD)
	18	Chromagram (unwrapped) centroid (M)
	19	Chromagram (unwrapped) centroid (SD)
	20	Chromagram (unwrapped) centroid (SL)
	21	Chromagram (unwrapped) centroid (H)
Harmony	22	Key clarity (M)
	23	Key clarity (SD)
	24	Key mode (majorness) (M)
	25	Key mode (SD)
	26	HCDF (M)
	27	Entropy of the octave collapsed spectrum (M)
	28	Roughness (M)
	29	Inharmonicity (M)
	30	Inharmonicity (SD)
Timbre (Centr.)	31	Brightness (cut-off 110 Hz) (M)
	32	Brightness (cut-off 110 Hz) (SD)
	33	Spectral Centroid (M)
	34	Spectral Centroid (SD)
	35	Zerocross (M)
	36	Zerocross (SD)
Timbre (Shape)	37	Spread (M)
	38	Skewness (M)
	39	Spectral entropy (M)
	40	Spectral entropy (SD)
	41	Spectral flux (M)
	42	Flatness (M)
	43	Regularity (M)
	44	Regularity (SD)
Structure	45	Repetition (spectrum, M)
	46	Repetition (spectrum, SD)
	47	Repetition (rhythm, M)
	48	Repetition (rhythm, SD)
	49	Repetition (tonality, M)
	50	Repetition (tonality, SD)
	51	Repetition (register, M)
	52	Repetition (register, SD)
Mel-frequency Cepstral Coefficients	53	1st MfCC (M)
	54	1st delta MfCC (M)
	55	2nd MfCC (M)
	56	2nd delta MfCC (M)
	57	3th MfCC (M)
	58	3rd delta MfCC (M)
	59	4th MfCC (M)
	60	4th delta MfCC (M)
	61	5th MfCC (M)
	62	5th delta MfCC (M)
	63	6th MfCC (M)
	64	6th delta MfCC (M)
	65	7th MfCC (M)
	66	7th delta MfCC (M)

Table 3.2: Extracted feature set. M = Mean, SD = Standard deviation, SL = Slope, H = Entropy.

3.1.1.5 Standardization

After splitting the primary set into train and test set, the train set was standardized at each run to give each feature initially the same importance in the feature selection process. The features in the test set as well as in the validation set were transformed to the same scale as the standardized train set. The transformation was done via:

$$\hat{z} \Rightarrow \frac{\hat{z} - \mu}{\sigma}, \tag{3.1}$$

where $\hat{z}$ is a feature vector in the test or validation set and μ and σ are the mean and standard deviation of the particular feature in the train set, respectively.

3.1.2 Feature Selection

Feature selection in this study concentrated on the wrapper approach described in the section 2.1.3. Because feature subsets found with wrapper selection are prone to overfitting, the implementation of the process was done following the guidelines given in the papers by Reunanen (2003, 2004, 2006, 2007) described in the section 2.1.4.

The performance of each candidate subset was assessed by cross-validation[4] with 4 folds. This relatively low number of folds in cross-validation was chosen to limit the computation time as the number corresponds to the number of times the learning algorithm must be used to estimate the performance of a single candidate subset.

Feature selection was done in Weka[5] software (Witten and Frank, 2005). Weka is a collection of machine learning algorithms and data preprocessing tools written in Java and distributed under the terms of the GNU General Public License. The software offers both graphical user interface for data processing and visualization as well as a possibility to use Weka via scripts or Java code. In this study the algorithms of Weka were used from code written in the Shell scripting language. This allowed for a large controlled experiment in which several methods could be tested.

Weka implements the wrapper selection in the function *WrapperSubsetEval*. The function allows choosing the learning and search methods used in selection as well as whether to use cross-validation (in this case the number of folds can be chosen) or a separate validation set to assess the performance of the candidate subsets. Weka offers implementations of a wide variety of learning and search methods that can be used in the selection. Next, the methods

[4] This refers to the inner cross-validation used in the selection process as opposed to the external cross-validation.

[5] Weka is available from `http://www.cs.waikato.ac.nz/ml/weka`.

that were used in this study are described.

3.1.2.1 Classifiers

The choice of the classifiers used in the wrapper selection is based on previous studies in musical emotion or mood classification. It was required that the used classifiers had been effective and widely used in the previous studies in the field. Thus, three classifiers were chosen: Naive Bayes, k-Nearest Neighbors and Support Vector Machine. These classifiers and their implementations in Weka are described next:

Naive Bayes

Naive Bayes (John and Langley, 1995; Han and Kamber, 2001, p. 297-299) is a probabilistic classifier that can predict class membership probabilities. The algorithm relies on two simplifying assumptions: First, predictive features are assumed to be conditionally independent given the class and second, no hidden or latent features are posited to influence the classification process. The first assumption is called the *class conditional independence*. Langley and Sage (1994) studied the assumption and considered it a serious limitation in classification of real-world data where correlations among features exist. An example was given from medical domain where "certain symptoms are more common among older patients than younger ones, regardless whether they are ill". Analysis of feature-based datasets in music show that there exist a high amount of correlation between the features. This suggests that several features can reflect the same background concept. Too much emphasis could therefore be laid upon this particular concept, reducing the influence of other features. In these type of datasets, dependencies between features potentially deteriorates the performance of the Naive Bayes classifier. Wrapper-based feature selection can therefore significantly improve the performance of the algorithm.

Moreover, Naive Bayes algorithm makes an assumption that continuous numeric features are normally distributed within each class. This is reflected in the representation of the input features within the algorithm as it treats numeric features by their means and standard deviations. Probabilities that are used in the learning process are then computed from these values. However, continuous features in real-world data-sets usually violate the normality assumption. In these cases, as the number of attributes grows, performance of Naive Bayes weakens. To overcome this weakness, John and Langley (1995) developed a modified version of the algorithm which replaces single Gaussians representing the attributes in each class with *Gaussian kernels*, calling the method the *Flexible Bayes* learning algorithm. In Flexible

Bayes the estimated distributions are averaged over a large set of kernels. For each attribute in each class, the set of kernels are Gaussians computed for each training instance.

In this study, *Flexible Bayes* modification was used in Weka. In Weka *Flexible Bayes* can be used by setting the parameter *UseKernelEstimator* in the Naive Bayes function to 'true'. Naive Bayes algorithm has been used in musical emotion and mood classification for example in the studies by Baum (2006) and Pohle et al. (2005).

K-Nearest Neighbors

K-Nearest Neighbors (k-NN) algorithm (Aha and Kibler, 1991; Han and Kamber, 2001, p. 314-315) is a common example of instance-based learning (IBL). Instance-based learning algorithms are considered as lazy as they delay the building of the classifier until a new sample needs to be classified. This contrasts with eager learning, which constructs a generalization model before classifying new instances.

In its basic form, the learning phase in IBL algorithms consists of simply saving the normalized feature values of all training instances. With k-NN, the classification phase is conducted for a given sample by calculating its pair-wise similarity with all training instances. The similarity is defined by a given similarity function, for example the additive inverse of the Euclidean distance. Given a new instance to be classified, its class membership is determined by the most common class of its k nearest neighbors in terms of pair-wise similarities. Because the computation is done in the classification phase rather than in learning, IBL algorithms are relatively fast at learning but slower at classification.

Nearest neighbor algorithms in general are susceptible to the curse of dimensionality. This is explained in the article by Friedman (1997). For an instance to be classified, the predicting region is defined to be the sub-region of the input-space containing its k nearest training instances. This formulation leads to a problem when the number of dimensions, n for example, in the input-space is large. Because of the geometry of the Euclidean spaces, the radius of the prediction region grows in the proportion of the nth root of the volume whereas the number of training points in the region varies linearly with the volume. Therefore, with large number of features, the variance of the similarities in the predicting regions is high to the proportions that can make the similarity measures misleading.

To overcome this problem, Friedman (1997) gave importance to the choice of the k-value to be used. A small k-value can reduce the growth of the volume of the predicting region while a big k-value can reduce the effect of noise in the data.

Also feature selection as a means to avoid the problem can be effective with the nearest neighbors classifiers. Because each feature alone is given the same weight in classification, redundant and irrelevant features can distort the performance of the classifier. An irrelevant feature introduces misleading bias to the similarities and redundant feature causes a particular background concept behind several features to dominate.

However, as pointed by Friedman (1997, p.60): “some simple highly biased procedures such as “naive” Bayes and nearest neighbor methods remain competitive with and sometimes outperform more sophisticated ones, even with moderate to large training samples”. Therefore the method is widely used in a large range of domains. In musical emotion and mood classification, k-NN has been used for example in the following studies: Trohidis et al. (2008); Wieczorkowska et al. (2005); Pohle et al. (2005); Wieczorkowska et al. (2006); Yang et al. (2006).

Preliminary analyzes in the present study implied that a value $k = 10$ is a good choice for the task at hand based on its performance with different subset sizes. The classifier was used in Weka (the classifier *IBk* under the category of lazy learners) with Euclidean distance as a similarity measure.

Support Vector Machine

Support vector machines (SVMs) are classifiers that are related to the margin classifiers in their special use of the margin representation and decision boundaries. One of the most extensive descriptions of the method is given by Vapnik (1998). In the primal form of the algorithm for the nominal classes, two classes are considered.[6] The task is to find a hyperplane, linear by a default, that separates the classes with a maximal margin. This margin is called the *maximal margin hyperplane* or the *optimal hyperplane* which is unique (the proof for the uniqueness is given in (Vapnik, 1998, ch. 10). The task leads to a quadratic optimization problem, which is computationally very demanding when the the *input-space* consists of a large amount of instances or dimensions. The found optimal hyperplane is represented with a set of *support vectors*, i.e. the samples of each class that are closest to the hyperplane in the input-space.

In the input-space, it may be impossible to find a linear hyperplane that is capable of separating all instances, resulting to a zero-valued training error. Therefore it has been common to form the hyperplane in a higher dimensional *feature space.* With this method, a hyperplane that is linear in the feature space can separate non-linearly separable samples in the input-space. The *kernels* determine the relation between the input space and feature space. Polynomials, radial basis functions and two-layer neural networks are the basic choices

[6] SVMs can also be generalized to estimate real-valued functions, relating to regression (Vapnik, 1998, ch. 11). In this generalized form, the margin parameters for separating hyperplanes are replaced by loss-functions.

for kernels. For example with polynomial kernel, the dimensionality N of the feature space is $N = \frac{n(n+3)}{2}$, where n denotes the dimensionality of the input space.

The factors affecting the generalizability of SVMs include the number of support vectors and the dimensionality of the feature space, depending on the chosen kernel and input space. These both factors affect the VC dimension (Vapnik–Chervonenkis dimension) of the model. VC dimension is defined as the cardinality of the training instances that can be left inseparable by the hyperplane, thus relating to the generalizability of the model as separability requires more complexity. A highly complex model might overfit to the anomalies or noise in the training data, irrelevant to the general concepts. With an appropriate kernel and model complexity the hyperplanes can be found that separate the training set in a very high degree. This requires care in choosing the research settings for the classification with SVM. The results obtained without appropriate evaluations on test data might lead to over-optimistic conclusions. In this study, the generalizability of the SVM models was controlled with both research framework and with kernel. The used kernel was a first-degree polynomial.

The SVM was used in Weka with the function *SMO*. The name of the function relates to the optimal hyperplane search algorithm that it implements. SMO (Platt, 1999) solves the computationally expensive optimization of quadratic function by breaking it into smallest possible sub-problems. In Weka, the implementation of SVM in the multi-class case is executed by using pair-wise coupling (Hastie and Tibshirani, 1998). The method involves constructing classification models for each class-pair separately and then coupling the resulting estimates together.

To output proper probability estimates for the predicted classes, logistic regression models were fitted to the outputs of the SVMs. With this choice, the results could be evaluated in terms of RMSE, if a need for that occurred. Among other parameters, the SMO function in Weka is controlled by the parameter c that loosely defines the VC dimension of the models, and therefore relates to the model complexity. A rather small value of $c = 1.0$ was used in the present study to avoid models with high complexity.

