Technical note PR-TN-2010/00115

Issued: 09/2010

Soft computing based feature selection for environmental sound classification

A. Shakoor, T. M. May, and N. H. van Schijndel

Philips Research Eindhoven

Concerns: End Report

Period of Work: April 2009 - April 2010

Notebooks: None

Authors' address A. Shakoor aash06@post.student.bth.se

T. M. May tobias.may@mail.uni-oldenburg.de

N. H. van Schijndel nicolle.van.schijndel@philips.com

Title: Soft computing based feature selection for environmental sound classification

Author(s): A. Shakoor, T. M. May, and N. H. van Schijndel

Reviewer(s): A. Harma, S. Srinivasan, and IPS.Facilities

Technical Note: PR-TN-2010/00115

Additional Numbers:

Subcategory:

Project:

Customer: Philips Research

Keywords: environmental sound classification, feature selection, rough/fuzzy set theory, pattern recognition, soft computing

Abstract: Environmental sound classification has a wide range of applications, like hearing aids, mobile communication devices, portable media players, and auditory protection devices. Sound classification systems typically extract features from the input sound. Using too many features increases complexity unnecessarily and can even reduce performance, but proper selection of features is a non-trivial task. It is promising to base such selection on soft computing, because this approach can handle uncertain information in an efficient way, using simple set theoretic functions, and because this approach is close to perception-based reasoning. Therefore, this thesis investigates different feature selection methods, including soft computing methods and classical information, entropy and correlation-based approaches. Results show that a soft computing based feature selection method performs best in terms of number of features selected, recognition rate and consistency of performance. In addition, the resulting classification system is robust for reverberation.

Conclusions:

- Proper selection of features clearly improves recognition performance.
- Among the seven evaluated methods, two are most promising: the soft computing method named rough set neighborhood model 1 (RS1), and the classical method named stepwise forward selection (SFS). Experiments show that RS1 gives 92.5 % recognition rate while selecting on average only 17 features out of the investigated 800 features. SFS gives a comparable recognition rate of 92.4 %, but selects many more features, i.e. 118 features on average.
- Among the 800 tested features, SNR, tonality, and the MFCCs are good features to discriminate between the classes "music," "speech," "noise," and "speech+noise."
- The confusion matrices show that the class "speech+noise" is the most difficult to classify, and the class "speech" the easiest.
- The validation results show that the proposed system gives robust performance for reverberated speech material, with a slight (1.5 %) reduction in performance for high reverberation time.
- Adding the class-dependent likelihoods across multiple frames is a good way to integrate information over a longer observation time. Accumulating likelihood probabilities of five frames of 1 s, the recognition rate increases to 99.7 % with RS1 and 98.4 % with SFS.
- If the training and the testing data set are randomly selected, there is no need to run the feature selection method for all five partitions of data independently.

Contents

Chapter 1

Introduction

In everyday life we are often in complex acoustic environments, where we are surrounded by acoustic mixtures consisting of various sound types. Whereas the human auditory system can instantly discriminate between different types of sound, e.g. speech and background noise, this task is not so trivial for computational systems. This thesis describes a computational system that is able to classify environmental sounds. This system may or may not follow the psychoacoustic models of the human auditory system.

Pattern recognition theory can be a good approach to the sound classification problem. In pattern recognition, objects (sound classes in this case) are identified on the basis of some attributes (features) of objects [1]. The selection of a set of features that is capable of distinguishing between classes is the most critical step in audio classification system design [2].

There is a wide range of potential applications of environmental sound classification. Particular applications of this system include intelligent wearable and handheld devices. A device can then adjust itself automatically to a mode of operation that is better suited to the specific acoustic situation. For example, a mobile phone may choose automatically to ring faintly when the user is in a quiet situation and ring loudly when the user is in a noisy situation. Another application could be to modify the processing parameters of a device according to the content of audio. For example, hearing instrument users desire different instrument settings in different acoustic situations. An automatic classification system can be used to automatically switch to an instrument setting that best fits the acoustic scenario.

1.1 State of the art classification system

The structure of a basic sound classification system is shown in figure 1.1. This structure comprises feature extraction, feature selection and pattern learning as important modules. These modules are briefly described in the following subsections.

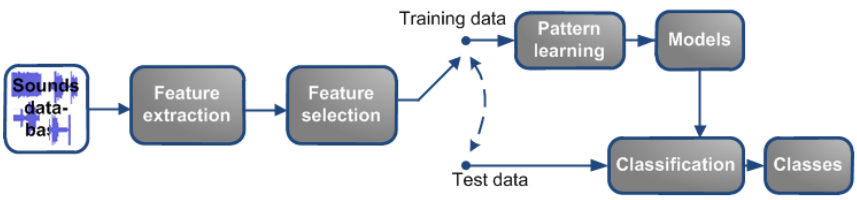

Figure 1.1: State of the art classification system.

1.1.1 Feature extraction

A digital representation of an audio signal contains a huge amount of data. For example one second of 16-bit mono audio signal with a sampling frequency of 44.1 kHz, contains 88.2 kB of data. From computational point of view, it is usually not a good idea to process that amount of data directly. Statistical and analytical processing of audio data is also needed during the processes of classification. Numerical representation of audio signal facilitates efficient processing and analysis of data. Therefore, a numerical representation of the audio signal is required. The audio signal is transformed into numerical features or so called feature vectors. This task is performed in the feature extraction block of the system. This block breaks the audio signal in short time frames and extracts numerical features from these frames. These numerical values or features may be time domain features such as tempo and silence ratio or frequency domain features such as pitch and sub-band energy. More information about audio features is given in section 3.2.2. For further analysis, statistics (such as mean, variance, skewness) and delta-features (first order and higher order derivatives) of these features are calculated.

1.1.2 Dimensionality reduction and features selection

There are many features that might be used to describe the properties of an audio sequence. One could expect that using a larger number of features increases the discriminability between different audio classes, but this is not always the case. Including too many features for the training of a classifier may lead it to find spurious patterns, which are not valid in general. This is called the so-called curse of dimensionality [3]. The cost of computation and curse of dimensionality demands us to reduce the dimensionality of the data. For these reasons the feature extraction module is followed by the dimensionality reduction module. Selection of superfluous features or reducing dimensionality by losing information can have a negative impact on performance of the system. On the other hand, a good combination of selected features may enhance the performance even with a simple classifier [4].

1.1.3 Pattern learning, modeling and classification

Finally, the classifier is applied on the reduced data set to distinguish and classify different classes of sounds. Various pattern recognition algorithms are available for pattern learning and classification. These algorithms include minimum distance and Bayes classifiers, K-nearest neighbor classifiers, neural networks, Gaussian mixture models, hidden Markov models, and rule-based approaches. Based on a given feature vector, the task of the classifier is to predict the corresponding audio class. Since the classification approach presented in this thesis[1] requires supervised learning (learning based on already known class labels), there must be training data (with known class labels) to learn the class models. Once the training of the algorithm is completed, the test data is applied to test the learned models. The classification module may also consist of different stages, for example a neural network for coarse classification and a rule-based approach for the fine classification.

1.2 Research questions and objectives

Although much research has been conducted in the area of acoustic signal classification in the last decade, the following points require additional research.

[1] Gaussian mixture model (GMM) classifier; details in section 3.4.

- Which algorithm or method is the better choice for the problem of dimensionality reduction or feature selection for environmental sound classification?
- Soft computing techniques (fuzzy/rough-set-based techniques) are only partially explored for acoustic feature selection. In [5], rough-set-based feature selection is studied for automatic recognition of musical instruments and musical styles. A rough set approach for the classification of sound signals produced by swallowing, is presented in [6]. Soft computing techniques are studied for musical acoustics in [7].

The overall objective of this research is to develop a computational system that is capable of classifying different acoustic signals. In particular the aim of this thesis is set to explore the feature selection module in detail, with emphasis on soft computing based feature selection methods. This module clarifies the problem view and has great impact on overall performance of the system.

1.3 Organization of report

The rest of this thesis is organized as follows. Chapter 2 describes a literature overview. Chapter 3 is dedicated to describe the system design and experimental setup. Chapter 4 shows the experimental results. Finally chapter 5 presents the summary and concludes the report.

Chapter 2

Literature overview

The purpose of this chapter is to give an overview of the background knowledge for the problem of feature selection, and to introduce soft computing techniques.

2.1 Dimensionality reduction and feature selection

As explained in the previous chapter, in audio signal classification problems, data is represented in the form of numerical valued data vectors that are called feature vectors. These vectors exhibit high dimensionality. For huge amount of data, the process may become computationally infeasible for certain systems. Also, the curse of dimensionality reduces performance of the system. For these reasons it is necessary to reduce the dimensionality of the dataset. The key role of this reduction is to reduce data size while preserving useful information present in the original data by discarding redundant information. There are a number of methods to approach this problem. A taxonomy of dimensionality reduction methods is presented in figure 2.1 [8]. There are two major classes of data reduction techniques:

1. Transformation-based techniques,

2. Selection-based techniques.

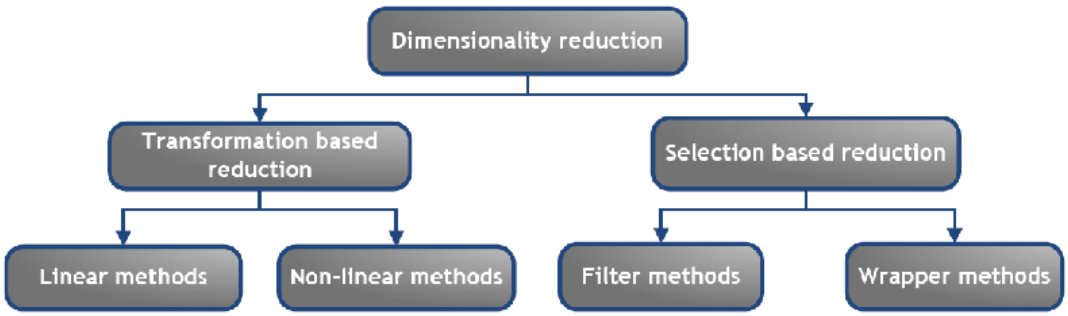

Figure 2.1: A taxonomy of dimensionality reduction.

2.1.1 Transformation-based dimensionality reduction

In transformation-based methods the original data is transformed to produce new data values. These techniques extract new features using various different combinations of the original features. Depending on how the original features are combined, the transformation-based methods can be divided into two classes:

1. Linear transformation,
2. Non-linear transformation.

Linear transformation-based methods

These methods transform the original features into a new feature set by using linear combinations of the original features. These methods include principal component analysis (PCA), linear discriminant analysis (LDA) and projection pursuit (PP). Principal component analysis is based on the assumption that the features with a large variance provide more information. The PCA method therefore orders features with respect to variance. This is achieved by finding eigenvectors of the covariance matrix, and then a linear transformation is done by matrix multiplication. In LDA, the ratio of inter-class variance to the intra-class variance is maximized for a particular dataset, thereby guaranteeing maximal separability.

Non-linear transformation-based methods

Most non-linear methods are extensions of linear methods. For example, Kernel-PCA is an extension of PCA and Kernel discriminant analysis (KDA) is an extension of LDA. In kernel methods, data is mapped to reduced dimension by using kernel functions.

2.1.2 Selection-based dimensionality reduction

Unlike the transformation-based methods, selection-based methods do not transform the original data. In these methods only a useful subset of the original features is chosen based on some criterion. The aim of selection-based methods is to select the subset of features with the least number of elements that preserves useful information present in the original set of features and reduces redundant information. There are different criteria to measure relevancy and redundancy; for example, the correlation of a feature with other features indicates the information redundancy. In simple words, feature selection is the search for features that are less associated with each other and more associated with the decision classes. With increasing data dimensionality, the number of features N increases, and finding the optimal subset of features becomes intractable [9]. It has been shown that the feature selection problem is NP-hard [10][1]. The overall process for feature selection is shown in figure 2.2. In a feature selection process there are typically the following three steps: subset generation, evaluation of generated subset, and stopping criteria [11].

Subset generation

Subset generation is a search procedure that produces feature candidates for evaluation. There are many ways to start this search. For example, one can start with no features in the search

[1] NP-hard (non-deterministic polynomial-time hard) is a complexity class of problems that are intrinsically harder than those that can be solved by a nondeterministic Turing machine in polynomial time.

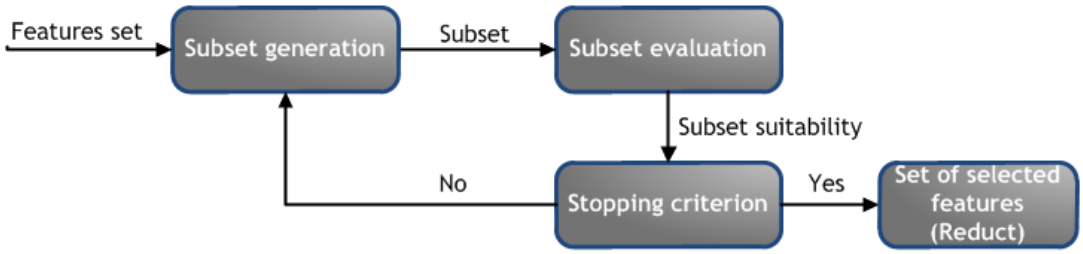

Figure 2.2: Feature selection procedure.

space, all features or a randomly selected subset of features. The choice of starting point effects the search direction. If the starting point is an empty set, the features are added iteratively and this search method is called *forward search*. If one starts with all features and removes features iteratively, this is called *backward search*. In case of a random subset as starting point, features can be added, removed or produced randomly; this search is called *bidirectional search*.

Evaluation of the generated subset

The evaluation function calculates the suitability of the selected features and compares it with previously selected subsets. There are two major types of feature selection methods: filter methods and wrapper methods.

1: Filter methods

Filter methods use independent criteria for evaluation of a feature subset. Features are evaluated intrinsically on the basis of feature characteristics, i.e. independent of any classification algorithm. An independent evaluation or filter-based approach for feature selection is shown in figure 2.3.

Commonly used evaluation criteria for filter methods are explained briefly in the following paragraphs.

1a: Distance measures

In these evaluation criteria, a feature is preferred if it induces greater distance between classes. Two features are indistinguishable if the difference in distance is zero. There exist many distance measures that can be used as evaluation criterion (Bhattacharyya distance, Fisher information metric etc).

The Fisher index can be used as an interclass distance [12].

$$S = \frac{\sigma^2_{between}}{\sigma^2_{within}}$$

Figure 2.3: Filter approach for feature evaluation.

where $\sigma^2_{between}$ is the interclass variance and σ^2_{within} is the intraclass variance of a feature.

1b: Information measures

In these criteria, a measure of the information gain (also called cross entropy or mutual information) [13] is determined. Information gain is defined as the prior uncertainty and the expected posterior uncertainty using a feature. A feature is preferred if it offers more information gain than another feature.
Consider two discrete random variables x and y that take values in $\{1, ..., a\}$ and $\{1, ..., b\}$ respectively, and an independent and identically distributed random processes with samples $(x, y) \in \{1, ..., a\} \times \{1, ..., b\}$ drawn with joint probabilities p_{xy}. Mutual information is the measure of stochastic relation of x and y [13].

$$I(p) = \sum_{x=1}^{a} \sum_{y=1}^{b} p_{xy} log \frac{p_{xy}}{\sum_x p_{xy} \sum_y p_{xy}}$$

.

1c: Dependency measures

This is the measure of similarity or association of a feature. A feature is considered to be more suitable if it has greater association with the decision classes and less association with other features. A common measure of dependence between two features is the correlation coefficient. Considering two features X and Y with σ_x, σ_y as standard deviations and μ_x, μ_y as mean values respectively, the correlation coefficient$\rho_{x,y}$ between X and Y is defined as

$$\rho_{x,y} = \frac{E\left[(X - \mu_x)\left(Y - \mu_y\right)\right]}{\sigma_x \sigma_y}$$

.

1d: Soft Computing based measures

These techniques use a dependency measure and a significance measure of a feature defined by rough/fuzzy set theory. Details of these methods and measures are given in section 2.2.

2: Wrapper methods

These methods evaluate the selected feature set depending on the classification algorithm. Since these methods include accuracy of the result as evaluation criterion, as shown in figure 2.4, it is intuitive to think that these methods give better performance. Unfortunately, as pointed out in [14], this is often not the case when performance is evaluated with independent test data. The cross-validation score of the best subset is typically not representative of the classification accuracy for new data. This supports the usage of filter methods rather than computationally intense wrapper methods [14]. Other drawbacks of these methods include high computational cost and that the selected feature set may not be suitable for other classification algorithms, because the selection of features depends on the classification system.

Stopping criteria

After every iteration, a stopping criterion is evaluated. This criterion reveals whether to continue the selection process or not. For example, such a criteria may be to stop the selection process

when a certain number of features is selected, a maximum number of iterations is reached or a pre-defined significance level of a feature set is achieved. When the stopping criterion is satisfied the selection process terminates.

2.2 Feature selection based on soft computing

The term soft computing does not refer to a single field of computation, but it has many components. Dealing with incomplete or imperfect knowledge and reasoning under uncertain circumstances are the core of soft computing techniques. The term soft computing is applied to methods or algorithms that give a useful but sometimes in-exact solution for computationally hard problems. These methods mimic human intelligence. Fuzzy logic is a model of human reasoning in an imprecise environment; an artificial neural network is a model of the human brain and mimics the interconnecting neurons in the human brain. Different elements of soft computing complement each other. Major constituents of soft computing are rough set theory, fuzzy set theory and fuzzy logic. Basics of rough set theory, fuzzy set theory and feature selection based on these theories is presented in the following subsections.

2.2.1 Introduction to rough set theory

Rough set theory (RST) is an extension of conventional (classical or crisp) set theory. Basic definitions used in RST are explained below.

Information systems

Unlike classical set theory, in rough sets a dataset is presented in the form of a table. This table is a collection of rows and columns. Rows represent an object or input (audio frames in this case), and columns represent attributes (features in this case) that give the values for the given input frame. This table is called an *information system* [15]. In formal mathematical notation, an information system is a pair $\mathbf{A} = (U, A)$, where U is a non-empty finite set of objects called universe and A is a non-empty finite set of features such that, $a : U \rightarrow V_a$ for every $a \in A$. The set V_a is called the value set of a. If the outcome of the classification (for example, in this case it is known which frame is computed for which audio class) is known, this is represented by a distinguishing attribute called *decision attribute*. Information systems of this kind are called *decision systems*. More formally a decision system is an information system of the form $\mathbf{A} = (U, A \cup \{d\})$. Where d $\notin A$ is a *decision attribute*, and elements of A are called *condition attribute* [15]. An example of a decision system is given in table 2.1.

In this table, the most left column represents the audio objects, which will be called frames further on, which is more common terminology in audio classification. In this specific example, f_i is i^{th} frame of audio data. The most right column is the decision attribute, in this example

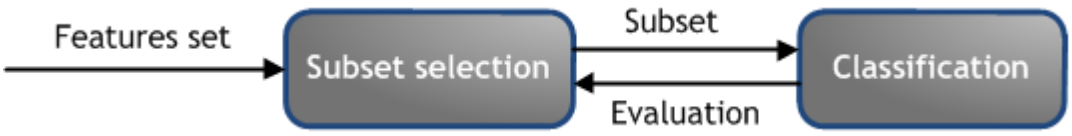

Figure 2.4: Wrapper approach for feature selection.

Object	Pitch	Tempo	Class
f_1	1	2	music
f_2	1	4	music
f_3	4	2	speech
f_4	3	4	speech
f_5	1	2	music
f_6	2	5	speech
f_7	3	4	speech+noise
f_8	5	4	speech+noise

Table 2.1: Example of a decision system.

representing the class labels for the corresponding frames. The rest of the columns represent values for the features pitch and tempo respectively.

Indiscernibility

Decision systems may be redundant in two ways:

1. the same objects may be present more than once,
2. some of the condition attributes may be superfluous.

Indiscernibility of two objects handles these issues as follows. Let $\mathsf{A} = (U, A)$ be an information system, then with any $B \subseteq A$ there is associated an equivalence relation $\text{IND}_A(B)$.

$$IND_A(B) = \left\{(x, x^{'}) \in U^2 | \forall a \in B \;\; a(x) = a(x^{'})\right\}$$

$IND_A(B)$ is called the B-indiscernibility relation [15]. If $(x, x^{'}) \in IND_A(B)$, then objects x and $x^{'}$ are indiscernible from each other by attributes from B. The equivalence classes of the B indiscernibility relation are denoted $[x]_B$, where the equivalence class of an element $x \in X$ consists of all objects $x^{'} \in X$.

In the information system of table 2.1, the non-empty set of condition attributes are $\{pitch\}$, $\{tempo\}$ and $\{pitch, tempo\}$. If one considers pitch, objects f_1, f_2, f_5 belong to the same equivalence class and are indiscernible. The following partitions of the universe can be made using the IND relation.

$$IND\,\{tempo\} = \{\{f_1, f_3, f_5\}, \{f_2, f_4, f_7, f_8\}, \{f_6\}\}$$

$$IND\,\{pitch\} = \{\{f_1, f_2, f_5\}, \{f_4, f_7\}, \{f_3\}, \{f_6\}, \{f_8\}\}$$

$$IND\,\{pitch, tempo\} = \{\{f_1, f_5\}, \{f_4, f_7\}, \{f_2\}, \{f_3\}, \{f_6\}, \{f_8\}\}$$

Set approximation

By applying an equivalence relation, partitions of the universe are found. These partitions can be used to build new subsets of the universe. The subsets of features that have the same value of

class labels are of most interest for repartitioning of the table. In most of the cases the concepts can not be defined crisply. For example, one can not crisply define the outcome "speech class" using the attributes in table 2.1. It is here that the idea of rough sets comes out. Concepts of rough sets can be defined as upper bounds and lower bounds. The *B-lower* bound is defined as the complete set of all elements that can be classified with certainty, as members of a X partition on base of knowledge in B. In mathematical notation

$$\underline{B}X = \{x|\,[x]_B \subseteq X\}$$

The *B-upper* bound is defined as the set of all elements that can not be classified as members of a X partition on base of knowledge in B. Formally

$$\overline{B}X = \{x|\,[x]_B \cap X \neq \phi\}$$

The *B-boundary region* is defined as the set of elements that might be classified (can neither be ruled in nor ruled out) as members of a X partition on the base of knowledge in B, mathematically

$$BN_B(X) = \overline{B}X - \underline{B}X.$$

A set is crisp if the boundary region is empty and is rough if the boundary region is non-empty. For example, in table 2.1, frames f_4 and f_7 have the same values for pitch and tempo; hence, the frames f_4 and f_7 can not be defined with certainty using the information of these features; f_4 and f_7 are lower bounds.

For feature selection, one is interested in finding a set of features that gives a high value in the positive region.

Feature dependency and significance

For B, $D \subset A$, D depends on B in a degree d $(0 \leq d \leq 1)$, if

$$\gamma_B(D) = \frac{\sum_{i=1}^{N} |\underline{B}D_i|}{|U|}$$

where $|\cdot|$ is the cardinality operator[2]. Significance of a feature a in a decision table is calculated by measuring the difference or change in dependency when a is removed from the considered set of features. For B, D and a feature $a \in B$. A feature is more significant if the change in dependency is increased. If significance is 0, the feature is insignificant. In formal notation significance is defined as

$$\sigma_B(D, a) = \gamma_B(D) - \gamma_{B-a}(D)$$

.

Reducts and cores

The goal of the feature selection process is to remove redundant or superfluous features, by finding the subset of features that preserves the indiscernibility relation. There are several such subsets. Subsets with the least number of features are called *reducts*, and features that are present in all subsets are called *cores*.