Because of its detailed learning, SVM has been used in a wide range of tasks. In music mood classification, SVM has been used for example in the following studies: Laurier et al. (2009); Li and Ogihara (2006); Wieczorkowska et al. (2005); Baum (2006); Li and Ogihara (2003); Muyuan et al. (2004).

3.1.2.2 Search Methods

Two widely used search methods, forward selection and backward elimination, described in the section 2.1.3.2, were used in the study. These methods were chosen for two reasons: First, they require a small amount of evaluated subsets when compared to many more complex heuristic search methods. It was important to reduce the computation time with the choice of the search methods to compensate the fact that the evaluation of subsets in wrapper method is computationally demanding. Referring to the section 2.1.4, the small search space is also increases the generalization capabilities of the models built on the selected feature sets. Second, as forward selection and backward elimination includes or excludes features one at a time, the subsets selected at each iteration are nested which makes it understandable and justified to analyze the subsets at every iteration.

In Weka, forward selection and backward elimination are found under the category of greedy stepwise search algorithms. Because of cross-indexing, the termination condition was omitted which led to adding or deleting features until the full or empty feature set was achieved. Also, to create ranked lists of features according to the subsets at each iteration, the parameter *generateranking* in the implementation in Weka was set to 'true'.

3.1.3 Classification

After feature selection, the learning algorithm corresponding to the algorithm used in selection was run on the whole train set represented with the subsets of 1 to 66 top-ranked features according to the wrapper selection. This resulted to 66 models per each feature selection run. These models were then used to classify the test set represented with the corresponding feature subsets. To ultimately validate the generalizability of the results of certain promising methods the validation set was also classified.

In addition to the classification of test data, the cross-validation accuracies that were used in the selection process were recorded.

3.1.4 Obtaining the Cross-indexing Estimates

To avoid overfitting, the wrapper selection methods were evaluated and compared primarily in terms of estimates obtained by the *generalized* $(N, K - N)$*-fold cross-indexing algorithm* developed by Reunanen (2007). Cross-indexing was originally suggested for obtaining an unbiased estimate of the optimal complexity of a given inductive model. The complexity in this case is defined by the number of input features. To that aim, cross-indexing involves number of feature selection runs from which the estimates of the optimal subset size as well as its performance are inferred. The section 2.1.4 gave an introduction to this method.

The generalized $(N, K - N)$-fold cross-indexing algorithm (with a modification, see the step 2b) is outlined in the algorithm 3.1. The step 1 refers to the cross-indexing loop visualized in the figure 3.1. Stratified splitting, feature selection and classification are conducted as described in the previous sections. Only the classification results on the test set are taken into account in the algorithm.

The loop in the step 2 produces K optimal subset size and performance estimates. Each optimal subset size estimate is based on the values computed in the step 2a by averaging N classification accuracies on the test set and the estimate of the performance attainable with the obtained optimal subset size in the kth iteration is computed in the step 2d by averaging the other $K - N$ classification accuracies not used in the step 2a. The final estimate of the optimal subset size and performance are obtained in the steps 3 and 4.

Based on the variances of the K size and performance estimates obtained with the algorithm, the value of $N = \frac{K}{2}$ was chosen in this study as it produced low amount of variance in both estimates.

When analyzing the preliminary results in this study, the algorithm was found to produce relatively large optimal subset size estimates while almost as high prediction accuracy estimates would have been obtained with notably smaller subsets. This means that the algorithm found the estimate of the optimal subset size although there existed a subset size that would potentially produce less complex models with almost as high prediction potential. Therefore a modification was added in the algorithm, outlined in the step 2b.

At each of the K iterations, all D values obtained in the step 2a are considered as in the original algorithm. Rather than by the global maximum of the D values, the subset size estimate is determined by examining all the local maximum values[7]. First, the local maximum with the smallest subset size is considered as the optimal. Then, comparison with the other local maxima is conducted by a selection criterion which states that increasing the size by one feature an increase of s percents in the maximum values is required for a bigger subset size to win the comparison. If a new winner is found, the optimal subset size is updated. These comparisons are repeated iteratively until all maxima have been evaluated.

[7] As in a graph plotted with the values on the vertical axis and subset sizes in the horizontal axis. The maxima were detected by the function *findpeaks* in *MatLab* after modifying the input vector so that the starting points of maximum plateaus as well as the first and last elements of the vector can be treated as maxima.

Algorithm 3.1 Generalized $(N, K-N)$-fold cross-indexing algorithm, modified from Reunanen (2007). In this study, the value $N = \frac{K}{2}$, the maximum subset size $D = 66$ and $s = 1$ were used. (When $a > b, \sum_a^b (\cdot) = 0$.)

1. Conduct feature selection and classification K times, each time with a given train-test split specific to the kth run $(k = 1, \ldots, K)$. Test set accuracies with every subset size, denoted by $\left(x_1^{(k)}, x_2^{(k)}, \ldots, x_D^{(k)}\right)$ are obtained.

2. for $k = 1, \ldots, K$

   (a) For each subset size $(i = 1, \ldots, D)$, compute the mean of N estimates: $\dot{x}_i^{(k)} = \frac{1}{N}\left(\sum_{j=\alpha}^{K} x_i^{(j)} + \sum_{j=\beta}^{k} x_i^{(j)}\right)$,
   where $\alpha = K + k - N + 1$
   and $\beta = \max(1, k - N + 1)$.

   (b) Use them to select the optimal model complexity:

      i. Find local maxima $p_l^{(k)}$ from $\dot{x}^{(k)}$. The maxima are ordered by the subset size, starting from the maximum with the smallest size (starting points of local maximum plateaus are considered as local maxima). Initialize the optimal complexity $d^{(k)} = p_1^{(k)}$.

      ii. For each local maximum $l = (2, \ldots, numberofmaxima)$, test if the current maximum satisfies the equation $\dot{x}_{p_l^{(k)}}^{(k)} > \dot{x}_{d^{(k)}}^{(k)} + \dot{x}_{d^{(k)}}^{(k)} \cdot \frac{s}{100} \cdot (p_l^{(k)} - d^{(k)})$. If so, assign the current maximum as the optimal complexity $d^{(k)} = p_l^{(k)}$.

   (c) The remaining optimal complexity $d^{(k)}$ is considered the optimal subset size at the kth iteration.

   (d) Calculate the average performance at this level of complexity for the other $K - N$ folds:
   $\ddot{x}_{d^{(k)}}^{(k)} = \frac{1}{K-N}\left(\sum_{l=1}^{k-N} x_{d^k}^{(l)} + \sum_{l=k+1}^{\gamma} x_{d^{(k)}}^{l}\right)$,
   where $\gamma = \min(K, K + k - N)$.

   (e) For comparison purposes, the performance for the full feature set on the kth iteration $(\ddot{x}_D^{(k)})$ can also be recorded.

3. Average all the K subset sizes $d^{(k)}$ obtained during the different executions of the step 2c. This average is the estimate for the optimal subset size.

4. Average all the K performance estimates obtained in the step 2d. This average is the estimate for the performance of the best subset having the size discovered during the step 3.

Based on the preliminary results, the addition of the modification both reduced the optimal complexity significantly while maintaining the prediction capability at almost the same level. The modification also reduced the variance of the optimal subset size estimates found in each iteration in the step 2.

The trade-off between prediction accuracy, subset size and the relation between the accuracies of the full subset and the optimal subset was examined in this study.

3.1.5 Redundancy and Relevance of Features

While cross-indexing produces estimates of the optimal subset size for a specific learning method, the analysis yields information that enables to study several aspects of the results. Among them are redundancy and usefulness of the features in the selected subsets (described in the section 2.1.2). The measures are obtained by taking account all runs in the cross-indexing loop. The reliability of the estimates depends on the number of runs. Therefore the contributions were analyzed only in the experiment 2.

Redundancy can be measured in terms of correlation between features as described in the section 2.1.2.2. In this case, the aim was to describe to what extent the selection or elimination of features with a given wrapper selection method made possible a reduction of the redundancy in the training data. Thus, the mean correlations in the selected feature subsets were computed separately for each subset size and the means and standard deviations over the runs were reported.

Usefulness was computed following the suggestions in (Caruana and Freitag, 1994), described in the section 2.1.2.3. The measure describes how useful a given feature is for a learning algorithm by comparing the performance of the algorithm with and without the feature. If adding the feature increases the performance, it is considered as useful. In the present study all feature selection runs provide estimate of the usefulness of each feature. Therefore it is easy to compute a general usefulness measure that takes all runs into account.

To estimate the usefulness of a feature i, first the difference $U_i^{(k)}$ was computed at every kth feature selection run:

$$U_i^{(k)} = x_{r_{i(k)}}^{(k)} - x_{r_{i(k)}-1}^{(k)},\; k \in \{1, 2, \ldots, K\}$$

where K is the number of runs, $x_a^{(k)}$ is the test set classification accuracy with the subset size a in the kth run, $r_{i(k)}$ is the ranking of the feature i in the kth run and $x_0^{(k)} = 100/numberofemotions$, i.e. the accuracy if the null hypothesis holds true. The final measure for feature i is then computed by dividing the number of runs where the feature was useful ($U_i^{(k)} > 0$) with the total number of runs, i.e. $\frac{1}{K}|\left\{U_i, U_i^{(k)} > 0\right\}|$, where $|\cdot|$ denotes the cardinality. These measures was computed for both test classification and validation.

3.1.6 Emotions in the Misclassified Excerpts

Selecting applicable excerpts, as described in the section 3.1.1.2 aimed at reducing the ambiguousness of the expressed emotions in the analyzed excerpts. Although the expressed emotions in the remaining excerpts can be considered as unambiguous it does not exclude the possibility that even emotions not highest rated emotions might describe the excerpts reasonably well. Therefore it would not be conclusive to lay too much emphasis on the accuracy measures that do not take into account the emotions not highest rated.

A novel measure used to deal with this issue is called the *Mean Ranking of the Predicted Emotion (MRPE)* that is computed over the set of misclassified excerpts. The ranking of the predicted emotion of a misclassified excerpt is given in terms of the participants' ratings described in the section 3.1.1.1.

In the experiments with four emotion labels a value $MRPE < 3$ indicates that the predicted emotions of the misclassified excerpts are more representative in average than random emotions while $MRPE \geq 3$ indicates the opposite[8]. Therefore the $MRPE$ values give further evidence of the extent in which a model has learned implications of the finer details in the data.

3.1.7 Experiments

The experiments in this context involve running the cross-indexing loop with chosen learning and search methods. To gain useful, justified and interesting information and to meet the presented aims of the study, the experimental design is divided into two stages. This choice was also made because of the high computational burden induced by the wrapper selection, especially when using Support Vector Machine as a learning method.

Experiment 1 In the experiment 1 the process was run *four* times with every pair of search and learning methods.[9] The aim of the experiment was to be able to compare the methods in terms of cross-indexing estimates while keeping the executing time reasonable.

[8] In this case the predicted emotions in the misclassified excerpts are ranked second, third or fourth in the ratings. Three is the average of these rankings.

[9] The number refers to the K-value presented earlier.

Based on the obtained cross-indexing estimates, few promising methods were selected for the experiment 2.

Experiment 2 The chosen feature selection methods were studied more thoroughly in the experiment 2. To improve the reliability of the results, a value $K = 30$ was chosen. Because of the sampling, a high amount of runs made the cross-indexing estimates more reliable as well as met the demands imposed by the analysis of feature redundancy and usefulness.

Apparatus The analysis was conducted on a single 32 bit desktop computer from Packard Bell NEC with the following specifications:

Operating system: Linux.

System memory size: 1536MiB.

Processor: Intel(R) Pentium(R) 4 CPU 3.06GHz.

3.2 Results and Discussion

3.2.1 Experiment 1

In the experiment 1 the cross-indexing loop was run with four folds, corresponding to the number of feature selection runs with each combination of the search and learning methods. The average running times are presented in the table 3.3 which shows that wrapper selection using the simple Naive Bayes and k-NN as learning methods resulted in rather short computation times whereas the SVM required more effort.