[2]Cardinality operator counts the members of a set.

2.2.2 Introduction to fuzzy set theory and fuzzy logic

If we need to describe human reasoning, bivalent-logic, that is logic based on two values 0 (*false*) and 1 (*true*), is inadequate. Fuzzy set theory and fuzzy logic address this point. Fuzzy set theory and fuzzy logic use the interval between 0 and 1 to describe (human perception-based) reasoning.[3].

Fuzzy sets

Definitions:
Let X be a space of points (objects), with a generic element of X denoted by x. A fuzzy set A in X is characterized by a membership function $\mu_A(x)$, which associates with each point in X a real number in the interval [0,1], where the value of $\mu_A(x)$ represents the *grade of membership* of x in A. More formally, if X is a collection of objects denoted generically by x, then a fuzzy set A in X is a set of ordered pairs:

$$A = \{(x, \mu_A(x))|x \in X\}$$

where $\mu_A(x)$ maps X to the membership space M ([0,1]) [17].

Example:
For example, take the statement "an audio fragment with a small silence ratio is music." If the value of silence ratio for an audio frame is 0.01, we may assign this statement a membership function 0.85. In set theory terminology this statement can be translated as "an audio fragment with a small value of silence ratio is a member of the set of music sounds":

$$\mu_{music}(audio) = 0.85$$

where μ is the membership function for audio to be a member of the set of music sounds.

2.3 Rough/Fuzzy-set-based feature selection

In rough-set-based feature selection or attribute reduction, feature dependencies and significance are used as evaluation criteria. As explained in the previous section, rough set theory is based on the equivalence classes. Rough-set-based algorithms are useful for nominal or discrete feature spaces. To use these methods for acoustic feature selection either their feature values have to be discretized or a neighborhood model has to be used, because acoustic features are continuous [18]. Brief descriptions of these two techniques are given below.

Neighborhood-based rough set model

These methods use a rough set model for nearest neighbor search. For an information system (feature dataset in this case), given arbitrary $x_i \in U$ and $B \subseteq C$, the neighborhood $\delta_B(x_i)$ of the x_i in the subspace B is defined as [19]

$$\delta_B(x_i) = \{x_j | x_j \in U, \Delta_B(x_i, x_j) \leq \delta\}$$

[3]Membership functions should not be confused with probabilities. Membership functions can be explained as "how much an element is in a set," whereas probabilities as "how probable it is that an element is in a set." Classical probability theory is a subset of the more generalized "possibility theory" [16]

where Δ is a metric function. There are different metric functions that can be used; for example, the Minkowsky function or the Manhattan function [20]. The δ defines the size of the neighborhood. The lower approximations can be used as measure of dependencies between condition and decision attributes. Fuzzy neighborhood is also used.

Feature discretization

There are many algorithms to discretize features. One of the methods is equal width interval binning. In this method values of features are sorted and the range of values is divided into k equally sized bins [21].

Chapter 3

System design

The overall design of the system is described in figure 3.1. In the following sections, each block of the system and the relevant parameters are described.

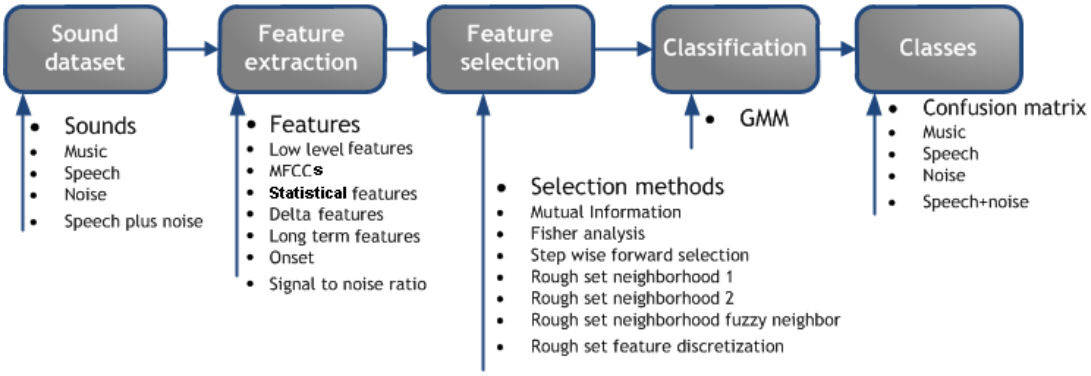

Figure 3.1: System design.

3.1 Sound database

The contents of the sound database have a great impact on the performance of the system. If there is an imbalance between the class representation of the sound classes, and for instance one of the classes is more dominantly represented in the database, this may bias the feature selection algorithm and the classifier. A critical selection of representative sound files for different sound classes therefore plays an important role in sound classification. In this study, the sound database contains four different classes of sounds, which are “music”, “speech”, “noise” and “speech+noise”. These four classes were chosen considering potential applications of the system in hearing aids and telecommunication devices [22]. For each of these four classes, there are 38, 24, 29 and 28 audio files respectively, with time lengths varying from 20 s to 60 s. Every audio file is stored as mono audio file of ‘.wav’ extension with a sampling frequency of 44.1 kHz and 16 bit resolution. Table 3.1 gives a summary for each class. Details are given in appendix A.

The sound database has to be partitioned into a training set and a testing set. This partitioning should be made such that each set covers each sound class fully. Since there is no clear rule for how to make this choice [22], random partitions of the database were made.

Class	Files	Length	Description
Music	38	1103 s	pop, rock, techno, classical, country, funk, electro
Speech	24	1205 s	male, female, Dutch, English
Noise	29	1225 s	babble, cafeteria, inside bus, machine, inside car, inside train, train station, busy street, shopping center, traffic, kindergarten, factory, supermarket
Speech+noise	28	1130 s	{male, female} + {wind, babble, inside train, train station, busy street, inside bus, machine, shopping center}

Table 3.1: Sound database.

3.2 Feature extraction

In this module of the classification system, the computation of the feature vectors is performed. The process of feature extraction and the feature dataset are explained in the following subsections.

3.2.1 Process of feature extraction

The overall process of feature extraction (computation of feature vectors) is shown in figure 3.2. With respect to the time length of the analysis window, the process of feature extraction is completed in two stages.

1. **Subframe analysis:** The original audio signal with a sampling frequency of 44.1 kHz is divided in segments of 20 ms with 50 % overlap. For each subframe, feature values are calculated, which are called subfeatures. Subframe analysis is the shortest analysis of the signal.

2. **Frame analysis:** After the computation of subfeatures, higher order statistics of these subfeatures are measured over a window length of 1 s without any overlap. These longer blocks are called frames. In addition to the statistics of subfeatures, there are long-term features, which are calculated directly on a frame basis (non-overlapping 1-s frames).

3.2.2 Feature dataset

A combination of statistics of the low-level features, statistics of the Mel-frequency cepstral coefficients (MFCCs) and the long-term features constructs a feature matrix or feature dataset. A complete list of all extracted features is given in appendix B. The description of the extracted features for this system is given below.

Subframe-based features

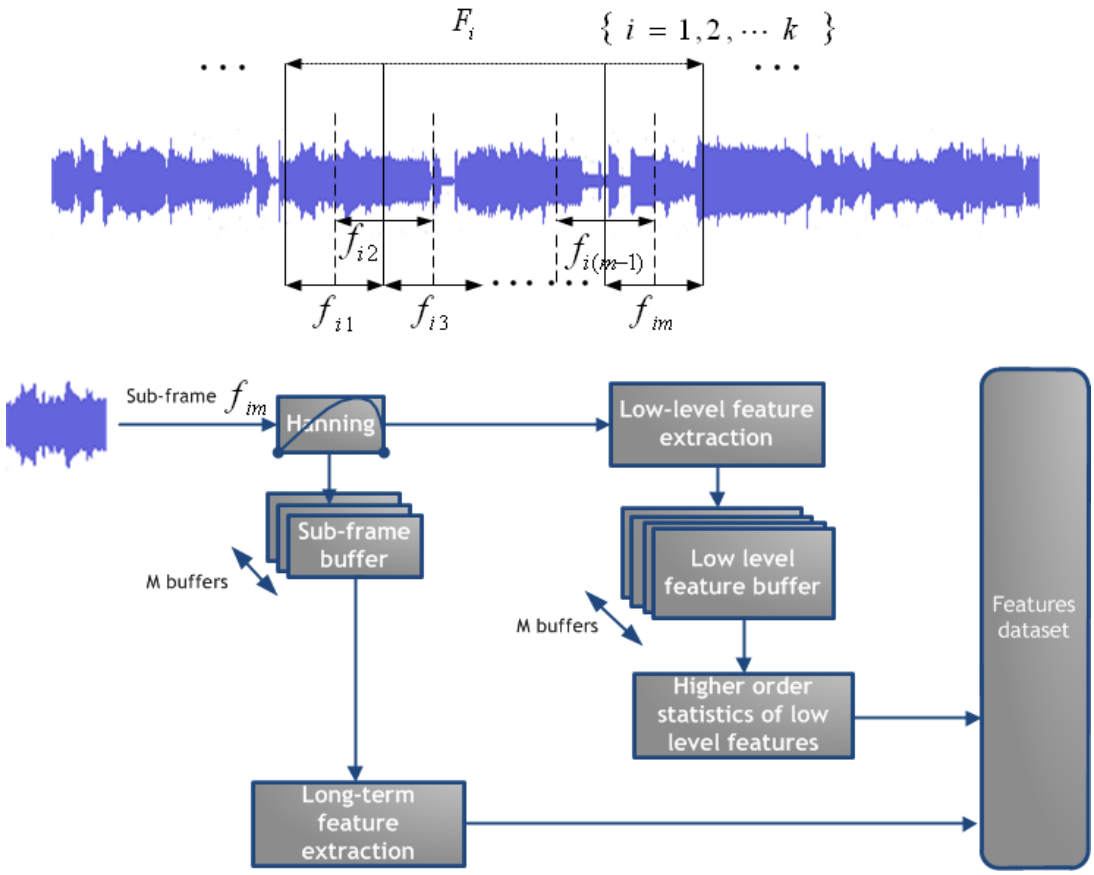

Figure 3.2: Feature extraction process.

Low-level features: 16 low-level features were calculated on a subframe basis with subframe parameters as explained in the previous subsection. Table 3.2 shows the 16 low-level features. References are given for mathematical details of the features.

Mel-frequency cepstral coefficients (MFCCs): There are also 13 MFCC coefficients with 25-ms window and hop size of 10 ms (40 % overlap). An auditory toolbox was used for MATLAB implementation of MFCCs [27]. Parameters were chosen according to HTK [28].

Delta features: After computing the set of low-level features and MFCC features, first and second order derivatives of these subfeatures were calculated.

Statistics of low-level features: For each low-level feature, MFCC and delta feature, nine statistical features were computed (mean, average deviation, standard deviation, variance, skewness, kurtosis, 10th, 50th and 90th percentile).

Frame-based features

Long-term features: There are 13 long-term features that are calculated by first applying a 20-ms Hanning window with 50 % overlap and then applying 1-s rectangular window (frame

Time domain low-level features	Frequency domain low-level features
Main peak of autocorrelation function	Spectrum band-energy ratio
Fundamental frequency estimation [23]	Spectrum spread [24]
Mean level fluctuation strength [22]	Spectrum irregularity [25]
RMS of frame energy in decibel [22]	High frequency content [25]
Zero-crossing rate	Spectrum bandwidth
Crest factor	Spectrum centroid
	Spectrum flatness [26]
	Spectrum roll-off [25]
	Spectrum entropy
	Spectrum flux [25]

Table 3.2: Time domain and frequency domain low-level features.

size). Long-term features in this system include five features based on amplitude histogram (histogram width, histogram symmetry, skewness, kurtosis and the histogram's lower half) [22], three features based on pitch track (tonality, pitch variance and delta of consecutive pitches) [23], two features based on periodicity (periodicity ratio and noisy frames ratio) [29], zero-crossing ratio [29], low short-time energy ratio [29], and fluctuation of spectral entropy.