	FS	**BE**
NB	3 min 29s	6min 35s
k-NN	50s	59s
SVM	4h 17min 59s	4h 13min 46s

Table 3.3: Mean computation times of the feature selection runs.

The aim of the experiment was to use cross-indexing to select the optimal subset sizes for the chosen learning methods and to estimate the performances of the learning methods with those sizes. The figure 3.2 demonstrates the benefits of cross-indexing and highlights the problems related to evaluation of feature selection methods with traditional accuracy measures. The figure displays the averaged accuracy estimates as well as the cross-indexing estimates for the optimal subset size and its performance. The cross-validation accuracies correspond to the measures used to guide the selection and the test set accuracies correspond to the measures produced by the traditional external cross-validation loop.

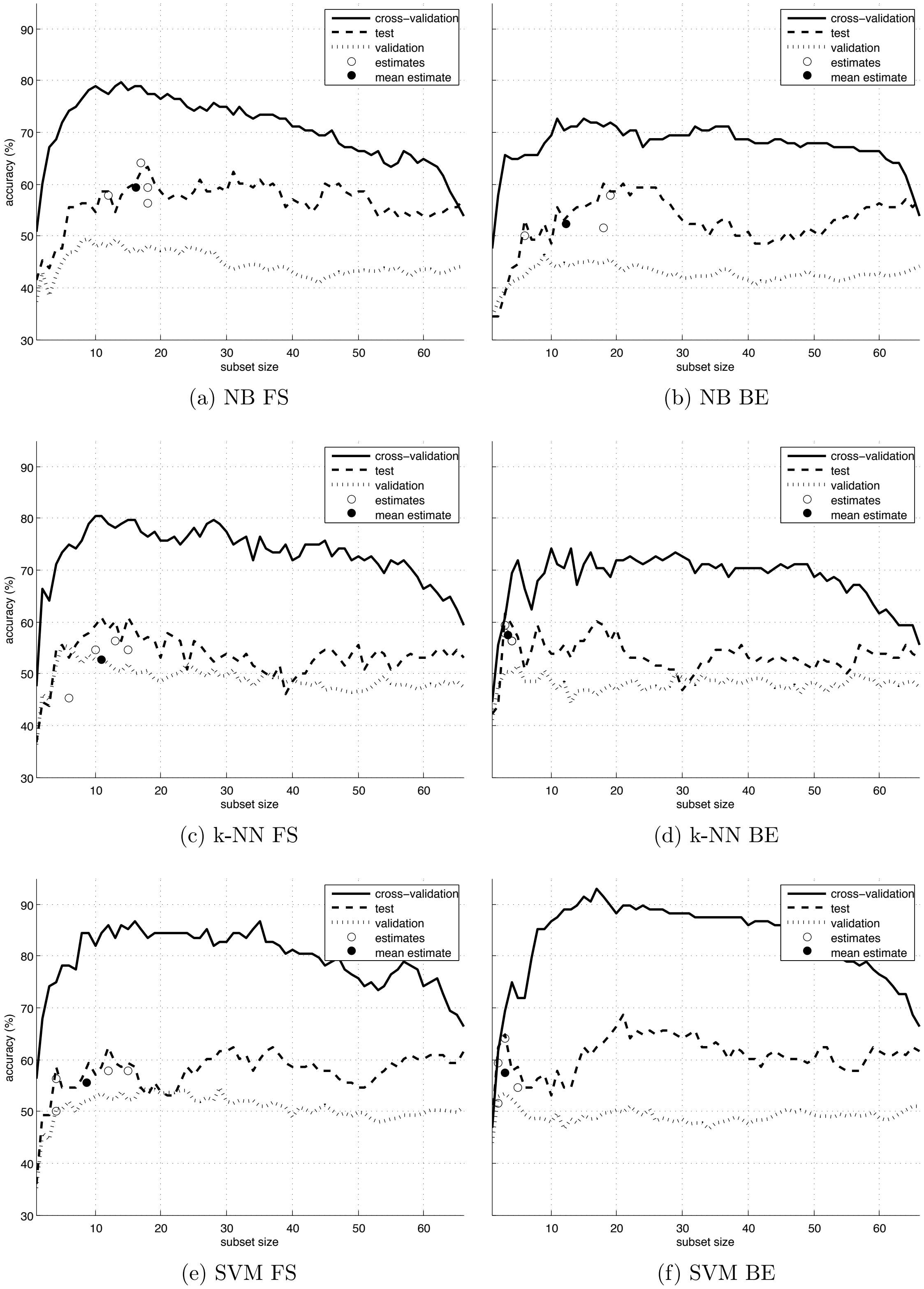
Based on the graphs, it is clear that the averaged cross-validation and test set estimates failed at predicting the performance of the methods when they were used for the larger set of validation data. It would therefore be misleading to use the averaged values for determining the optimal subset sizes for a given learning method and to use these values for estimating the performance of the method with that size. Moreover, the lack of clear maxima in the test set performances make it hard to assess measures that could be used to compare different methods. These results support the evidence discussed in the section 2.1.4 where the measures used to guide the selection as well as the measures obtained by the external loop of cross-validation were considered to give over-optimistic, biased and misleading information about the true performance of the wrapper methods.

When comparing the cross-indexing estimates to the validation set performances, it is notable that these estimates generally correspond fairly well to the subset sizes which were the optimal in terms of the validation set. Also the performance estimates with the optimal subset sizes are closer to the validation set performances than the averaged test set accuracies. The cross-indexing estimates are plotted in the figure 3.2 (denoted by circles). The estimates are also summarized in the table 3.4.

Method	**Subset Size**	**Accuracy (%)**	**Improvement (%)**
NB FS	16.25 ± 2.87	59.38 ± 3.38	2.34 ± 5.34
NB BE	12.25 ± 7.23	52.34 ± 3.72	-4.69 ± 9.88
k-NN FS	11.00 ± 3.92	52.73 ± 5.00	-0.39 ± 6.30
k-NN BE	3.50 ± 0.58	57.42 ± 1.50	4.30 ± 3.46
SVM FS	8.75 ± 5.62	55.47 ± 3.72	-6.25 ± 3.38
SVM BE	3.00 ± 1.41	57.42 ± 5.47	-4.30 ± 5.32

Table 3.4: Cross-indexing estimates.

Comparing the cross-indexing estimates, Naive Bayes with forward selection as well as k-Nearest Neighbors and Support Vector Machine with backward elimination yielded the highest performances. However, the accuracies varied notably between the iterations (with standard deviations from 1.50% to 5.47%) which causes uncertainty into the comparison of the mean accuracies, which differ by 6.65% at most. Therefore a high importance must be given to the subset sizes, which are preferably small to keep the models simple and interpretable. In this comparison SVM and k-NN with backward elimination consistently yielded the smallest optimal subset sizes with under 5 features. With the Naive Bayes classifier the found optimal



(a) NB FS (b) NB BE

(c) k-NN FS (d) k-NN BE

(e) SVM FS (f) SVM BE

Figure 3.2: Cross-indexing estimates and the averaged accuracies.

subset sizes were the largest. Generally, all classifiers yielded bigger subset sizes with forward selection than with backward elimination, which was surprising since forward selection is usually considered to give bigger subset sizes than backward elimination.

Last, when analyzing the average improvements in the prediction accuracies that the optimal subset sizes gave in comparison to the full feature sets, it is notable that the performance with SVM decreased when the optimal subset sizes were used. This could be due to the extensive learning of the classifier, which could also result in overfitting when the number of features is large. Looking at the results more carefully, the performances decreased constantly only with SVM and forward selection. With k-NN and backward elimination the results support the effectiveness of feature selection the most clearly as the performances increased in all iterations.

Based on the results, SVM and k-NN with backward elimination gave the most convincing evidence of the effectiveness of feature selection. The optimal subset sizes with these both methods were the smallest which makes the models the most interpretable. The subset sizes had only small amount of variance and therefore these methods are potentially useful in practice. Both of these methods gave also relatively good prediction accuracies. For these reasons, SVM and k-NN classifiers with backward elimination were chosen for the experiment 2.

3.2.2 Experiment 2

The feature selection was run 30 times in the experiment 2 to obtain reliable estimates of the performances. The table 3.5 shows that k-NN yielded better results than SVM in terms of all measures. Although difference in the estimated accuracies between the methods was not significant because of relatively large variation, the difference in the optimal subset sizes is larger. The rounded optimal subset sizes are 4 for k-NN and 6 for SVM. K-NN yielded constantly at least as good accuracies as SVM, but with smaller subsets. Especially the small variation in the subset sizes give reliability to the results.

Method	**Subset Size**	**Accuracy (%)**	**Improvement (%)**
k-NN BE	3.50 ± 0.86	56.52 ± 2.77	0.79 ± 2.40
SVM BE	6.30 ± 1.78	54.25 ± 1.85	-8.04 ± 1.92

Table 3.5: Cross-indexing estimates.

To test the validity of the results, the cross-indexing estimates were again compared to the validation set performances, leaving out the cross-validation and test set accuracies. The figures 3.3 and 3.4 show the results for the used methods. The continuous lines show the mean classification accuracies over all 30 models build with the specific subset sizes and the horizontal distances between the triangles demonstrate the standard deviations of these accuracies.

When using the k-NN, it is notable that the average validation set accuracies generally decreased after adding more than six features in the set. This indicates that k-NN truly benefits from backward elimination. The figure 3.3 shows that the mean cross-indexing estimate of the optimal subset size corresponds to the maximum of the average validation set performances. The level of the maximum accuracy was reached with three features which exactly matches most of the estimates. Moreover, all of the estimates were located at the interval where the maximum level of the validation set performance is reached.

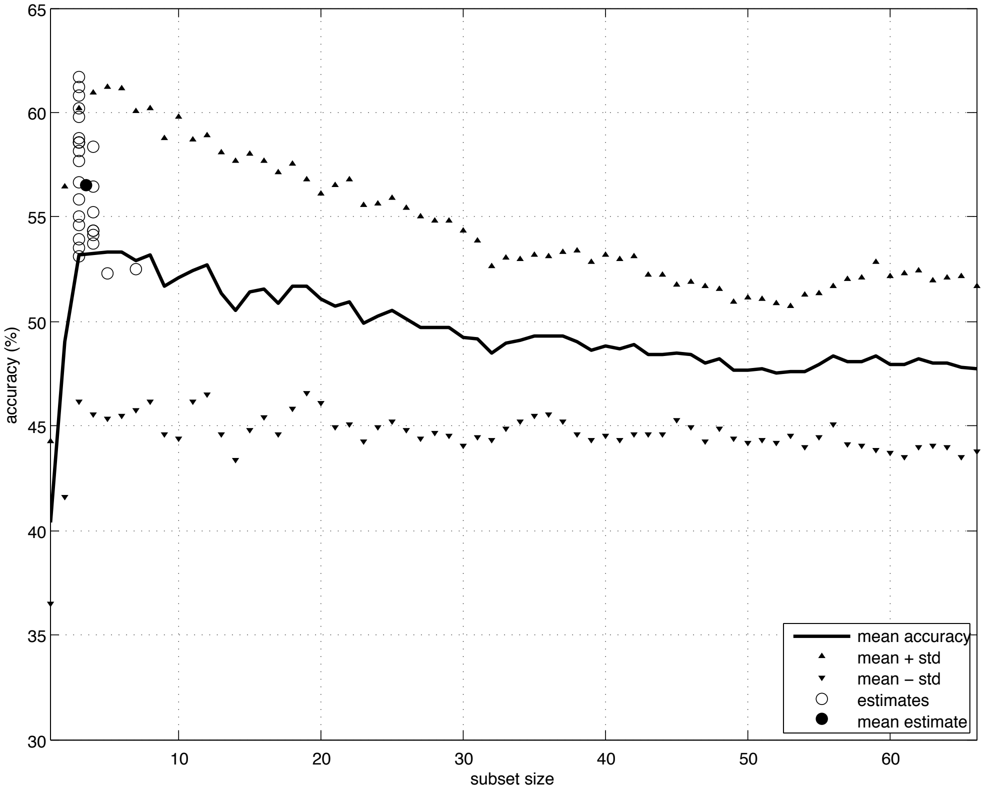


Figure 3.3: k-NN BE

While the optimal subset size was detected reliably, there exists large variation between the cross-indexing estimates of the performances. The mean of the accuracies is also slightly optimistic when compared to the validation set performance. However, it must be noted that the groups of participants who rated the primary and validation set were different, which could have induced slight inconsistency between the sets. Although the estimated accuracies varied, 27 of the 30 estimates were less than one standard deviation from the corresponding mean validation set accuracies.

The figure 3.4 shows the performance of SVM with backward elimination. The validation set accuracies imply that SVM gives relatively good performances with subsets of 4 to 7 features, but models built on larger sets with more than 25 features generalize also well. This indicates that SVM is not overly prone to overfitting.

However, the reason for adding the modification to the cross-indexing algorithm was to weight the algorithm towards finding small but effective feature sets. The figure shows that cross-indexing yielded accurate estimates of the validation set performance peaks with small subset sizes – both of the estimate clusters point to the two peaks in the accuracy curve and the averaged estimate of the subset sizes, in fact, points to the higher of these peaks.

As with k-NN, the majority of the accuracy estimates with SVM are optimistic but again, most of the distances between the estimates and the corresponding validation set accuracies are less than the standard deviations.

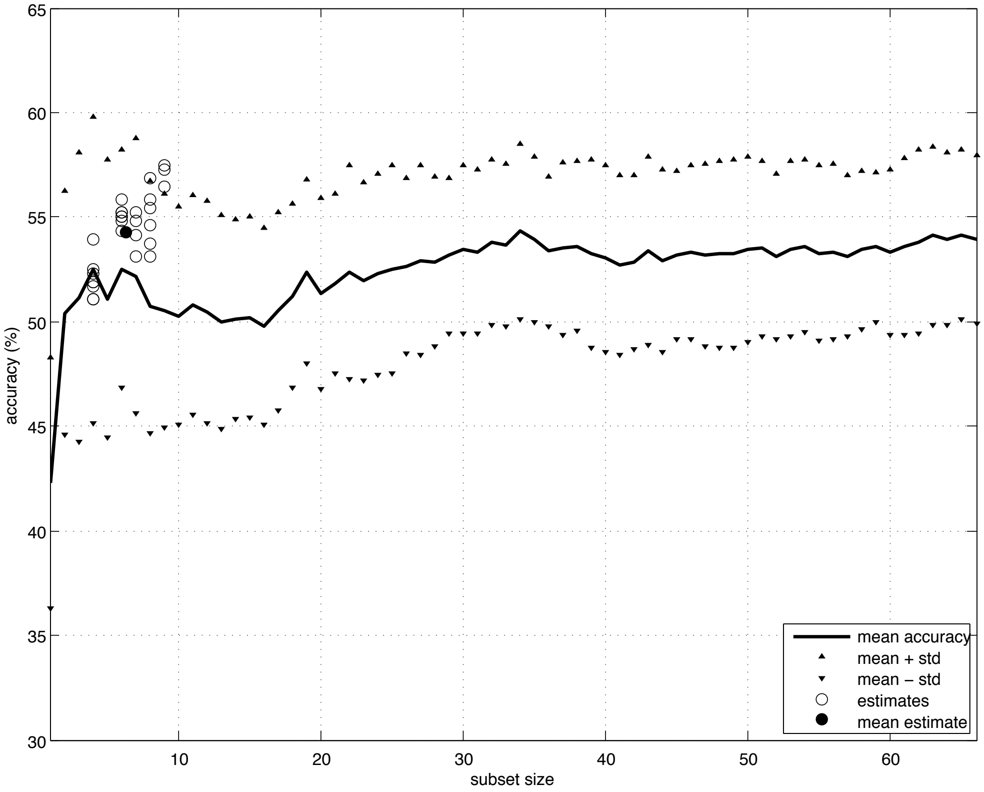
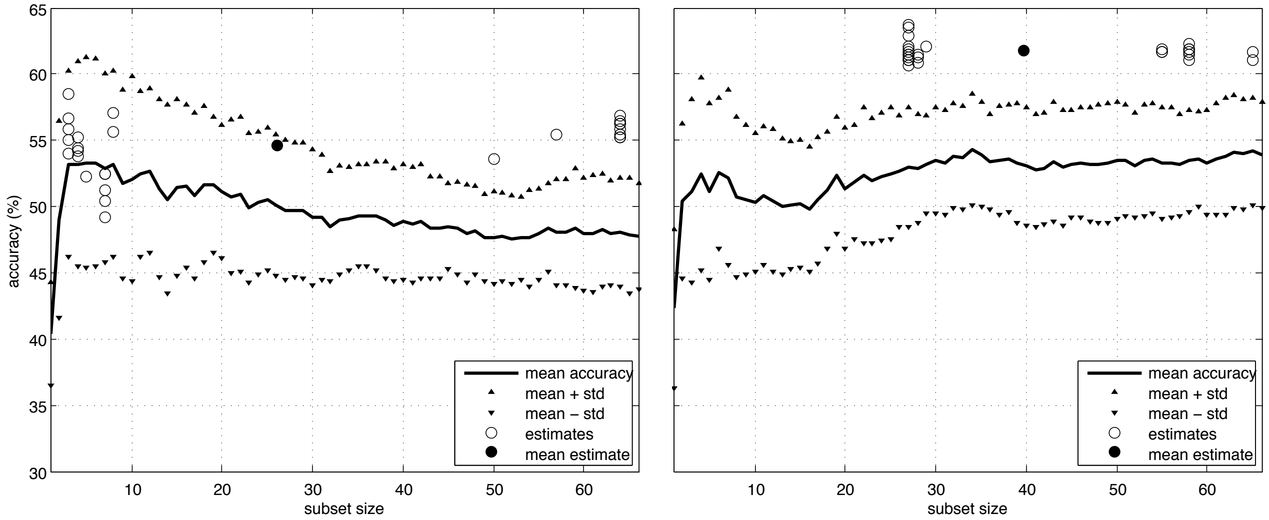


Figure 3.4: SVM BE

To test the efficiency of using the modification to the cross-indexing algorithm, the modified algorithm was compared to the original version. To this aim the original cross-indexing algorithm with the same parameters was used to estimate the optimal subset sizes and performances of the methods in the experiment 2. In fact, the original algorithm can be thought as a special case of the modified algorithm: if the parameter s that controls the amount of weight given to the small subset sizes is set to $s = 0$, the global maximum performances at each iteration are considered as the optimal subset sizes, which corresponds to the original cross-indexing algorithm. The results are presented in the figures 3.5a and 3.5b.

As can be seen in the figures, the estimates for the optimal subset size gained with the original cross-indexing algorithm varied notably to the extent that the average estimate became rather unrepresentative. In addition, especially with SVM, the accuracy estimates were significantly optimistic when compared to the validation set performance.

The results could be nevertheless analyzed in view of the clusters of estimates. With k-NN, a cluster is located at the optimal subset according to the validation set accuracies but the estimates in the another cluster can be considered as false estimates. With SVM the clusters can be considered as satisfying estimates of globally optimal subset sizes. However, the results indicate generally that the modified version of the algorithm gives more reliable estimates than the original version.



(a) k-NN BE

(b) SVM BE

Figure 3.5: The original cross-indexing algorithm without modification, i.e. with the parameter $s = 0$.

3.2.2.1 Confusions Between the Emotions

The tables 3.6a and 3.6b show the confusion matrices of the classifications with the estimated optimal subset size 4 of k-NN. Each cell in the matrices is averaged over the 30 runs. Information retrieval measures of each class are also presented. It can be seen that with the *test* set classification there was most confusion within class pairs *anger_fear* - *sad* and *happy* - *tender*. Especially samples belonging to the class *happy* were often classified as *tender*. There was not much confusion between these groups as happiness was well-separated from anger and fear and especially from sadness. This same held true for sadness, which was rarely classified as *happy*. Also tenderness was well-separated from anger and fear. Most confusions between the pairs was caused by the samples belonging to the class *sad* which had tendency to be classified as *tender*.

The same patterns are indicated also in the validation set classification. However, in the validation set, sadness and tenderness were confused more than in the test sets. The classes that were confused the least were again *anger_fear* and *tender* as well as *happy* and *sad*. This was satisfying because supposedly these specific emotions are also perceptually distant.

Act.↓ /Pred.→	**Anger_Fear**	**Happy**	**Sad**	**Tender**	**Precision**	**Recall**	**F**
Anger_Fear	12.4	4.1	5.4	3.1	0.57	0.50	0.51
Happy	2.0	16.0	0.7	6.3	0.62	0.64	0.60
Sad	5.2	2.3	12.4	5.1	0.60	0.50	0.51
Tender	1.7	4.6	2.6	16.2	0.55	0.65	0.57

(a) Test sets.

Act.↓ /Pred.→	**Anger_Fear**	**Happy**	**Sad**	**Tender**	**Precision**	**Recall**	**F**
Anger_Fear	14.5	3.9	5.0	1.7	0.60	0.58	0.58
Happy	3.4	11.8	2.4	7.4	0.59	0.47	0.50
Sad	4.9	1.8	12.8	5.6	0.52	0.51	0.50
Tender	2.2	4.2	4.5	14.2	0.50	0.57	0.52

(b) Validation set.

Table 3.6: Averaged confusion matrices with the subset size 4 with k-NN represented in percents together with averaged information retrieval measures for the classes.

Similar patterns can be found in the classification with SVM with the estimated optimal subset size 6. The confusion matrices are presented in the figures 3.7a and 3.7b. Again, in the *test* set classification, most of the confusions occurred in overall within the class-pairs *anger_fear* - *sad* and *happy* - *tender*. However, this time the division into the pairs was weaker. For instance, tenderness was confused with all classes to some extent and the confusions between tenderness and happiness was smaller. The patterns in the validation set classification were more similar to the ones found with k-NN – *anger_fear* and *tender* as well as *happy* and *sad* were better separated in the validation set compared to the other class-pairs.

Act.↓ /Pred.→	**Anger_Fear**	**Happy**	**Sad**	**Tender**	**Precision**	**Recall**	**F**
Anger_Fear	9.5	3.2	7.8	4.5	0.55	0.38	0.42
Happy	1.7	18.2	0.7	4.4	0.71	0.73	0.70
Sad	5.2	1.0	13.7	5.1	0.55	0.55	0.53
Tender	1.3	4.50	3.4	15.7	0.55	0.63	0.57

(a) Test sets.

Act.↓ /Pred.→	**Anger_Fear**	**Happy**	**Sad**	**Tender**	**Precision**	**Recall**	**F**
Anger_Fear	12.3	4.1	6.1	2.4	0.54	0.49	0.50
Happy	3.1	13.5	2.1	6.2	0.62	0.54	0.56
Sad	5.6	1.6	13.0	4.8	0.51	0.52	0.50
Tender	3.0	3.8	4.6	13.7	0.50	0.55	0.52

(b) Validation set.

Table 3.7: Averaged confusion matrices with the subset size 6 with SVM represented in percents together with averaged information retrieval measures for the classes.