Onset features: There are three onset features [22], which are also calculated on a frame basis.

Signal-to-noise ratio (SNR): SNR was calculated for each 1-s block.

By adding all these features, a total of 800 features are gained in the feature dataset or feature matrix, i.e, for each one second frame 800 features are calculated.

3.3 Feature selection

Looking at figure 3.1, the feature selection module of the system selects useful features and removes redundant information from the feature dataset. For this study seven feature selection algorithms are selected:

1. Step wise forward selection (SFS)
2. Fisher-index-based ranking (Fisher)
3. Mutual-information-enhancement-based ranking (MutualInfo)
4. Rough set neighborhood model 1 (RS1)
5. Rough set neighborhood model 2 (RS2)
6. Rough set neighborhood model (a fuzzy neighborhood was used) (RSF)
7. Feature discretization method (FD)

In this section, all four steps of the feature selection processes (subset generation, evaluation of generated subsets, stopping criteria, and result validation, as explained in section 2.1.2) are described for each of the methods. With respect to the evaluation criteria for the generated subsets, all methods are filter methods, i.e. use independent evaluation criteria.

3.3.1 Crisp methods

Stepwise forward selection (SFS)

Starting point and search strategy: The SFS method uses a forward search strategy. Starting with one feature that is most correlated with the target classes, SFS adds a new variable, which, together with the old one(s), most accurately predicts the target.
Evaluation criterion: At each step a linear regression model is calculated for a newly selected feature. The feature is considered to be a better choice if the linear regression model predicts the target classes giving lower regression error than the old model.
Stopping criterion: Selection stops when the significance of the new candidate model is equal or less than a specific pre-determined value. The significance is calculated by partial-f statistics (P value). In this study, $P = 0.05$, 0.01, and 0.001 were used.

Feature ranking based on the Fisher ratio

This method is based on Fisher analysis. Fisher scores are used to rank the total feature set, instead of selecting a subset. From the ranked feature list, an appropriate number of features has to be selected manually.

Feature ranking based on mutual information distribution

This method uses mutual information enhancement to rank the features. A suitable number of features has to be selected manually.

3.3.2 Soft-computing-based methods

Neighborhood-based rough set

There are three methods (RS1, RS2, RSF) that use neighborhood-based rough set models. In these methods, a neighborhood model is used to overcome the problem of continuous valued features. RS1 and RS2 use crisp neighborhood and RSF uses fuzzy neighborhood, which means that a neighboring point is considered to be a fuzzy member of the set of neighbors of the current point.
Evaluation criterion: Significance and dependency is used as evaluation criteria.
Search strategy: A forward greedy search strategy is used in these methods, and dependency is employed as the heuristic rule.
Stopping criterion: The search stops when the significance of remaining features is zero.

Features discretization

The feature selection methods feature discretization uses a global discretization technique.

3.4 Classifier

The reduced dataset of features is used to train a GMM-based classifier. For classification using this method, each class is represented by a GMM and referred to by its estimated model parameters. The class-dependent feature space is approximated by a sum of Gaussian mixtures.

3.4.1 GMM description

A Gaussian mixture model is a convex combination of M component probability densities as follows [30]

$$p\left(\vec{x}\,|\lambda\right) = \sum_{i=1}^{M} b_i p_i\left(\vec{x}\right)$$

where

$$\sum_{i=1}^{M} b_i = 1 \; for \;\; 0 \leq b_i \leq 1$$

where $\vec{x}$ is a D-dimensional feature vector and $p_i\left(\vec{x}\right)$ are component densities with weights b_i.

$$p_i\left(\vec{x}\right) = \frac{1}{\left(2\pi^{\frac{D}{2}}\right)\left|\Sigma_i\right|^{\frac{1}{2}}} exp\left[-\frac{1}{2}\left(\vec{x}-\vec{\mu_i}\right)^T \Sigma_i^{-1}\left(\vec{x}-\vec{\mu_i}\right)\right]$$

where $\vec{\mu_i}$ is the mean vector and Σ_i is the covariance matrix. Thus $\vec{\mu_i}$, Σ_i and b_i are three parameters characterizing each Gaussian mixture density. A set of these parameters is represented by λ, which represents a Gaussian mixture model. Each acoustic signal is represented by a Gaussian mixture model, that is

$$\lambda = \left\{\vec{\mu_i}, \Sigma_i, b_i\right\}, \; i = 1, ...M.$$

3.4.2 GMM model training

The GMM is used to approximate the distribution of the feature space using a set of Gaussian distributions. Given a training dataset that belongs to a specific class of acoustic signals (for example music), the goal of the class model training is to estimate the set of GMM parameters λ that best describes the distribution of the features of the training dataset. The expectation-maximization (EM) [31] algorithm is used for the training of the GMM model. Maximum likelihood estimation of parameters is iteratively obtained using the EM algorithm. Starting from an initial model λ, EM estimates a new model $\bar{\lambda}$, such that $p\left(X\,|\bar{\lambda}\right) \geq p\left(X\,|\lambda\right)$. The new model becomes the initial model for the next iteration, and a predefined convergence threshold is reached iteratively.

Chapter 4

Results and discussion

This chapter presents the results of the different feature selection methods and a comparison of these results. Important aspects for this comparison are recognition rate, complexity and consistency. Complexity and consistency can be detailed as follows.

- Complexity
  - Number of selected features
  - Number of Gaussian mixtures for model training
- Consistency
  - Performance of a feature set for all partitions of data, where that set is selected for one particular partition of data.
  - Variation of performance
    - Variation in number of selected features
    - Variation in recognition rate

4.1 Evaluation procedure

For the implementation of the system described in the previous chapter, MATLAB and RSES 2.2 (rough set exploration system 2.2) environments were used. RSES is a freely available software tool that offers means for analysis of data sets through the use of rough set theory [32]. Only the feature selection based on feature discretization was done by RSES, all other feature selection methods and other system modules were implemented in MATLAB. The feature set selected by RSES was used again in MATLAB as input to the classifier for comparison with other methods. As mentioned in section 3.1 there is no explicit rule to choose the test and training datasets. Five random partitions of the data set were made, i.e, there are five different training and testing data sets, where test and training data have the same size (50 % partitioning). For each random partition, the feature selection was applied. Using the selected features, the classifier was trained with the training data. Subsequently, the system was evaluated using the selected features and the testing data set. The recognition rate is calculated as follows:

- Individual class recognition rates were calculated and put into a confusion matrix for each partition. The confusion matrices are presented as the arithmetic mean of the confusion matrices of the five partitions, see section 4.3.1.

- The overall recognition rate for each partition is calculated as the arithmetic mean of the individual class recognition rates of that partition.
- The overall recognition rate is calculated as the arithmetic mean of all five partitions.

As explained in chapter 3, there are seven feature selection methods: stepwise forward selection (SFS), Fisher-score-based ranking (Fisher), mutual-information-based ranking (Mutual-info), Rough set neighborhood 1 (RS1), Rough set neighborhood 2 (RS1), Rough set neighborhood with fuzzy neighborhood (RSF), feature discretization using cuts (FD). In addition, various baseline feature sets are used to validate the performance of the feature selection module. These feature sets are called baseline feature sets, because there is no feature selection applied. The averages[1] of all low-level features (LLF), all long-term features (LTF) and of all MFCCs, are used as baseline feature sets.

4.2 Number of selected features

The output of a feature selection method is a subset of features. The goal is to find a set with the minimum number of features that gives the best classification rate. For the baseline feature sets, there is a fixed number of features: 16, 13, and 13 features for LLF, LTF and MFCCs respectively. The output of the feature selection methods that rank the features (Fisher and Mutualinfo) is the full set of 800 features ranked according to feature importance. For these methods, an appropriate number of features is selected manually. Figure 4.1 shows that the majority of soft computing based feature selection methods select between 16 and 23 features. For fair comparison, 25 features are selected for the ranking-based feature selection methods. So, there are also feature selection methods that do not vary in number of selected features. In figure 4.1 results are shown only for the methods that vary in number of selected features for each partition, to compare the consistency in the selected number of features for each iteration. Figure 4.1 also shows the mean and standard deviation of the number of features. It is clear from the figure that stepwise forward selection (SFS) performs worst with respect to the number of features selected as well as the deviation across all five partitions (118 features on average with a standard deviation of 6.2). RS1 outperforms all methods with respect to the mean of the selected number of features. RS1 selects 17.2 features on average, with a standard deviation of 1.3.

4.3 Recognition rate

In this section, individual class recognition rates and overall recognition rates are presented.

4.3.1 Confusion matrix

As described in section 4.1, the overall recognition rate is calculated as the average recognition rate of individual classes. The recognition rate of individual classes can be represented by a confusion matrix. A confusion matrix shows the actual classes and corresponding prediction of the classifier. The confusion matrices shown in figure 4.2 are the average of the five confusion matrices for the five partitions of data. It can be seen that the highest recognition rate is consistently observed for the audio class “speech”, whereas the audio class “speech+noise” was most difficult to predict. The audio class “noise” and “music” were most often confused with the audio class “speech+noise”, and vice versa.

[1] The average is calculated as the mean of 1-s frames (MFCCs and LLF are extracted on 20 ms subframes).

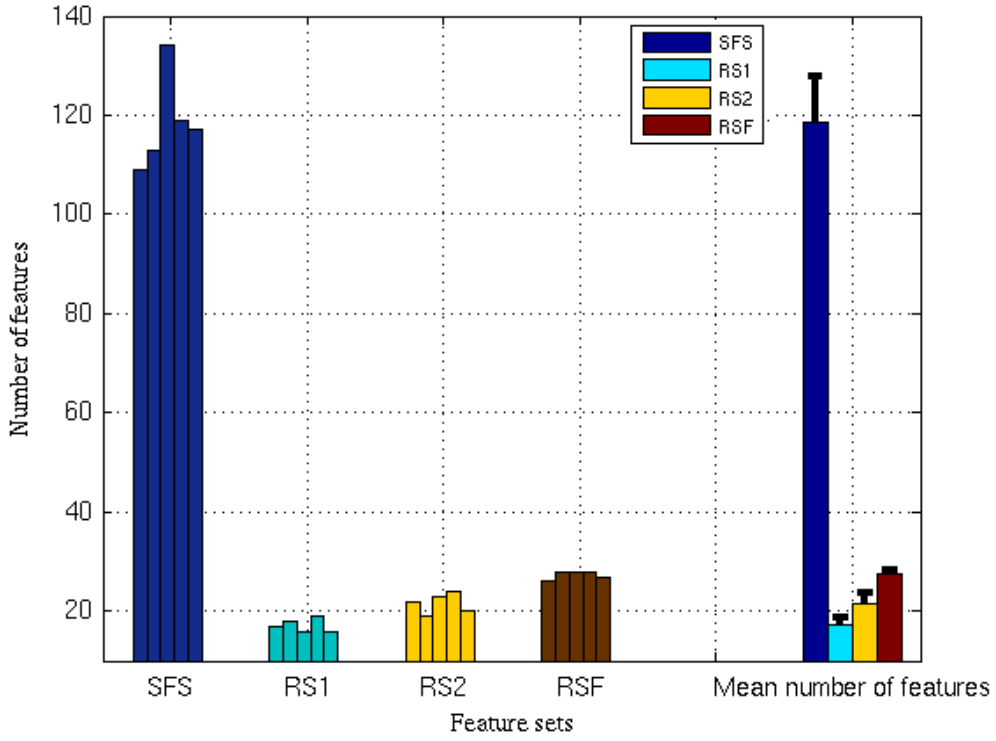

Figure 4.1: Number of features for each of the five partitions and the mean number of features.

4.3.2 Overall recognition rate for five partitions

In this section the overall recognition rates for the baseline feature sets and the feature sets selected by the feature selection methods are presented. In figure 4.3, LLF, LTF and MFCCs represent the performance for the baseline feature sets. SFS, Fisher, MutualInfo, RS1, RS2, RSF and FD show the performance for the feature selection methods. For each partition, the recognition rate is presented. In addition, the mean recognition rate of the five partitions is given; the error bars represent the standard deviation. It can be seen that SFS and RS1 outperform all other methods. These methods perform very similar (there is only a very small difference in mean and standard deviation between these methods).

4.3.3 Overlap in selected feature sets for five partitions

As described in section 4.1, five random partitions of the database were made. It is important to look whether the selected features by the feature selection methods for the five partitions have a high overlap (overlap refers to the same features present in all five feature sets). Another way to investigate this is to apply the features selected for one partitioning of data to all other four partitions. Comparing the variance of performance, it can be analyzed whether the features selected by the feature selection method are good representative features for the sound class and how close all these five feature sets are. Looking at figure 4.4, it is clear that even if the four other random partitions of data are classified, based on features selected for one particular partition: the performance does not vary much.

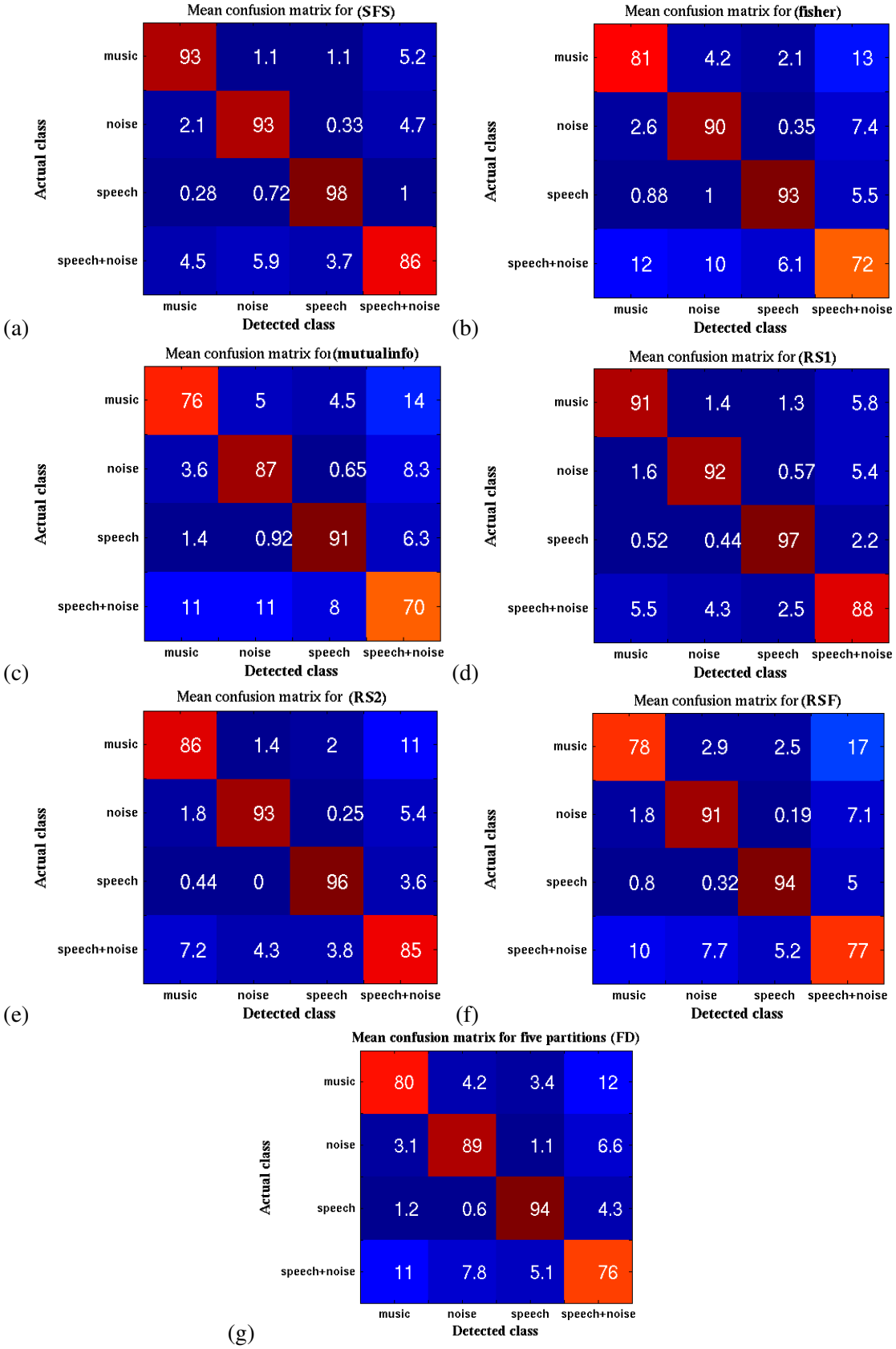

Figure 4.2: Confusion matrices for feature selection methods.

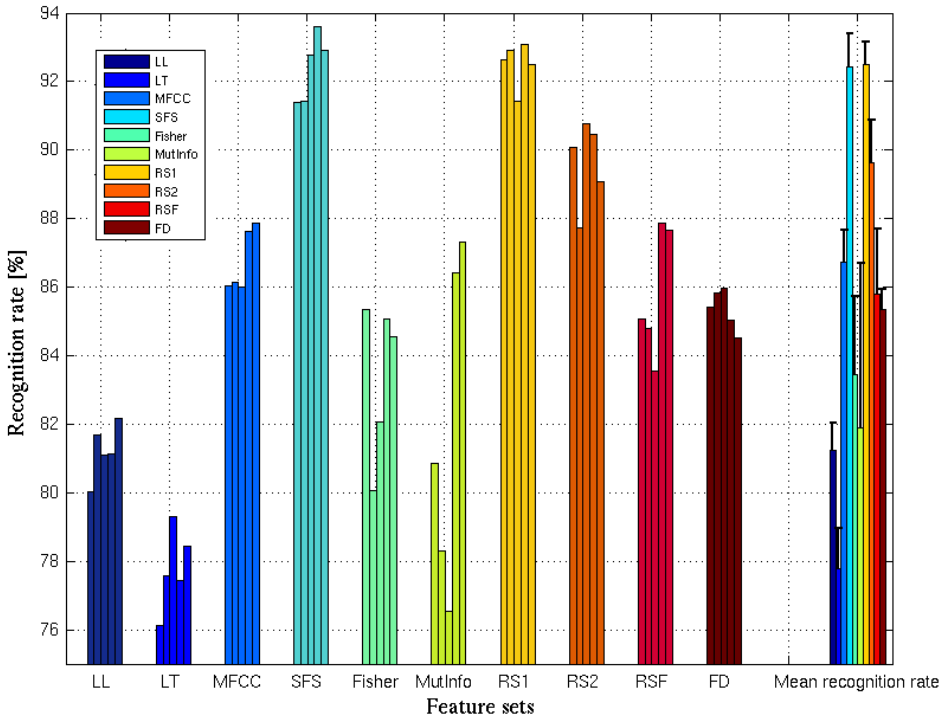

Figure 4.3: Recognition rate for five partitions, mean recognition rate and standard deviation for baseline features sets and feature selection methods (GMM20).

4.4 Performance as a function of window length

Chapter 3 describes that features are calculated on the basis of 1-s frames. To investigate the effect of the frame size, the class-dependent likelihoods are integrated across multiple frames. For this evaluation 20 Gaussian mixtures were used in the GMM classifier. Figure 4.5 shows that the system performance increases as the analysis time increases for all feature selection methods, except for SFS. SFS gives an optimal performance at a 3-s window length. An overall performance of more then 99 % is achieved with the RS1 and RS2 methods.

4.5 Performance as a function of GMM order

The number of Gaussian distributions in a mixture model is an important parameter that influences the overall complexity and performance of the system. The GMM classifier was tested with a varying number of Gaussian distributions. A 1-s window length was used for all GMM orders, and the mean recognition rate was calculated for the five partitions. Figure 4.6 shows that the best performance was achieved by using a GMM classifier with 35 components, in combination with the RS1 feature selection method, for which a 93.5 % mean recognition rate was achieved.

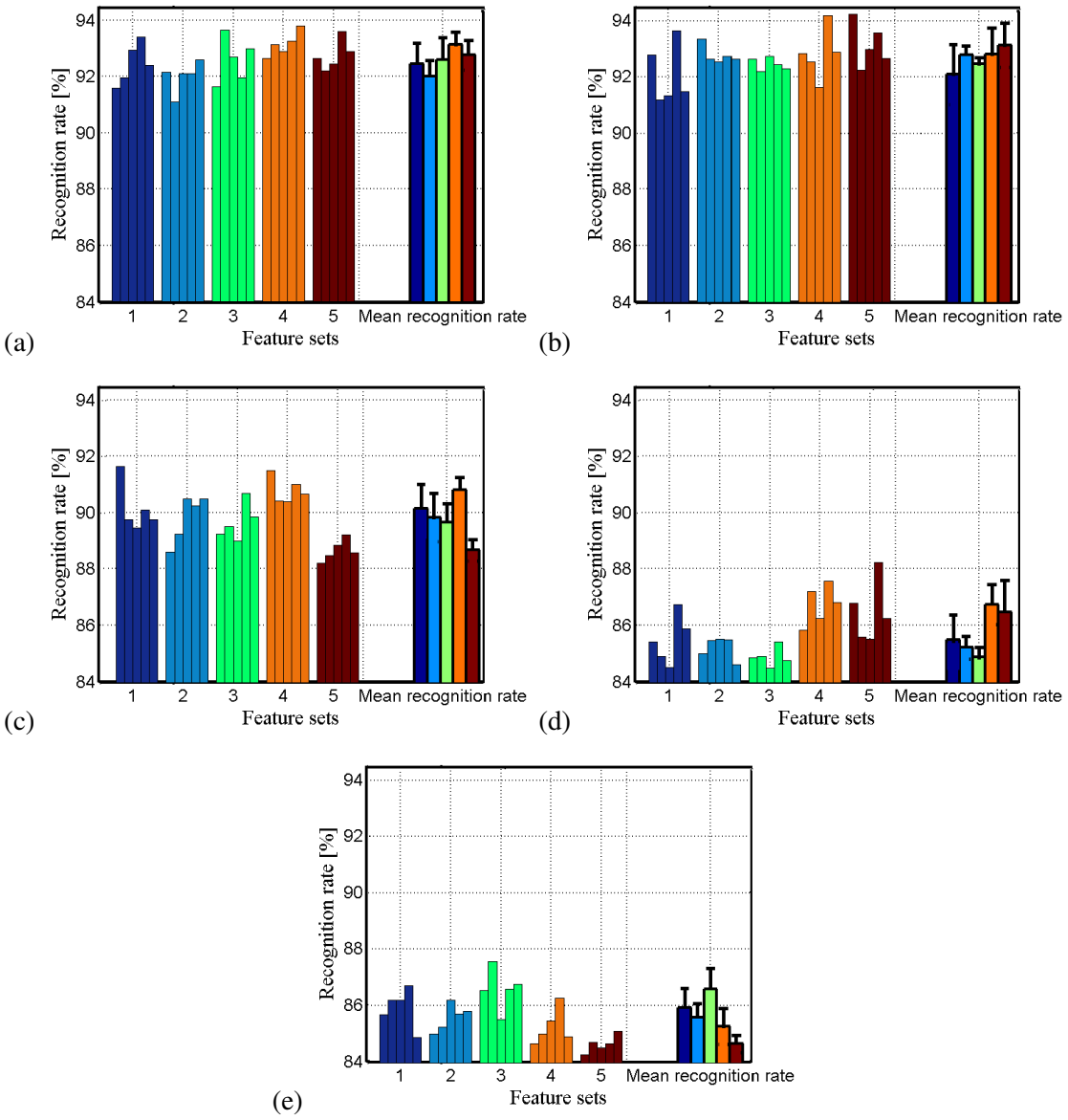

Figure 4.4: Recognition rate for all five partitions of the data based on the feature set selected for one particular partition, for (a) SFS (b) RS1 (c) RS2 (d) RSF (e) FD.