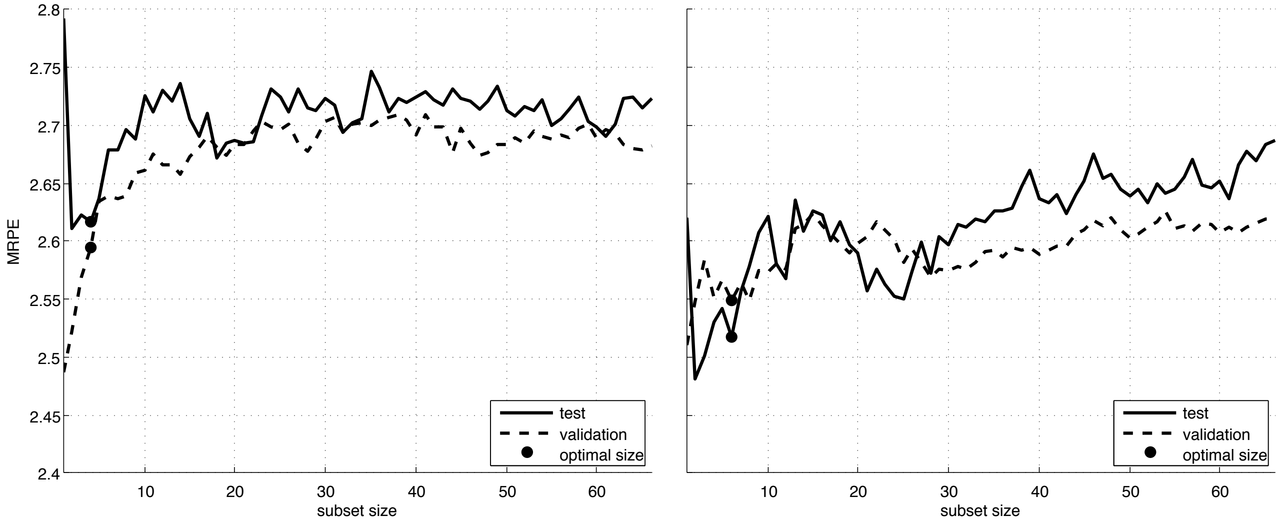
3.2.2.2 Emotions in the Misclassified Excerpts

The analysis of emotion contributions concentrated on the excerpts that were misclassified by the models. The aim was to estimate how the misclassified excerpts had been rated in terms of the predicted emotion when the material was annotated. This could reveal the effectiveness of the models in detecting more profound characteristics and similarities in the data than could be described with single class labels. To this aim, *MRPE* values described in the section 3.1.6 were calculated. The values specific to a given subset size were averaged over the 30 runs. The results are shown in the figures 3.6a and 3.6b.

It can be seen that the optimal subset sizes found with both k-NN and SVM gained good results in both test and validation set. This was very satisfying since the *MRPE* values are not necessarily in a straight relation with the prediction accuracies. Both optimal subset sizes of the classifiers gave better values in both test classification and validation than larger subset sizes. In fact, the results generally show that smaller subsets perform better than larger subsets in terms of *MRPE* values.

3.2.2.3 Redundancy

The figure 3.7 presents the mean correlations in the feature subsets. The mean correlations generally decreased with both k-NN and SVM along with the size of the subsets. This indicates that the both methods fairly consistently eliminated features whose redundancies were higher than the average redundancies in the subsets. However, the variance of the redundancies in the subsets of less than 10 features was higher than in larger subsets – it

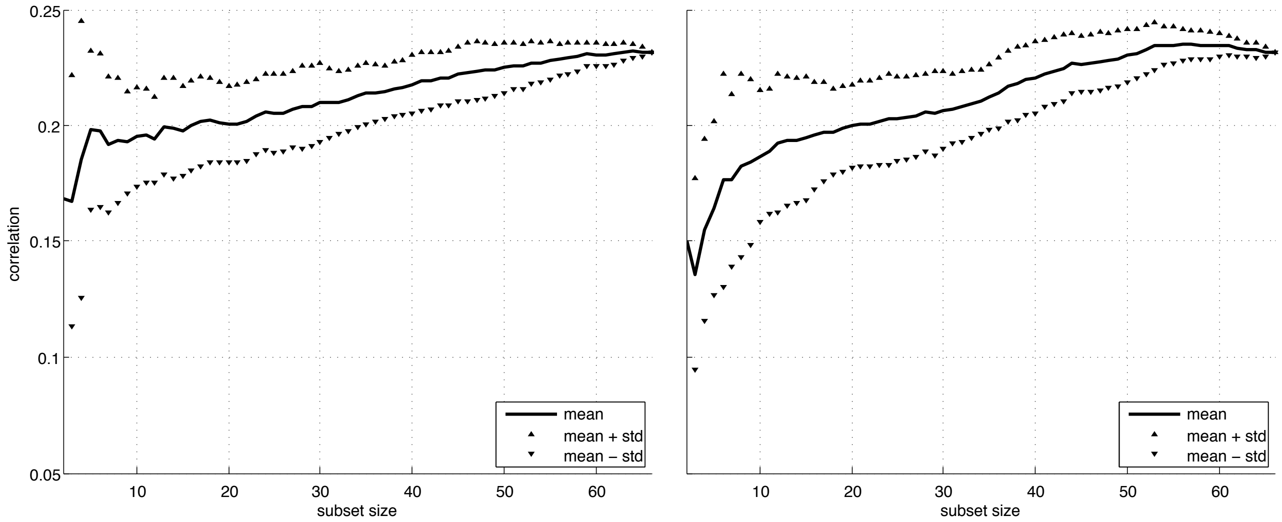


(a) k-NN backward

(b) SVM backward

Figure 3.6: *MRPE* values. The values refer to the average rankings of the predicted emotions in the ratings of the misclassified samples. A small value is an implication of efficient models.

must be noted that the feature sets from different feature selection runs are mutually less overlapping when the subset size is small. This could nevertheless indicate that the relevance of features play more significant role in the classification performance of the models when the subset size is small while the effect of redundancy is more important when the subset size is large.



(a) k-NN BE

(b) SVM BE

Figure 3.7: The mean correlations averaged over runs.

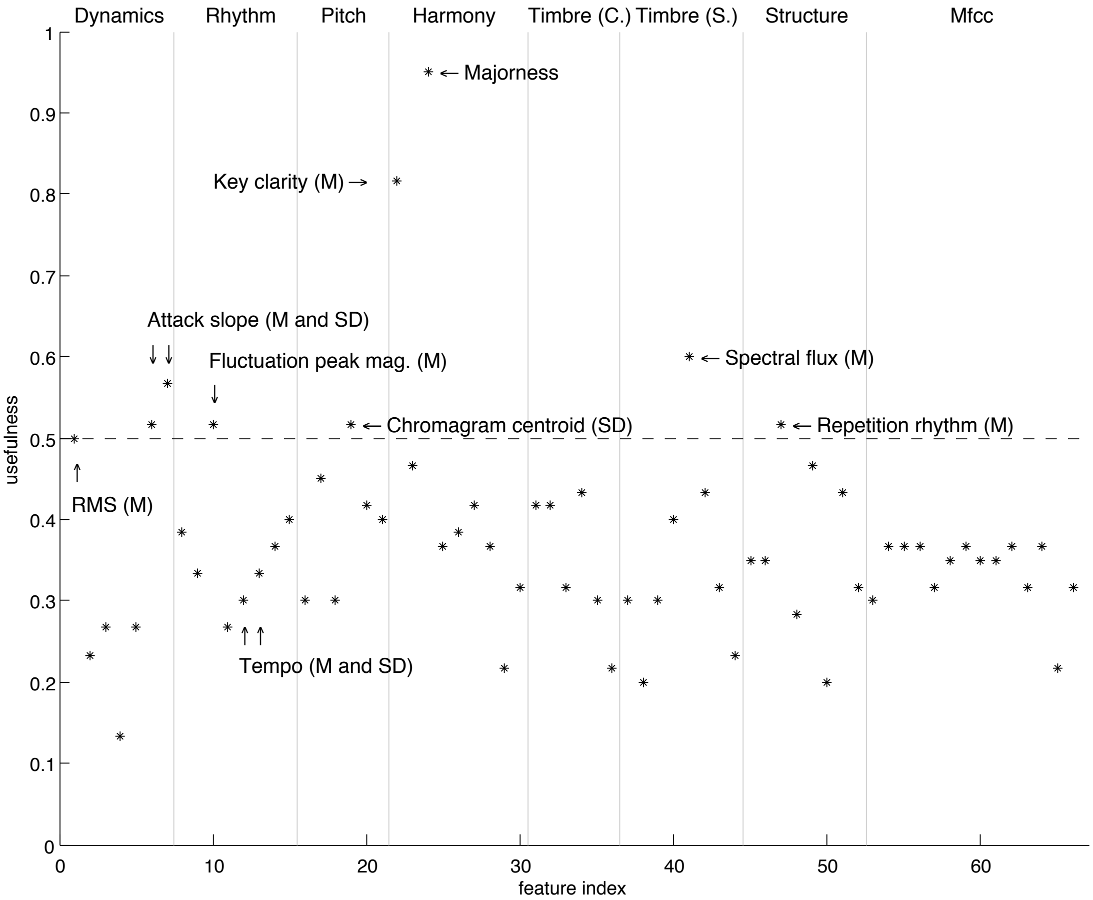
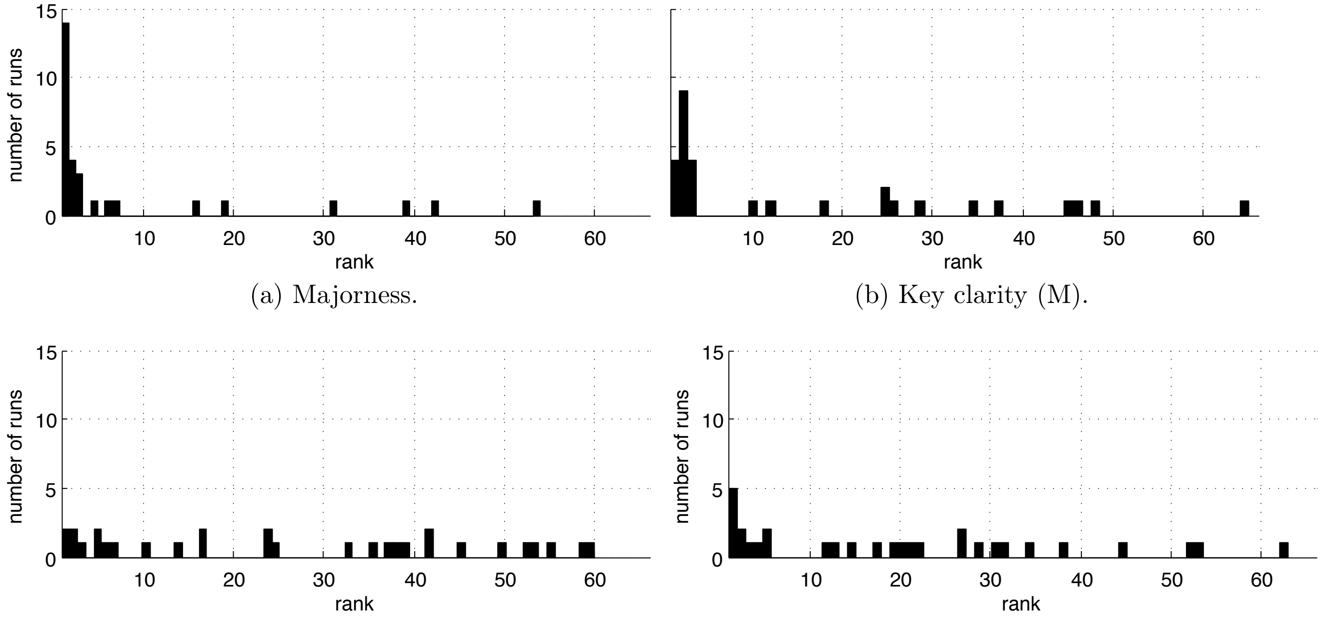


Figure 3.8: Usefulness of the features indexed in the table 3.2. Features with values above 0.5 can be considered as useful, meaning that they generally provided increase in the performance of the classifier. Value 1 is the maximum.

3.2.2.4 Usefulness of the Features

The analysis of the usefulness of features concentrated on k-NN since it yielded higher accuracies and smaller optimal subsets than SVM. The results are summarized in the figure 3.8. Both test classification and validation were taken into account in the analysis by taking the mean of the two usefulness measures for each feature. The figures 3.9a – 3.9d show the rankings of several interesting features. The rankings relate to the subset sizes where the features were eliminated at each run.

Two rather high-level features perceptually – majorness and key clarity – were clearly found to be the most useful features for k-NN. Also several features – spectral flux, mean and standard deviation of attack slope, peak magnitude of fluctuation, standard deviation of chromagram centroid, repetition of rhythm and RMS – were found to increase the performance at half of the runs at least. For most of the feature categories (dynamics, rhythm



(a) Majorness.

(b) Key clarity (M).

(c) Spectral flux (M).

(d) Attack slope (M).

Figure 3.9: Feature rankings.

etc.) one or two features stood out. Only timbre (centroid) and MfCCs did not provide any features with usefulness over 0.5.

The high usefulness of majorness can easily be explained by its relation to emotion perception – it is considered as one of the best-known attributes that determine whether music is happy or sad. For example Laurier et al. (2009) found in their analysis of the *110 set* that the mode of the excerpt is good predictor of all of the emotion categories. Their analysis showed that 95% of the excerpts expressing happiness or tenderness were in major mode while 85% of sad, 80% of fearful and 60% of angry excerpts were in the minor mode. Majorness was ranked the best single feature in 14 runs and was included in the optimal subset in 21 runs, as seen in the figure 3.9a.

Also the high performance of key clarity in the prediction of emotions can be related to emotion perception. In the study by Thompson and Robitaille (1992), tonal music was perceived by the listeners as joyful and peaceful while atonal music was related to angry melodies and chromatic music was perceived as rather angry and sad. Key clarity was included in the optimal subset in 17 runs, as seen in the figure 3.9b.

An explanation for the high performance of spectral flux in emotion prediction can be derived from the finding in the study by Alluri and Toiviainen (2009). Using Indian popular music as a stimulus material, they found high correlation ($r = .75$, significant at the level $p < .001$) between spectral flux and the perceived activity in the music. Eerola and Vuoskoski (2009) found a relation between activity and emotion perception in music: studying the correspondence between discrete and dimensional models of emotions in the *110 set* by canonical correlations they found that a variate related to activity (inverted) correlated with energy ($r = -.92$), happiness ($r = -.64$), and sadness ($r = .85$). The variate explained 30% of the variance in the ratings of discrete emotions. Although spectral flux was found to be generally useful feature, it was included in the optimal subset in only 5 runs.

Although the relevance of articulation cannot be denied in the context of emotion perception, the perceptually meaningful phenomenon behind attack slope could have been the same as with spectral flux since 71.1% of their variances were overlapping. Also RMS feature could have been used as a description of the same concept in the models as its variance was 83.9% overlapping with spectral flux and 71.1% overlapping with attack slope. The standard deviation of attack slope was found to be slightly more useful feature than its mean. However, analysis of these features reveals high 97% correlation which indicates that both of these features could represent the same phenomenon. Based on the rankings, the mean attack slope can be considered as more effective than standard deviation since it was included the optimal subset in 9 runs whereas the standard deviation was in the subset in only 4 runs.

It was notable especially that tempo was found to be relatively useless feature for k-NN since previous studies have found a clear influence of tempo on perceived emotion (Juslin, 1997; Fabiani and Friberg, 2008). This was supported by findings that mean and standard deviation of tempo did not correlate highly (below the level $p < .001$) with any other features in the set which means that the variance in tempo could not have been represented by any other feature. However, event density (feature number 8) that is a cue for tempo perception (Goebl and Dixon, 2001) was found to be more significant feature.

The figure 3.10 shows the average number of features of each category in the optimal subsets of the size 4. In overall, perceptually rather high level features relating to dynamics and harmony and structure were included in the optimal subsets. By contrast, low-level features describing timbre, including MfCCs were eliminated at earlier stages in the wrapper selection process and did not make it to the subsets.

3.2.2.5 Contributions of the Features to the Confusions Between Emotions

To elucidate how a particular feature contributed to the recognition of – and confusions between – specific emotions, the differences in the confusion matrices with and without the feature were studied. This analysis was conducted for the most useful features – the mean of majorness, key clarity, spectral flux and attack slope.

For each feature, the mean of the confusions between each pair of emotions was computed

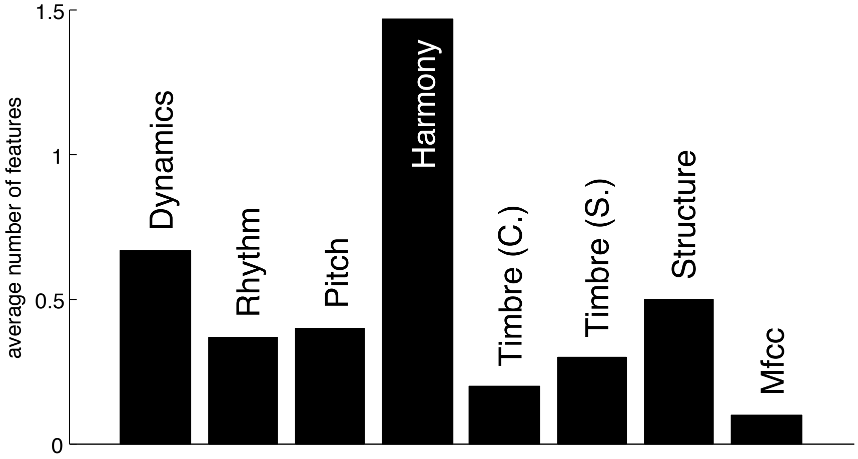


Figure 3.10: Feature categories in the subsets of the size 4.

separately over the classification performances of the models before and after including the feature. In the cases when the feature was ranked first, the confusion matrix was compared to a matrix with the expected values for a random classification (each cell with a value $\frac{numberofinstances}{numberofemotions^2}$). Then, differences between these means were tested by independent two-tailed two-sample T-tests[10]. The same procedure was repeated on the validation set classification performances. The results are shown in the table 3.8.

As expected, majorness significantly reduced the confusions between happiness and sadness in both test and validation sets. Majorness also reduced the confusion between tenderness and sadness. Other notable differences was found with the excerpts that expressed anger or fear. With majorness in the feature set, *anger_fear* was both classified correctly more often and caused less confusion with the labels *happy* and *tender*. A single notable class-pair whose confusion was affected negatively by majorness in both test and validation sets was *sad* and *anger_fear* – majorness increased misclassification of sad excerpts into angry or fearful. In overall, these patterns give evidence that k-NN succeeded to learn properties of the well-known perceptive qualities of the majorness of mode.

As hypothesized specifically by the distinction between the expressive qualities of tonal and atonal/chromatic music, inclusion of key clarity in the set reduced confusions between *anger_fear* and the other emotions, namely misclassification into *anger_fear*. The recognition rate of the label was also increased remarkably. The most significant effect of key clarity was the reduction of the confusions between *anger_fear* and *tender*.

Adding spectral flux into the models did not result in many significant differences in the confusions. However, the effects that were found were positive and generally related to

[10] MatLab function *ttest2* was used for T-test.

Act.↓ /Pred.→	**Anger_Fear**	**Happy**	**Sad**	**Tender**
Anger_Fear	$+^{***}(+^{***})$	$-^{**}(-^{***})$		$-^{**}(-^{**})$
Happy			$-^{***}(-^{***})$	$(+^{*})$
Sad	$+^{***}(+^{***})$	$-^{***}(-^{***})$		$-^{**}(-^{**})$
Tender			$-^{***}(-^{***})$	$(+^{*})$

(a) Majorness

Act.↓ /Pred.→	**Anger_Fear**	**Happy**	**Sad**	**Tender**
Anger_Fear	$+^{**}(+^{***})$			$-^{*}(-^{***})$
Happy	$-^{***}(-^{***})$			
Sad	$(-^{**})$			
Tender	$-^{***}(-^{***})$			$(+^{*})$

(b) Key clarity (M)

Act.↓ /Pred.→	**Anger_Fear**	**Happy**	**Sad**	**Tender**
Anger_Fear				
Happy		$+^{*}(+^{*})$	$-^{*}$	
Sad				
Tender		$-^{*}$		

(c) Spectral flux (M)

Act.↓ /Pred.→	**Anger_Fear**	**Happy**	**Sad**	**Tender**
Anger_Fear				
Happy		$+^{*}$	$-^{*}$	$-^{*}$
Sad				
Tender		$-^{*}$		

(d) Attack slope (M)

Table 3.8: Effects of the individual features in confusions between emotions in the models. Only the significant differences $(^{***}p < .001,^{**} p < .01,^{*} p < .05)$ along with the direction of the effect ($+/-$ for a positive/negative effect if the feature is in the set) are reported. $+$ and $-$ denote the test set and $(+), (-)$ denote the validation set. Note that the positive effects on the diagonal and negative effects outside the diagonal are considered as eligible.

the recognition of happiness. After including the feature, happy excerpts were more often classified correctly and less often classified as *sad*. Also tender excerpts were less often classified as *happy*. These patterns support the previous evidence that spectral flux reflects the same phenomena that effect the perception of happiness and sadness.

The effects found by adding attack slope into the models were similar to adding spectral flux. This can be understood as reflection of the same phenomenon behind these features.

Chapter 4

Conclusions and Implications for Future Research

The results in this study showed that classification in emotion recognition, or generally in the field of MIR, can be improved by wrapper selection when the methods are evaluated in terms of classification rates, generalizability and interpretability of the produced models. Moreover, the used framework proved to give proper estimates of the optimal subset sizes for the classifiers under study as shown by the performances on the validation set and by the representativeness of the emotions in the misclassified excerpts. Especially the modification to the cross-indexing algorithm was found to significantly improve the performance of the framework in the task at hand compared to the original algorithm.

The results also indicated that simple classifiers perform rather well in comparison to the more complex SVM classifier. This exhibits somewhat contrasting view to the common assumption in the field that have stated the superiority of SVM in emotion and mood recognition. Wrapper selection with 10 Nearest Neighbors classifier and backward elimination was found to yield the most promising performance – 56.5% classification rate with only 4 features. Therefore more extensive analysis was conducted on the results with that classifier-search method combination. It was shown that perceptually relatively high level features – majorness and key clarity were most useful whereas most of the low-level timbral features in general were not found to increase the performance of the classifier. The effect of majorness and key clarity was congruent with their perceptual meanings.

Analysis of the emotions in the sets of misclassified excerpts provided an interesting device for a thorough understanding of the classification rates. This was made possible only by the detailed rating data in the ground truth – proper perceptual studies are the key behind proper classification studies.

The suggested framework is not intended to produce a single “best” feature subset for a given classifier but rather a set of subsets for deriving general conclusions about the features useful for a chosen classifier. It must be noted that suggesting a single optimal subset would be trivial both research-wise and application-wise. Considering the sensitivity of the problem, applicability of such subset would be highly hypothetical in any utilization out of this study.

Although the results showed promise for the used framework, the highest classification rates of 56.5% were not eminently high. The level of the obtained accuracy rates can be attributed to certain limitations of the study. The sets of only 32 excerpts used in training and testing were rather small in order to account for all important aspects related to emotions in the analyzed music style, possibly reducing the classification rates and causing high variance in the results obtained with different splits. Moreover, the standardization of the data was based on only 32 excerpts. To improve the classification rates and variance in the results, analysis of a large collection of music would be needed to enable more generally valid standardization.

Although the used framework reduces the problems related to the irrelevancy of the input features, it is clear that the performance of the framework is still essentially limited by the degree at which the features capture perceptually relevant information. Therefore the performance of the framework could be improved by studying the representativeness and reliability of the features by modifying certain parameters in the feature extraction such as the frame length or filtering.

The classification performance could also be increased by evaluations with different classifiers. The obvious choice would be to run the analysis with different k-values for the k-NN classifier to find whether the performance of k-NN could be improved. Also other classifiers such as neural networks and GMM or multilabel classification (Trohidis et al., 2008) to overcome the problem of the ambiguousness of the expressed emotions could be used. However, one must acknowledge that the pursuable accuracies of classification according to perceptual categories are always bounded by the certainty at which humans recognize the concepts – previous research has reported 80% accuracies in the human recognition of clearly expressed emotions in music (Gosselin et al., 2005; Dellacherie et al., 2008).