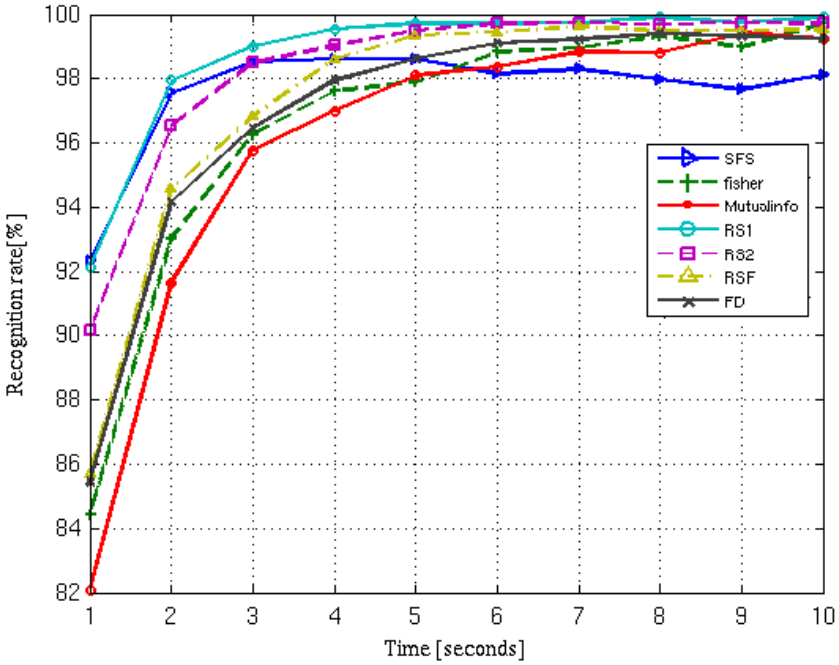

Figure 4.5: Recognition rate as a function of time length

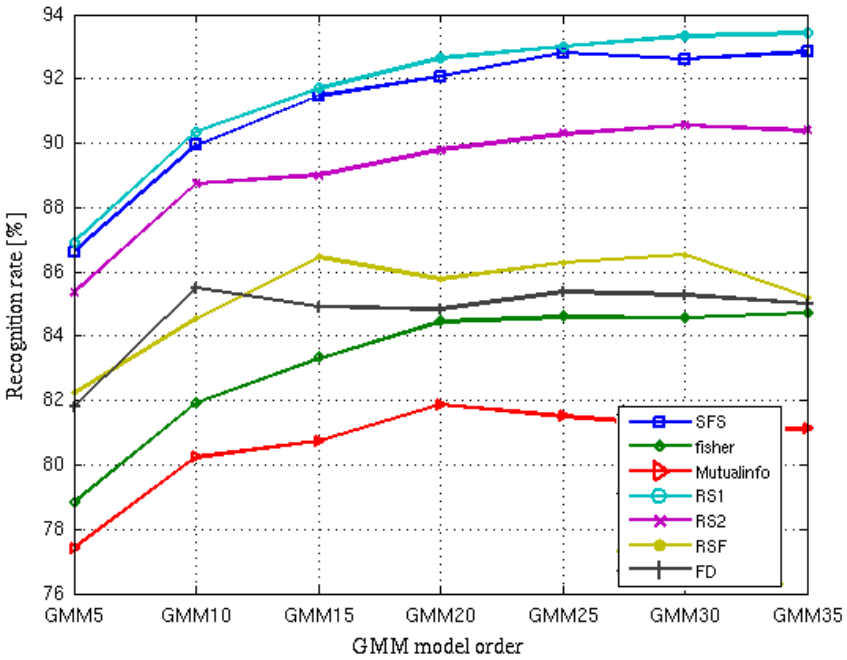

Figure 4.6: Recognition rate as a function of GMM order.

4.6 System validation

The amount of data and the content of the training and the test database have great impact on the performance of the system. For the purpose of validation of results, an additional experiment was conducted using an external database. That database was provided by M. Büchler. This validation database is not only more critical in terms of subclasses (more types of noise, different speakers in class speech, reverberated speech files with T_{60} up to 7000 ms), but also contains much more data. Table 4.1 presents a summary of the validation database. Details can be found in [22]. For this database, the system has been validated using feature selection methods SFS and RS1. Figure 4.7 shows that the average number of features selected by SFS is 157 and by RS1 is 22. Figure 4.8 shows recognition rates for 1-s frame lengths and figure 4.9 for 5-s lengths. The recognition rates of 89.5 % for 1-s frame lengths and 99.7 % for 5-s lengths with RS1 agree with the previous results, and show that the presented system is capable to achieve a robust recognition rate for reverberated speech (see sound class speech in table 4.1 for details of reverberation time).

Class	Files	Length	Description
Speech	60	1800 s	Clean speech with normal reverberation ($T_{60} \approx 500$ ms). Compressed and more reverberated speech from radio and TV. Stronger reverberated speech ($T_{60} \approx 1200\ to\ 7000$ ms).
Speech+noise	74	2220 s	Speech in social noise, speech in industrial noise, speech in car, speech in traffic noise, speech in industrial noise
Noise	80	2400 s	Social noise (cafeteria, speech babble, exhibition on hall, party, restaurant, public festival). Traffic noise (inside car, inside train, train station). Industrial noise (chain saw, construction site, drilling, grinding, jackhammer, electric saw, manual saw, steam shovel, weaving machine). Other noise (frying, office, printer, rain, shower).
Music	73	2190 s	Classical, pop, jazz, rock'n roll, folk, single instruments

Table 4.1: Sound database for system validation.

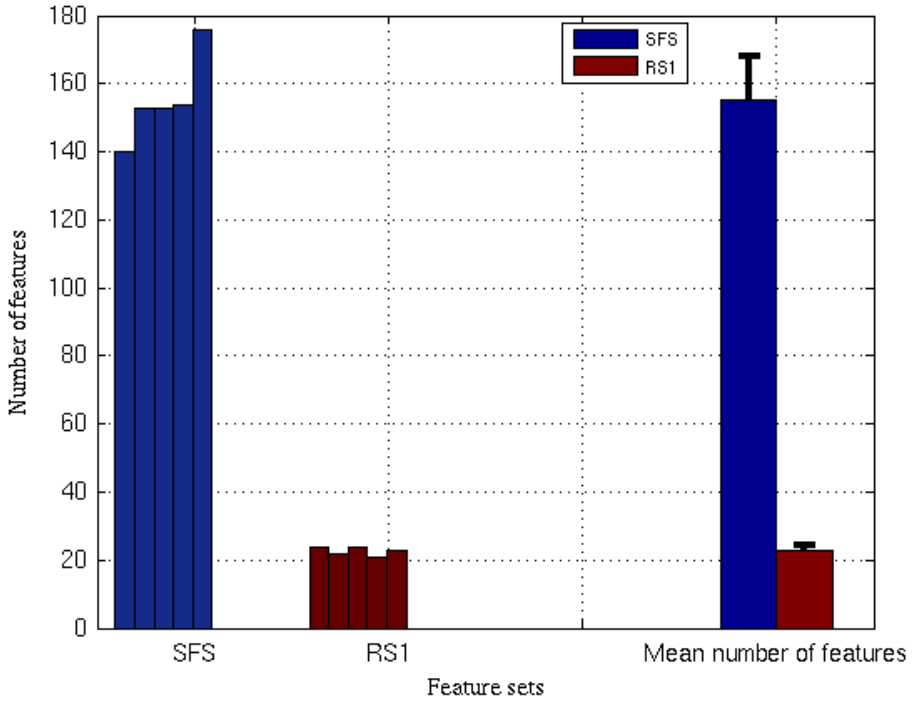

Figure 4.7: Number of features for each of five partitions and mean number of features, for RS1 and SFS methods.

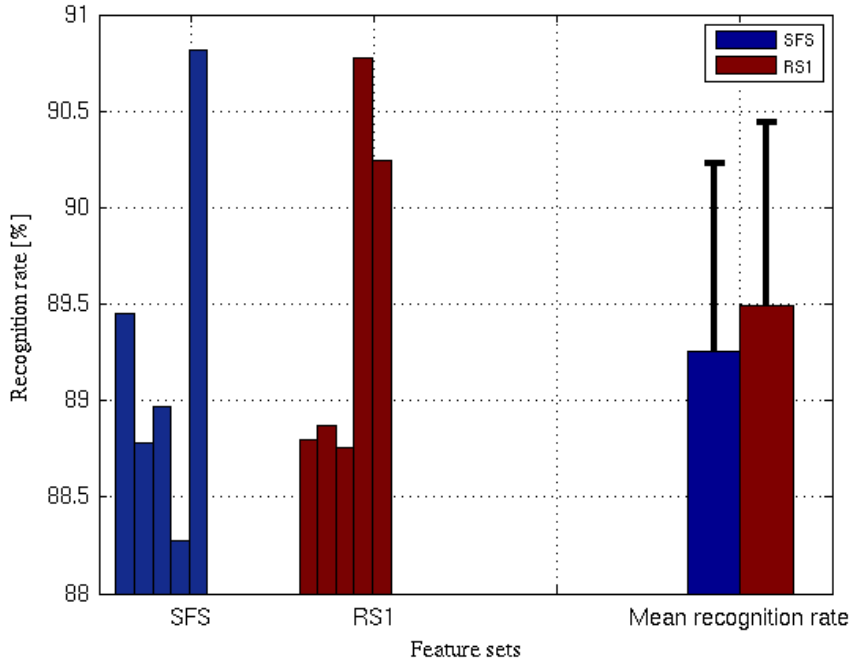

Figure 4.8: Recognition rate for five partitions, mean recognition rate and standard deviation for RS1 and SFS (1-s frame and GMM with 20 components).

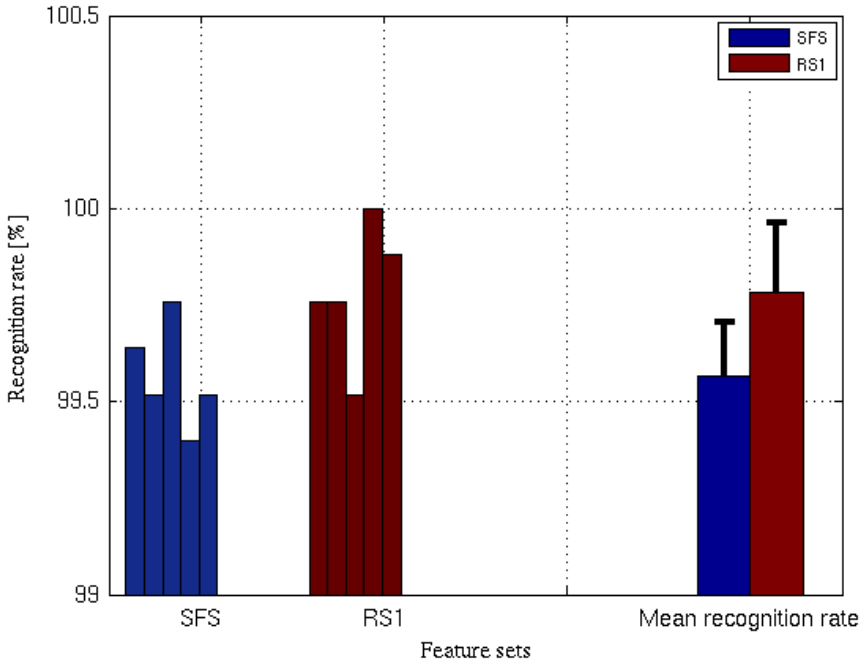

Figure 4.9: Recognition rate for five partitions, mean recognition rate and standard deviation for RS1 and SFS (5-s frame and GMM with 20 components).

4.6.1 Effect of reverberation on recognition rate

The reverberation time (RT_{60})[2] is an important descriptor of the acoustic environment. Reverberation effects the sound classes in different ways. For example, high reverberation time causes listening difficulties for hearing-impaired people [33], while music experience can be enhanced in highly reverberant environments. Figures 4.10 and 4.11 show the effect of reverberation time on recognition rate. Performance is reduced for high reverberation time. By looking at the individual class recognition rates presented in figure 4.11, one can see that the recognition rate for the class speech is reduced from 97 % to 91 % for the SFS method and from 97 % to 95 % for the RS1 method as the reverberation time increases.

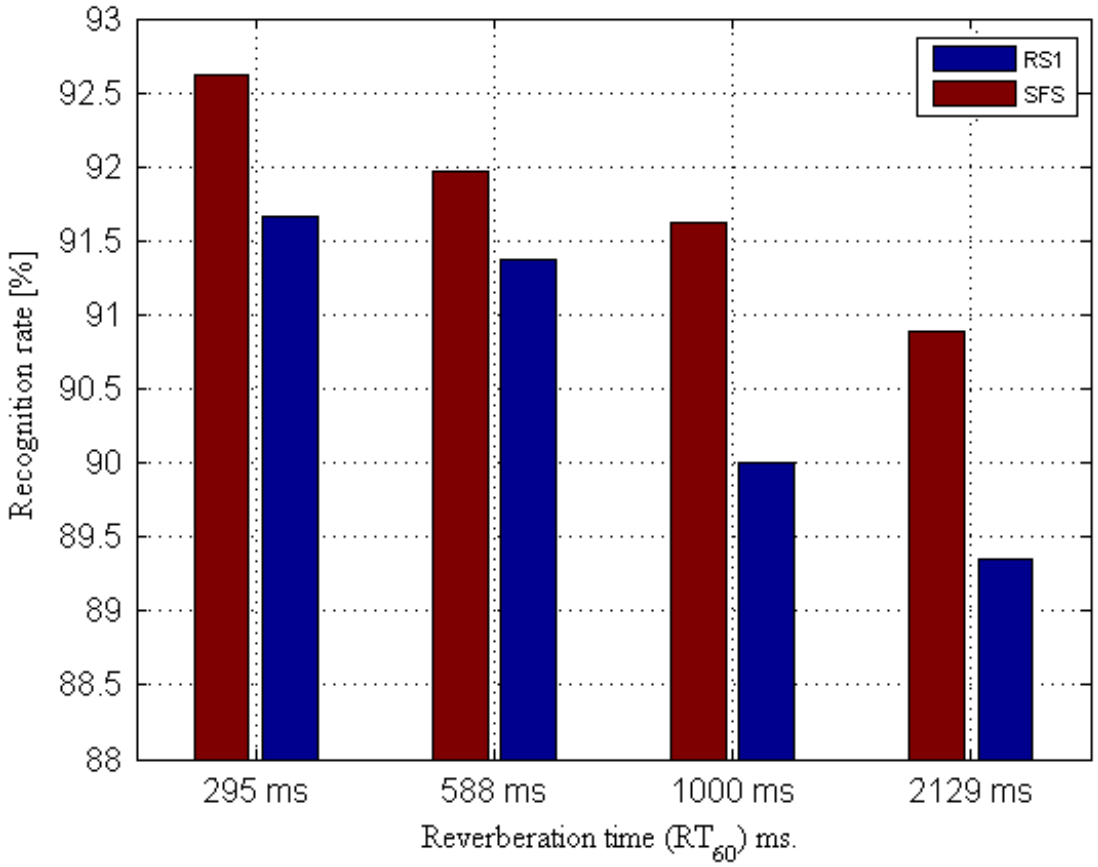

Figure 4.10: Recognition rate as a function of reverberation time.

[2]Reverberation time (RT_{60}) is defined as the time required, for the average sound in a room, to decrease by 60 decibels after a source stops generating sound.

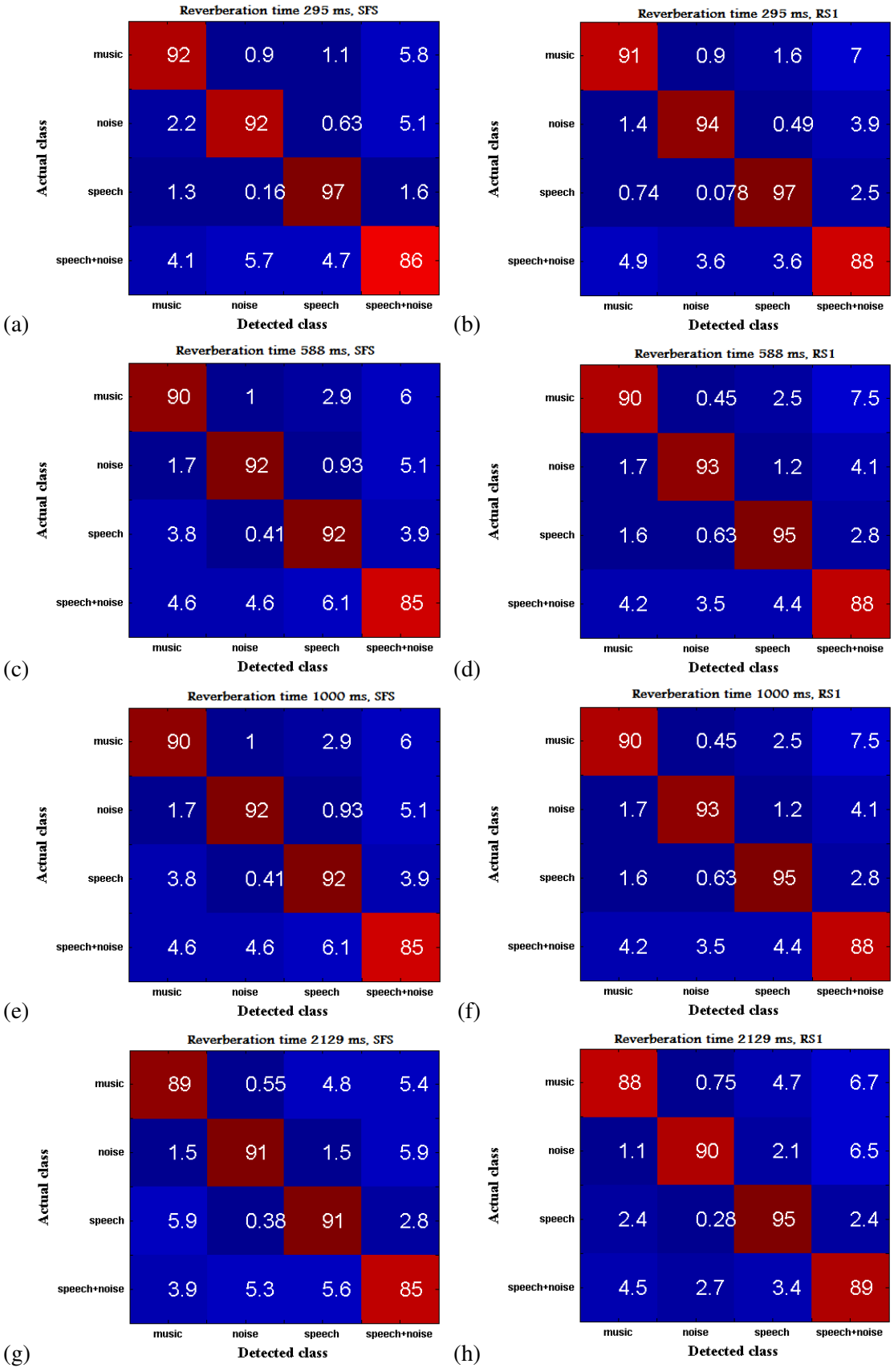

Figure 4.11: Confusion matrices for various reverberation times, for the SFS and RS1 methods.

Chapter 5

Summary and conclusions

The problem of environmental sound classification has been studied. The focus of this work was to investigate the usability of different feature selection methods for environmental sound classification, in particular soft computing based techniques.

An important aspect of the work was to design a balanced audio database, which contains sufficient audio material to properly train and evaluate the classification system.

In this study seven different feature selection methods are compared, which leads to the following conclusions:

- The selection of discriminating features clearly improves recognition performance.
- Among all evaluated methods there are two promising methods that give high correct recognition rate for the four used audio classes: rough set neighborhood model 1 (RS1) and stepwise forward selection (SFS). Experimental results show that RS1 gives 92.5 % recognition rate while selecting on average 17 features (average number of selected feature for five partitions of data) out of a total of 800 features. With the second method a comparable recognition rate of 92.4 % is achieved while selecting on average 118 features. The recognition rates based on features selected by these two methods show that it is not reasonable to use as many features as possible.
- If training and testing data are randomly selected, there is no need to run the feature selection method for all five partitions of data independently.
- Adding the class-dependent likelihoods across multiple frames is a good way to integrate information over a longer observation time. Accumulating likelihood probabilities of five frames, a recognition rate of 99.7 % is achieved with the RS1 method and 98.4 % with the SFS method.
- The confusion matrices show that the audio class “speech+noise” is the most difficult class to classify and the class “speech” is the easiest to classify.
- Validation results show that the proposed system gives robust performance (with a slight (1.5 %) reduction in performance for high reverberation time) for reverberated speech material.
- Looking at the feature sets selected for all five partitions of data, it is found that SNR, Tonality and MFCCs are good discriminating features.

Coming back to the use of soft computing based techniques for feature selection, it can be concluded that these are indeed very useful, as has been shown by the outstanding results of RS1. It would be interesting to investigate the performance of the soft computing based feature selection methods with simple rule based classifiers, and to investigate the performance of fuzzy inference classifiers, i.e., classifiers that are also based on soft computing strategies.

References

[1] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. San Diego, California: Academic Press, 1990. 1

[2] V. Peltonen, J. Tuomi, A. Klapuri, and Jyri, "Computational auditory scene recognition," *IEEE conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1941–1944, 2002. 1

[3] R. E. Bellman, *Adaptive control processes - A guided tour*. Princeton, New Jersey, U.S.A.: Princeton University Press, 1961. 2

[4] L. Hanna, "Feature selection vs. feature space transformation in automatic music genre classification tasks," *AES Convention*, May 2009. 2

[5] B. Kostek, P. Szczuko, P. Zwan, and P. Dalka, "Processing of musical data employing rough sets and artificial neural networks," *Transactions on Rough Sets III*, vol. 34, pp. 112–133, 2005. 3

[6] L. Lazareck and S. Ramanna, "Classification of swallowing sound signals: A rough set approach," in *Rough Sets and Current Trends in Computing*, 2004, pp. 679–684. 3

[7] B. Kostek, *Soft Computing in Acoustics: Applications of Neural Networks, Fuzzy Logic and Rough Sets to Musical Acoustics*. Physica-Verlag HD, 1999. 3

[8] R. Jensen and Q. Shen, *Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches*. Wiley-IEEE Press, 2008. 4

[9] R. Kohavi and G. H. John, "Wrappers for feature subset selection. artificial intelligence," *Artificial intelligence*, vol. 97, pp. 273 – 324, 1997. 5

[10] A. L. Blum and R. L. Rivest, "Training a 3-node neural network is np-complete," *Neural Networks*, vol. 5, pp. 117 – 127, 1992. 5

[11] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, pp. 131–156, 1997. 5

[12] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan, "Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies," *VLDB J.*, vol. 7, no. 3, pp. 163–178, 1998. 6

[13] M. Zaffalon and M. Hutter, "Robust feature selection by mutual information distributions," in *Proceedings of the 18th International Conference on Uncertainty in Artificial Intelligence (UAI-2002*. Morgan Kaufmann, 2002, pp. 577–584. 7

[14] H. Stoppiglia, G. Dreyfus, R. Dubois, and Y. Oussar, "Overfitting in making comparisons between variable selection methods," *Journal of Machine Learning Research*, vol. 3, pp. 1371–1382, 2003. 7

[15] J. Komorowski, L. Polkowski, and A. Skowron, "Rough sets: A tutorial," 1998. 8, 9

[16] D. Dubois and H. Prade, "Possibility theory, probability theory and multiple-valued logics: A clarification," *Annals of Mathematics and Artificial Intelligence*, vol. 32, pp. 35–66, 2001. 11