Another interesting direction for further research would be to build ensemble classifiers by combining the models produced by the framework. The figure 4.1 shows the validation set classification accuracies of an ensemble classifier that combines the models of each subset size. The class for each excerpt is determined by the maximum of the summed class-prior probabilities, or classification scores (Zadrozny and Elkan, 2002) produced by the 30 models. The ensemble was able to increase the validation set classification accuracies of k-NN and

SVM to the level of 70%. Surprisingly, comparing ensembles with different subset sizes the optimal subset sizes for both k-NN and SVM were found to be the optimal also for the ensemble.

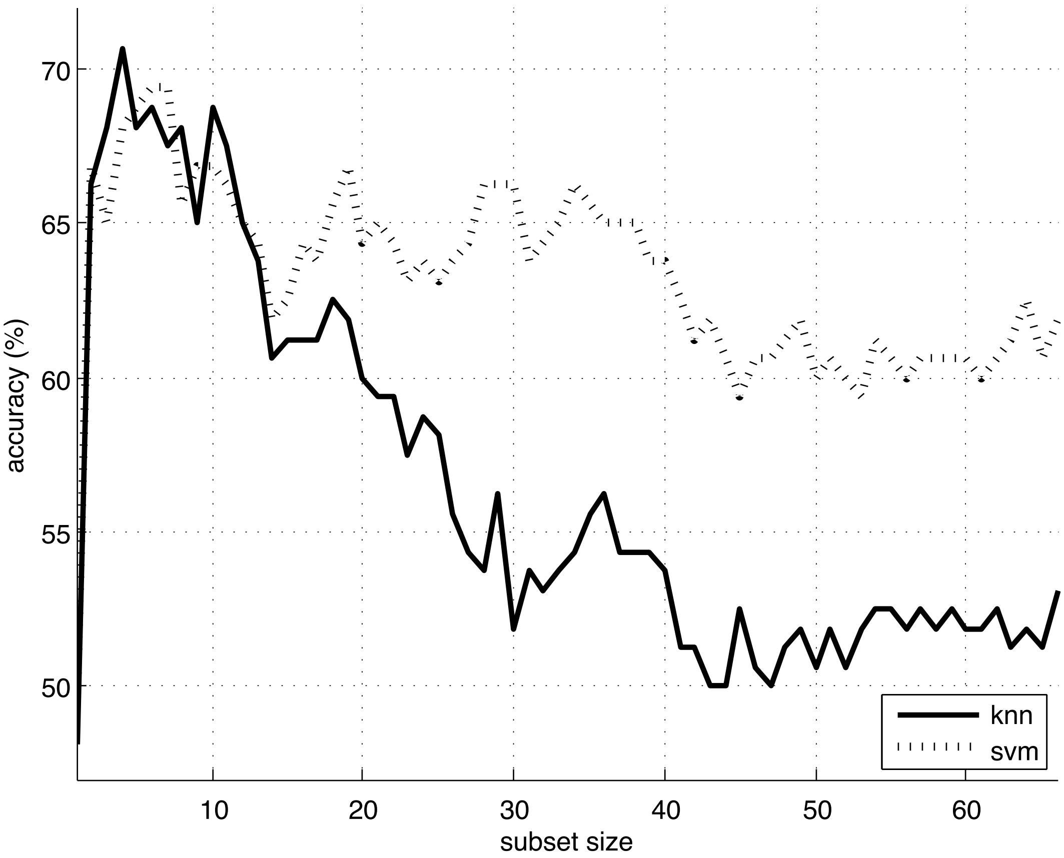


Figure 4.1: Ensemble accuracies.

Bibliography

Aha, D. and Kibler, D. (1991). Noise-tolerant instance-based learning algorithms. In *Machine Learning*, volume 6, pages 37–66.

Alluri, V. and Toiviainen, P. (2009). In search of perceptual and acoustical correlates of polyphonic timbre. In *Proceedings of the 7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM 2009).*

Alonso, M., David, B., and Richard, G. (2003). A study of tempo tracking algorithms from polyphonic music signals. In *Proceedings of the 4th. COST 276 Workshop.*

Aucouturier, J., Defreville, B., and Pachet, F. (2007). The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. *The Journal of the Acoustical Society of America*, 122:881–891.

Aucouturier, J.-J. and Pachet, F. (2004). Improving timbre similarity: How high is the sky? *Journal of Negative Results in Speech and Audio Sciences*, 1(1).

Baum, D. (2006). Emomusic - classifying music according to emotion. In *Proceedings of the 7th Workshop on Data Analysis (WDA2006).*

Bellman, R. E. (1961). *Adaptive Control Processes: A Guided Tour.* Princeton University Press.

Blum, A. L. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271.

Burnham, K. P. and Anderson, D. R. (2004). Multimodel inference: Understanding aic and bic in model selection. *Sociological Methods Research*, 33:261–304.

Cao, C. and Li, M. (2009). Thinkit's submissions for mirex2009 audio music classification and similarity tasks.

Caruana, R. and Freitag, D. (1994). How useful is relevance? Technical report, Relevance, Papers from the 1994 AAAI Fall Symposium.

Collins, N. (2006). Investigating computational models of perceptual attack time. In *Preceedings of the International Conference on Music Perception and Cognition.*

Cooper, M. and Foote., J. (2001). Media segmentation using self-similarity decomposition. In *Proceedings of SPIE Storage and Retrieval for Multimedia Databases*, volume 5021, pages 167–175.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. The MIT Press, 2nd edition.

Cunningham, P. (2000). Overfitting and diversity in classification ensembles based on feature selection. Technical report, Department of Computer Science, Trinity College Dublin.

Cunningham, P. (2008). *Machine Learning Techniques for Multimedia*, chapter Dimension Reduction, pages 91–112. Springer Berlin Heidelberg.

Dellacherie, D., Ehrlé, N., and Samson, S. (2008). Is the neutral condition relevant to study musical emotion in patients? is the neutral condition relevant to study musical emotion in patients? *Music Perception*, 25(4):285–294.

Devaney, M. and Ram, A. (1997). Efficient feature selection in conceptual clustering. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 92–97.

Duda, R. O. and Hart, P. E. (2001). *Pattern classification*. DG Stork.

Eerola, T., Lartillot, O., and Toiviainen, P. (2009). Prediction of multidimensional emotional ratings in music from audio using multivariate regression models. In *10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, pages 621–626.

Eerola, T. and Vuoskoski, J. (2009). A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music (submitted)*.

Ekman, P. (1992). An argument for basic emotions. *Cognition & Emotion*, 6:169–200.

Fabiani, M. and Friberg, A. (2008). Rule-based expressive modifications of tempo in polyphonic audio recordings. *Lecture Notes In Computer Science*, 4969:288–302.

Fastl, H. (1982). Fluctuation strength and temporal masking patterns of amplitude-modulated broadband noise. *Hearing Research*, 8:59–69.

Feng, Y., Zhuang, Y., and Pan, Y. (2003). Popular music retrieval by detecting mood. In *Proceedings of the 26th annual international ACM SIGIR conference*, Toronto.

Fiebrink, R. and Fujinaga, I. (2006). Feature selection pitfalls and music classification. In *Proceedings of the 7th International Conference on Music Information Retrieval*, pages 340–341.

Fiebrink, R., McKay, C., and Fujinaga, I. (2005). Combining d2k and jgap for efficient feature weighting for classification tasks in music information retrieval. In *Proceedings of the 2005 International Conference on Music Information Retrieval*, pages 510–513.

Friedman, J. H. (1997). On bias, variance, 0/1-loss, and the curse of dimensionality. *Data Mining and Knowledge Discovery*, 1:1.

Gabrielsson, A. and Lindström, E. (2001). *Music and Emotion*, chapter The influence of Musical Structure on Emotional Expression, pages 223–248. Oxford University Press.

Goebl, W. and Dixon, S. (2001). Analysis of tempo classes in performances of mozart sonatas. In *Proceedings of VII International Symposium on Systematic and Comparative Musicology and III International Conference on Cognitive Musicology*, pages 65–76.

Gómez, E. (2006). Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing*, 18(3):294–304.

Gordon, J. (1987). The perceptual attack time of musical tones. *The Journal of the Acoustical Society of America*, 82(1):88–105.

Gosselin, N., Peretz, I., Noulhiane, M., Hasboun, D., Beckett, C., Baulac, M., and Samson, S. (2005). Impaired recognition of scary music following unilateral temporal lobe excision. *Brain*, 128(3):628–640.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.

Hall, M. A. (1998). *Correlation-based Feature Selection for Machine Learning*. PhD thesis, University of Waikato, New Zealand.

Han, J. and Kamber, M. (2001). *Data mining : concepts and techniques*. San Francisco (CA) : Morgan Kaufmann.

Harte, C., Sandler, M., and Gasser, M. (2006). Detecting harmonic change in musical audio. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 21–26.

Hastie, T. and Tibshirani, R. (1998). Classification by pairwise coupling. In Jordan, M. I., Kearns, M. J., and Solla, S. A., editors, *Advances in Neural Information Processing Systems*, volume 10.

He, X., Cai, D., and Niyogi, P. (2006). Laplacian score for feature selection. In *Advances in Neural Information Processing Systems*, pages 507–514.

Hu, X., Bay, M., and Downie, J. (2007). Creating a simplified music mood classification ground-truth set. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR' 07)*.

Hu, X. and Downie, J. S. (2007). Exploring mood metadata: relationships with genre, artist and usage metadata. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR' 07)*.

Hu, X., Downie, J. S., Laurier, C., Bay, M., and Ehmann, A. F. (2008). The 2007 mirex audio mood classification task: Lessons learned. pages 462–467.

Hunt, M., Lennig, M., and Mermelstein, P. (1980). Experiments in syllable-based recognition of continuous speech. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '80.*, volume 5, pages 880–883.

Izmirli, O. (2000). Using a spectral flatness based feature for audio segmentation and retrieval. In *1st International Conference on Music Information Retrieval*.

Jensen, D. and Cohen, P. R. (2000). Multiple comparisons in induction algorithms. *Machine Learning*, 38:309–338.

Jensen, K. (1999). *Timbre Models of Musical Sounds*. PhD thesis, Department of Computer Science, University of Copenhagen.

John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *International Conference on Machine Learning*, pages 121–129.

John, G. H. and Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo*, pages 338–345.

Juslin, P. (1997). Emotional communication in music performance: A functionalist perspective and some data. *Music Perception*, 14(4):383–418.

Juslin, P. and Laukka, P. (2004). Expression, perception, and induction of musical emotions: A review and a questionnaire study of everyday listening. *Journal of New Music Research*, 33:217–238.

Juslin, P. N. (2000). Cue utilization in communication of emotion in music performance: Relating performance to perception. *Journal of Experimental Psychology: Human Perception and Performance*, 26(6):1797–1813.

Keller, J. (1985). A fuzzy k-nearest neighbor algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(4):581.

Klapuri, A. (1999). Sound onset detection by applying psychoacoustic knowledge. In *Proceedings of the Acoustics, Speech, and Signal Processing, 1999*.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324.

Krimphoff, J., McAdams, S., and Winsberg, S. (1994). Caractérisation du timbre des sons complexes. ii : Analyses acoustiques et quantification psychophysique. *Journal de Physique*, 4:625–628.

Krumhansl, C. L. (1990). *Cognitive Foundations of Musical Pitch*. Oxford University Press, New York.

Langley, P. and Sage, S. (1994). Induction of selective bayesian classifiers. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 399–406. Morgan Kaufmann.

Lartillot, O., Eerola, T., Toiviainen, P., and Fornari, J. (2008). Multi-feature modeling of pulse clarity: Design, validation, and optimization. In *International Conference on Music Information Retrieval*.

Lartillot, O. and Toiviainen, P. (2007). Mir in matlab (ii): A toolbox for musical feature extraction from audio. In *Proceedings of the International Conference on Music Information Retrieval, Wien, Austria*.

Laurier, C., Lartillot, O., Eerola, T., and Toiviainen, P. (2009). Exploring relationships between audio features and emotion in music. In *7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM 2009)*.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791.

Leman, M. (2000). Visualization and calculation of the roughness of acoustical musical signals using the synchronization index model (sim). In *Proceedings of the COST G-6 Conference of Digital Audio Effects*.

Leman, M., Vermeulen, V., De Voogdt, L., Moelants, D., and Lesaffre, M. (2005). Prediction of musical affect using a combination of acoustic structural cues. *Journal of New Music Research*, 34(1):39–67.

Li, T. and Ogihara, M. (2003). Detecting emotion in music. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 239–240.

Li, T. and Ogihara, M. (2006). Toward intelligent music information retrieval. *IEEE Transactions on Multimedia*, 8(3):564–574.

Lidy, T. and Rauber, A. (2006). Mirex 2006: Computing statistical spectrum descriptors for audio music similarity and retrieval. In *MIREX 2006 - Music Information Retrieval Evaluation eXchange*.

Liu, H. and Motoda, H. (1998). *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, Boston, USA.

Logan, B. (2000). Mel frequency cepstral coefficients for music modeling. In *International Symposium on Music Information Retrieval*, volume 28.

Loughrey, J. and Cunningham, P. (2005). Using early-stopping to avoid overfitting in wrapper-based feature selection employing stochastic search. Technical report, Department of Computer Science, Trinity College Dublin.

Lu, L., Liu, D., and Zhang, H.-J. (2006). Automatic mood detection and tracking of music audio signals. *IEEE Trans.on Audio, Speech, and Language Processing*, 14(1):5–18.

Meyer, L. B. (1956). *Emotion and Meaning in Music*. The University of Chicago Press.

Muyuan, W., Naiyao, Z., and Hancheng, Z. (2004). User-adaptive music emotion recognition. In *Signal Processing, 2004. Proceedings. ICSP '04. 2004 7th International Conference on*, volume 2, pages 1352–1355 vol.2. ID: 1.

Nabney, I. (2002). Netlab: Algorithms for pattern recognition. *Springer Advances In Pattern Recognition Series*.

Pampalk, E., Rauber, A., and Merkl, D. (2002). Content-based organization and visualization of music archives. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 570 – 579.

Pauws, S. (2004). Musical key extraction from audio. In *Proceedings of International Conference on Music Information Retrieval*.

Peeters, G. (2008). A generic training and classicication system for mirex08 classification tasks: Audio music mood, audio genre, audio artist and audio tag.

Platt, J. C. (1999). *Advances in kernel methods: support vector learning*, chapter Fast training of support vector machines using sequential minimal optimization, pages 185–208. MIT Press.

Pohle, T., Pampalk, E., and Widmer, G. (2005). Evaluation of frequently used audio features for classification of music into perceptual categories. In *Proceedings of the Fourth International Workshop on Content-Based Multimedia Indexing*.

Pudil, P., Ferri, F. J., Novovicova, J., and Kittler, J. (1994). Floating search methods for feature selection with nonmonotonic criterion functions. In *Proceedings of the Twelveth International Conference on Pattern Recognition, IAPR*, pages 279–283.

Rauhala, J., Lehtonen, H.-M., and Välimäki, V. (2007). Fast automatic inharmonicity estimation algorithm. *Journal of the Acoustical Society of America*, 121(5):184–189.

Reunanen, J. (2003). Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3:1371–1382.

Reunanen, J. (2004). A pitfall in determining the optimal feature subset size. In *Proceedings of the Fourth International Workshop on Pattern Recognition in Information Systems (PRIS 2004)*.

Reunanen, J. (2006). Less biased measurement of feature selection benefits. *Subspace, Latent Structure and Feature Selection: Statistical and Optimization Perspectives Workshop, SLSFS 2005, Bohinj, Slovenia, February 23-25, 2005*, 3940:198–208.

Reunanen, J. (2007). Model selection and assessment using cross-indexing. In *Proceedings of the Twentieth International Joint Conference on Neural Networks (IJCNN 2007)*, pages 2581–2585.

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178.

Scheirer, E. and Slaney, M. (1997). Construction and evaluation of a robust multifeature speech/music discriminator. In *ICASSP '97: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)-Volume 2*, pages 1331–1334.

Sethares, W. A. (1998). *Tuning, Timbre, Spectrum, Scale*. Springer-Verlag London.

Shannon, C. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, Illinois.

Silla Jr, C., Koerich, A., and Kaestner, C. (2008). Feature selection in automatic music genre classification. In *Proceedings of the 2008 Tenth IEEE International Symposium on Multimedia*, pages 39–44. IEEE Computer Society Washington, DC, USA.

Skowronek, J., McKinney, M., and ven de Par, S. (2006). Ground-truth for automatic music mood classification. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR' 06)*, pages 395–396.

Slaney, M. (1998). Auditory toolbox version 2. Technical report 1998-010, Interval Research Corporation.

Sloboda, J. A. and Juslin, P. N. (2001). *Music and Emotion*, chapter Psychological Perspectives on Music and Emotion, pages 71–104. Oxford University Press.

Thayer, R. E. (1989). *The Biopsychology of Mood and Arousal.* Oxford University Press, New York, USA.

Thompson, W. and Robitaille, B. (1992). Can composers express emotions through music? *Empirical Studies of the Arts*, 10(1):79–89.

Toh, A. M., Togneri, R., and Nordholm, S. (2005). Spectral entropy as speech features for speech recognition. In *Proceedings of PEECS*, pages 22–25.

Tolonen, T. and Karjalainen, M. (2000). A computationally efficient multipitch analysis model. *IEEE Transactions on Speech and Audio Processing*, 8(6):708–716.

Tran, D., Wagner, M., and Zheng, T. (1999). Fuzzy nearest prototype classifier applied to speaker identification. In *Proceedings of the European Symposium on Intelligent Techniques.*

Trohidis, K., Tsoumakas, G., Kalliris, G., and Vlahavas, I. (2008). Multilabel classification of music into emotions. In *Proc. 9th International Conference on Music Information Retrieval (ISMIR 2008), Philadelphia, PA, USA, 2008.*

Tzanetakis, G. (2007). Marsyas submissions to mirex 2007.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302.

Vafai, H. and Jong, K. D. (1992). Genetic algorithms as a tool for feature selection in machine learning. In *Fourth International Conference on Tools with Artifcial Intelligence.*

Vapnik, V. N. (1998). *Statistical Learning Theory.* Wiley Interscience.

Vesanto, J. (1999). Self-organizing map in matlab: the som toolbox. In *MatLab DSP Conference*, pages 35–40.

Wieczorkowska, A., Synak, P., Lewis, R., and W.Ras, Z. (2005). *Foundations of Intelligent Systems*, chapter Extracting Emotions from Music Data, pages 456–465. Springer.

Wieczorkowska, A., Synak, P., and Zbigniew, R. (2006). Multi-label classification of emotions in music. In *Proceedings of Intelligent Information Processing and Web Mining*, pages 307–315.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical machine learning tools with Java implementations.* Morgan Kaufmann, San Francisco, 2nd edition.

Wold, E., Blum, T., Keislar, D., and Wheaton, J. (1996). Content-based classification, search, and retrieval of audio. *IEEE Multimedia*, 3(2):27–36.

Wolf, L. and Shashua, A. (2005). Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *Journal of Machine Learning Research*, 6:1855–1887.

Wundt, W. (1897). *Outlines of psychology.* Lepzig: Englemann. trans. C. H. Judd.

Yang, Y. H., Lin, Y. C., Su, Y. F., and Chen, H. H. (2008). A regression approach to music emotion recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):448–457.

Yang, Y.-H., Liu, C.-C., and Chen, H. H. (2006). Music emotion classification: a fuzzy approach. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 81–84, New York, NY, USA. ACM.

Yaslan, Y. and Cataltepe, Z. (2006). Audio music genre classification using different classifiers and feature selection methods. In *18th International Conference on Pattern Recognition (ICPR 2006)*, volume 2.

Zadrozny, B. and Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699. ACM New York, NY, USA.

Zentner, M., Grandjean, D., and Scherer, K. (2008). Emotions evoked by the sound of music: Characterization, classification, and measurement. *Emotion*, 8(4):494–521.

Zentner, M. R., Meylan, S., and Scherer, K. R. (2000). Exploring musical emotions across five genres of music. In *Sixth Conference of the International Society for Music Perception and Cognition (IMPC)*.

Appendix A

Feature extraction in MIRtoolbox

```
% a is the audio excerpt to be analyzed

r = mirstruct;

%% DEFINE ANALYSIS WINDOW SIZES

    lwinlen = .0464;
    hwinlen = 2;
    struct_winlen = .1;

%% COMPUTE TEMPORARY FEATURES

    onsets = mironsets(a,’Filterbank’,15,’Contrast’,0.1);
    attacks = mironsets(onsets,’Attacks’);
    fluctuation = mirfluctuation(a,’Summary’);
    frames = mirframe(a,lwinlen,.5);
    spectrum = mirspectrum(frames); % Hamming window as a default
    chromagram = mirchromagram(a,’Frame’,lwinlen,.5,’Wrap’,0,’Pitch’,0);
    chromagram2 = mirchromagram(a,’Frame’,hwinlen,.5);
    keystrengths = mirkeystrength(chromagram2);
    [k1 ks]=mirkey(keystrengths,’Total’,1);

%%  DYNAMICS

    r.dynamics.rms = mirrms(a,’Frame’,lwinlen,.5);
    r.dynamics.lowenergy = mirlowenergy(a,’ASR’);
    r.rhythm.attack.time = mirattacktime(attacks);
    r.rhythm.attack.slope = mirattackslope(attacks);
```

```
%% RHYTHM

    r.rhythm.eventdensity = mirED(onsets,'Option1');
    r.rhythm.fluctuation.peak = mirpeaks(fluctuation,'Total',1);
    r.rhythm.fluctuation.centroid = mircentroid(fluctuation);
    r.rhythm.tempo = mirtempo(a,'Frame',hwinlen,.5,'Autocor','Spectrum');
    r.rhythm.pulseclarity = mirpulseclarity(a,'Frame',hwinlen,.5);

%% PITCH

    r.pitch.pitch = mirpitch(a,'Frame',lwinlen,.5,'Tolonen');
    r.pitch.chromagram.centroid = mircentroid(chromagram);

%% HARMONY

    r.harmony.keyclarity = ks;
    r.harmony.mode = mirmode(keystrengths);
    r.harmony.hcdf = mirhcdf(chromagram2);
    r.harmony.spectentr = mirentropy(mirspectrum(a,'Collapsed','Min',40,'Smooth',70));
    r.harmony.roughness = mirroughness(spectrum);
    r.harmony.inharmonicity = mirinharmonicity(spectrum,'f0',r.pitch.pitch);

%% TIMBRE (Centroid)

    r.timbre_c.brightness = mirbrightness(spectrum, 'CutOff', 110);
    r.timbre_c.centroid = mircentroid(spectrum);
    r.timbre_c.zerocross = mirzerocross(frames);

%% TIMBRE (Shape)

    r.timbre_s.spread = mirspread(spectrum);
    r.timbre_s.skewness = mirskewness(mirspectrum(frames,'Max',2500));
    r.timbre_s.spectentropy = mirentropy(mirspectrum(frames,'Max',5000));
    r.timbre_s.spectralflux = mirflux(frames);
    r.timbre_s.flatness = mirflatness(mirspectrum(frames,'Max',5000));
    r.timbre_s.irregularity = mirregularity(spectrum);

%% STRUCTURE

    r.structure.repetition_spectrum = ...
      mirnovelty(mirspectrum(a,'Frame',struct_winlen,.5,'Max',5000),'Normal',0);
    r.structure.repetition_rhythm = ...
      mirnovelty(mirautocor(a,'Frame',struct_winlen,.5),'Normal',0);
    r.structure.repetition_tonality = ...
      mirnovelty(mirchromagram(a,'Frame',struct_winlen,.5),'Normal',0);
    r.structure.repetition_register = ...
      mirnovelty(mirchromagram(a,'Frame',struct_winlen,.5,'Wrap',0),'Normal',0);

%%  MFCC

    r.mfcc=mirmfcc(spectrum,'Rank',7);
    r.dmfcc=mirmfcc(spectrum,'Delta',1,'Rank',7);

%%  STATISTICS

    r = mirstat(r);
```