[17] J. Ramík and M. Vlach, *A Non-controversial Definition of Fuzzy Sets*, ser. Lecture Notes in Computer Science. Springer, 2004, vol. 3135. 11

[18] Y. Yong, W. Guoyin, and H. Kong, "Self-learning facial emotional feature selection based on rough set theory," *Mathematical Problems in Engineering*, 2009. 11

[19] Q. Hu, D. Yu, and Z. Xie, "Neighborhood classifiers," *Expert Syst. Appl.*, vol. 34, no. 2, pp. 866–876, 2008. 11

[20] D. R. Wilson and T. R. Martinez, "Improved heterogeneous distance functions," *CoRR*, vol. cs.AI/9701101, 1997. 12

[21] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proceedings of the Twelfth International Conference on Machine Learning*, A. Prieditis and S. Russell, Eds., 1995, pp. 194–202. [Online]. Available: http://citeseer.ist.psu.edu/109288.html 12

[22] M. Büchler, "Algorithms for sound classification in hearing instruments," Ph.D. dissertation, Institute for biomedical Engineering, University and ETH Zürich, 2002. 13, 16, 26

[23] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound." *Proceedings of the Institute of Phonetic Sciences, Amsterdam*, vol. 17, pp. 97–110, 1993. [Online]. Available: http://www.fon.hum.uva.nl/Proceedings/Proceedings_17_paper_abs.html 16

[24] T. Andersson, "Audio classification and content description," Master's thesis, Lulea University of Technology, 2004. 16

[25] K. Jensen and T. H. Andersen, "Real-time beat estimation using feature extraction," 2003. 16

[26] P. Jouni, "Methods of automatic audio content classification," Master's thesis, Helsinki university of technology, 2007. 16

[27] [Online]. Available: http://cobweb.ecn.purdue.edu/~malcolm/interval/1998-010/ 15

[28] [Online]. Available: http://htk.eng.cam.ac.uk/ 15

[29] L. Lu and H. J. Zhang, "Content analysis for audio classification and segmentation," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 7, pp. 504–516, 2002. 16

[30] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995. [Online]. Available: http://dx.doi.org/10.1109/89.365379 18

[31] B. Jeff, "A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden Markov models," Tech. Rep., 1998. 18

[32] "RSES 2.2 User's Guide," Warsaw University, Tech. Rep., 2005. [Online]. Available: http://logic.mimuw.edu.pl/~rses 19

[33] K. Shiraishi, M. Inoue, K. Yonemoto, and A. Imamura, "Evaluation method for hearing aid fitting under reverberation: comparison between monaural and binaural hearing aids," *J Physiol Anthropol Appl Human Sci.*, vol. 7, no. 3, pp. 163–178, 2004 Nov. 29

Appendix A

Sound database

- The sound class *music* contains files which represent pop, rock, techno, classical, country, funk and electro music genres.
- In class *speech* there are male and female speakers speaking in Dutch and English. Almost each file contains a different speaker.
- The *noise* class of the sound database contains real recordings of different noise types, of which there is babble noise, cafeteria noise, inside-bus, inside-car and inside-train noise, train station, busy street, shopping center, kindergarten, factory, supermarket, machine noise, traffic noise; besides these real recordings there is also white and pink noise which are computer generated.
- The sound class *speech+noise* contains real recordings in different noisy environments (outside environment with wind noise and cocktail party environment). In this sound class there are also computer simulated files which contain almost all of the noise categories described in class "noise", mixed with different male and female speakers at different signal-to-noise ratios.

Class	File number	Description
29 Noise files	1	Babble noise
	2	Machine noise
	3	Inside bus noise
	4	Airplane noise
	5	Less busy road traffic noise
	6	Kitchen noise
	7	Helicopter noise
	8	Printer noise
	9	Street noise
	10	Foot steps noise (Walking/running)
	11	Inside car noise (Without speech)
	12	FM channel noise
	13	Inside car noise (passengers speaking)
	14	Inside train noise

Class	File number	Description
	15	kindergarten noise
	16	Inside train (without speech) noise
	17	Factory noise
	18	Machine-gun noise
	19	Water-fall noise
	20	Street noise
	21	TV noise
	22	Pub noise
	23	Basketball match noise
	24	Inside train (people speaking) noise
	25	Inside bus noise
	26	Train station noise
	27	Car noise
	28	TV noise
	29	Babble noise
38 Music files	1	Country
	2	Country
	3	Pop
	4	Pop
	5	Electro
	6	Electro
	7	Electro
	8	Rock
	9	Rock
	10	Pop
	11	Pop
	12	Electro
	13	Electro
	14	Folk
	15	Gospel
	16	Gospel
	17	Rock
	18	Rock
	19	Rock/pop
	20	Rock/pop
	21	Disco/Pop
	22	Disco/Pop
	23	Folk
	24	Folk
	25	Classic
	26	Classic
	27	Techno
	28	Techno
	29	Rock
	30	Rock
	31	Funk
	32	Funk

Class	File number	Description
	33	Rock
	34	Rock
	35	Country
	36	Country
	37	Trance
	38	Pop
24 Speech files	1	Female Dutch
	2	Female Dutch
	3	Female Dutch
	4	Male Dutch
	5	Male Dutch
	6	Make Dutch
	7	Male & female Dutch
	8	Make & female Dutch
	9	Male & female Dutch
	10	Male Dutch
	11	Female English
	12	Female English
	13	Male English
	14	Male English
	15	Male English
	16	Male German
	17	Female German
	18	Male German
	19	Female English
	20	Male English
	21	Male English
	22	Female English
	23	Male English
	24	Male English
27 Speech + noise files	1	Speech + Cocktail Party noise
	2	Speech + Machine noise SNR 3 dB
	3	Speech + Inside bus noise SNR 4 dB
	4	Speech + Airplane noise SNR 3 dB
	5	Speech + Less busy road traffic noise SNR 0 dB
	6	Speech + Kitchen noise SNR 4 dB
	7	Speech + Helicopter noise SNR 5 dB
	8	Speech + Printer noise SNR 2 dB
	9	Speech + Street noise SNR 4 dB
	10	Speech + Foot steps noise SNR 2 (dB
	11	Speech + Inside car noise SNR 4 dB
	12	Speech + FM channel noise SNR 1 dB
	13	Speech + Inside car noise SNR 2 dB
	14	Speech + Inside train noise SNR 0 dB
	15	Speech + kindergarten noise SNR 0 dB

Class	File number	Description
	16	Speech + Inside train noise SNR 4 dB
	17	Speech + Factory noise SNR 0 dB
	18	Speech + Machine-gun noise SNR 2 dB
	19	Speech + Water-fall noise SNR 2 dB
	20	Speech + Street noise SNR 1 dB
	21	Speech + TV noise SNR 4 dB
	22	Speech + Pub noise SNR 5 dB
	23	Speech + Wind noise
	24	Speech + Traffic noise
	25	Speech + Foot steps noise
	26	Speech + (Laugh + wind)
	27	Speech + Wind noise

Appendix B

List of features

Index	Features
1-9	Statistical features[1] of zero-crossing rate
10-18	Statistical features of crest factor
19-27	Statistical features of periodicity
28-36	Statistical features of RMS of frame energy
37-45	Statistical features of RMS of frame energy in decibel
46-54	Statistical features of mean level fluctuation strength
55-63	Statistical features of spectrum centroid
64-72	Statistical features of spectrum spread
73-81	Statistical features of spectrum flatness
82-90	Statistical features of spectrum flux
91-99	Statistical features of spectrum irregularity
100-108	Statistical features of high frequency content
109-117	Statistical features of spectrum roll-off
118-126	Statistical features of spectrum bandwidth
127-135	Statistical features of spectrum band-energy ratio
136-144	Statistical features of spectrum entropy
145-153	Statistical features of delta zero-crossing rate
154-162	Statistical features of delta crest factor
163-171	Statistical features of delta periodicity
172-180	Statistical features of delta RMS of frame energy
181-189	Statistical features of delta RMS of frame energy in decibel
190-198	Statistical features of delta mean level fluctuation strength
199-207	Statistical features of delta spectrum centroid
208-216	Statistical features of delta spectrum spread
217-225	Statistical features of delta spectrum flatness
226-234	Statistical features of delta spectrum flux
235-243	Statistical features of delta spectrum irregularity
244-252	Statistical features of delta high frequency content

[1] Statistical features include, mean, average deviation, standard deviation, variance, skewness, kurtosis, 10th percentile, 50th percentile, 90th percentile, and calculated respectively.

Index	Features
253-261	Statistical features of delta spectrum roll-off
262-270	Statistical features of delta spectrum bandwidth
271-279	Statistical features of delta spectrum band-energy ratio
280-288	Statistical features of delta spectrum entropy
289-297	Statistical features of double-delta zero-crossing rate
298-306	Statistical features of double-delta crest factor
307-315	Statistical features of double-delta periodicity
316-324	Statistical features of double-delta RMS of frame energy
325-333	Statistical features of double-delta RMS of frame energy in decibel
334-342	Statistical features of double-delta mean level fluctuation strength
325-333	Statistical features of double-delta RMS of frame energy in decibel
334-342	Statistical features of double-delta mean level fluctuation strength
343-351	Statistical features of double-delta spectrum centroid
352-360	Statistical features of double-delta spectrum spread
361-369	Statistical features of double-delta spectrum flatness
370-378	Statistical features of double-delta spectrum flux
379-387	Statistical features of double-delta spectrum irregularity
388-396	Statistical features of double-delta high frequency content
397-405	Statistical features of double-delta spectrum roll-off
406-414	Statistical features of double-delta spectrum bandwidth
415-423	Statistical features of double-delta spectrum band-energy ratio
424-432	Statistical features of double-delta spectrum entropy
433	Histogram width
434	Histogram symmetry
435	Histogram skewness
436	Histogram kurtosis
437	Histogram lower half
438	Tonality
439	Pitch variance
440	Delta consecutive pitches
441	Periodicity ratio
442	Noisy frames ratio
443	High zero-crossing ratio
444	Low short-time energy ratio
445	Fluctuation of spectral entropy
446-454	Statistical features of MFCC1
455-463	Statistical features of MFCC2
464-472	Statistical features of MFCC3
473-481	Statistical features of MFCC4
482-490	Statistical features of MFCC5
491-499	Statistical features of MFCC6
500-508	Statistical features of MFCC7
509-517	Statistical features of MFCC8
518-526	Statistical features of MFCC9
527-535	Statistical features of MFCC10
536-544	Statistical features of MFCC11

Index	Features
545-553	Statistical features of MFCC12
554-562	Statistical features of MFCC13
563-571	Statistical features of delta MFCC1
572-580	Statistical features of delta MFCC2
581-589	Statistical features of delta MFCC3
590-598	Statistical features of delta MFCC4
599-607	Statistical features of delta MFCC5
608-616	Statistical features of delta MFCC6
617-625	Statistical features of delta MFCC7
626-634	Statistical features of delta MFCC8
635-643	Statistical features of delta MFCC9
644-652	Statistical features of delta MFCC10
653-661	Statistical features of delta MFCC11
662-670	Statistical features of delta MFCC12
671-679	Statistical features of delta MFCC13
680-688	Statistical features of double-delta MFCC1
689-697	Statistical features of double-delta MFCC2
698-706	Statistical features of double-delta MFCC3
707-715	Statistical features of double-delta MFCC4
716-724	Statistical features of double-delta MFCC5
725-733	Statistical features of double-delta MFCC6
734-742	Statistical features of double-delta MFCC7
743-751	Statistical features of double-delta MFCC8
752-760	Statistical features of double-delta MFCC9
761-769	Statistical features of double-delta MFCC10
770-778	Statistical features of double-delta MFCC11
779-787	Statistical features of double-delta MFCC12
788-796	Statistical features of double-delta MFCC13
797	Signal to noise ratio
798	Onset1
799	Onset2
800	Onset3

