

Project 3.1: Who is talking?

Group 2

Andreas Bierfert, Mark Boeren,
Pieter Goddijn, Kay de Vries

Block 3.3
Maastricht, January 24th, 2007

University of Maastricht
Maastricht ICT Competence Centre
Project 3.1: Who is talking?

Group 2
Andreas Bierfert, Mark Boeren,
Pieter Goddijn, Kay de Vries

Block 3.3
Maastricht, January 24th, 2007
Jan Paredis

Contents

1	Introduction	6
1.1	Background/History	6
1.2	Relevancy	6
1.3	Report Structure	7
2	Theoretical Background	8
2.1	Feature Sets	8
2.1.1	Mel Frequency Cepstral Coefficients	8
2.1.2	Relative Spectral Feature Set	9
2.1.3	9 Band Filter Cross Correlation	9
2.2	Classification	10
2.2.1	K-nearest neighbour	11
2.2.2	Kohonen's Self organizing feature maps	11
2.2.3	K-means	12
2.2.4	Evaluation	13
3	Testing	15
3.1	Recording of Samples	15
3.2	Setup of the tests	15
3.3	1. Test: Calibrating Parameters	15
3.4	2. Test: Single Speaker Recognition	16
3.5	3. Test: Multi Speaker Recognition	16
4	Results	17
4.1	Test 1: Calibrating Parameters	17
4.2	Dimensionality	18
4.3	Test 2: Single Speaker Recognition	18
4.4	Test 3: Multi Speaker Recognition	20
5	Conclusion	23
5.1	Summary	23
5.2	Discussion	23
A	Additional Graphs	25
A.1	Project Flow	25
A.2	Results	25
B	Additional Material	28
B.1	Sample Sentences	28

List of Figures

2.1	MFCC feature set of a voice file	9
2.2	RASTA feature set of a voice file	10
2.3	9BAND feature set of a voice file	11
2.4	Two kohonen grid forms	12
2.5	Visualization of the evaluation process	14
4.1	Influence of SOM settings	17
4.2	Influence of number of dimensions	18
4.3	Comparison between feature sets and classifiers	19
4.4	Average classification voice	19
4.5	Average classification noise	20
4.6	False positive voice	21
4.7	False positive noise	21
4.8	Running Time	22
4.9	Evaluation multi-speaker	22
A.1	Flow of the project during speaker recognition	26
A.2	Flow of the project during the initialization	27

Abstract

As the need for security systems increases, voice recognition is an important field of research. This report describes a comparative study into the performance of several voice recognition techniques. First several means to extract relevant information from the data are discussed. These feature sets used are the Mel Frequency Cepstral Coefficient (MFCC), the Relative spectral (RASTA), the 9-Band Filter Cross Correlation (9BAND) and the naive fast Fourier Transform feature sets. The feature sets are then classified on a frame per frame basis by means of K-nearest neighbour and a Kohonen Self Organizing Map(SOM). These tests are run in a 10-fold cross validation manner to get a measure of performance on unseen data. After the frame by frame classification research is done into determining the presence of speakers in multi speaker samples. Testing this, the Relative spectral feature set was performing rather poor for speaker recognition, while the Mel Frequency Cepstral Coefficient feature set produced the best results. The difference between Kohonen and nearest neighbour proved small in terms of result but huge in terms of classification speed. Further research is needed, especially in the multi speaker section.

Chapter 1

Introduction

1.1 Background/History

In the last few years, the field of speech recognition is rapidly maturing and has become a major area of research in the artificial intelligence community. However one is mistaken if he thinks his field has emerged in the last two decades. When Alexander Graham Bell invented his telephone, he was inventing a system to translate speech into text for his deaf wife. Off course this was only the beginning!

Since the 1960s computer scientists have been researching ways and means to make computers able to record interpret and understand human speech. Throughout the decades this has been a daunting task. Even the most rudimentary problem such as digitalizing (sampling) voice was a huge challenge in the early years. It took until the 1980s before the first systems arrived which could actually decipher speech. Off course these early systems were very limited in scope and power.

Shortly before Second World War a related field emerged, the field of speaker recognition. During the war Bell laboratories did extensive research on identifying (German) speech from radio waves by means of analyzing voice spectrograms. This research wasn't perfected before the war ended and with the war over research slowed down dramatically. It wasn't until the late 1950s before this research regained momentum. The cause for this was a large number of bomb threats to airplanes (by telephone). Research done by Lawrence G. Kersta (Bell labs) led to a device supposedly able to identify speakers with 99.4% accuracy. For a short term this was even used in US courts of justice, but soon after deemed inadmissible again. While this was done without the use of computers as is the case these days, the direction of the field was set, security.

1.2 Relevancy

In this day and age, Security has been playing an ever increasing role in our society. Not only has the threat of terrorism increased again to levels comparable to the late seventies and early eighties (RAF, ETA, IRA, among others). But also the means amounts of (digital) communication and information exchange has

increased enormously. This requires protection (against access by unauthorized parties) and detection (law enforcement/ national security). One could say the demand for digital speaker recognition has never been greater. Of course the use of these systems has many ethical consequences and if these systems are used accuracy is of utmost importance.

Therefore it was decided to do research comparing the accuracy of different speaker recognition algorithms both for single speaker recordings which could be used in for example a voice activated lock. And multi speaker conversations which could be used for identifying speakers on a phone tab.

1.3 Report Structure

In the next chapter the algorithms and concepts that were used are introduced and explained. Chapter 3 will explain the test setup and chapter 4 will give a detailed analysis of the results. Chapter 5 will end this report with a discussion of the work done and ideas for futur works.

Chapter 2

Theoretical Background

In order to do automatic speaker verification certain steps have to be taken in order for the system to perform well. The problem of false verification of an invalid speaker is as problematic as the problem of not verifying a valid speaker. To make the verification possible and accurate certain steps and transformations of the waveform data have to be taken.

The process of speaker recognition is as follows. The first step is to take the recorded speaker samples and extracting feature information, the so called feature sets, from them. Then these feature sets can be used together with the classifiers to verify to which speaker an unknown input belongs to. In the end a final algorithm is applied to make a more global decision about the verification as the classifiers only look at a single frame for their decision.

2.1 Feature Sets

For the comparative study of different voice recognition techniques several means of generating feature sets are used. Feature extraction estimates variables from the speaker into a feature vector. The goal is to find a relatively low-dimensional feature set that preserves useful information.

As a basic feature set the naive Fourier Transform (FFT), reduced by a Discrete Cosine Transform (DCT), is implemented. While this feature set is expected to perform rather poorly it is used as a base line to judge the performance of the other feature sets. The other three feature sets that are compared with FFT will be discussed in more detail in the next paragraphs.

2.1.1 Mel Frequency Cepstral Coefficients

The Mel Frequency Cepstral Coefficients (MFCC) [1],[4] feature set is based on the human perception of sound. Humans perceive sound on a logarithmic scale also known as Mel scale. Applying filters from the Mel domain to the signal enhances the speech related part of the spectrum and reduces the rest. As a result the generated features are more likely to yield information about the actual speech.

The algorithm works as follows. The input signal is split up into overlapping windows to avoid sharp edges at the cut points of the windows. To avoid

emphasising the overlapping regions the edges are smoothed by a hamming window function [12]. Then a Fourier Transform is applied to these windows. Afterwards 30 filters from the Mel domain are applied to the windows and the logarithmic sum of all values for each filter gives one distinct feature. Then a Discrete Cosine Transform is applied to reduce the feature vector dimensionality (see picture 2.1).

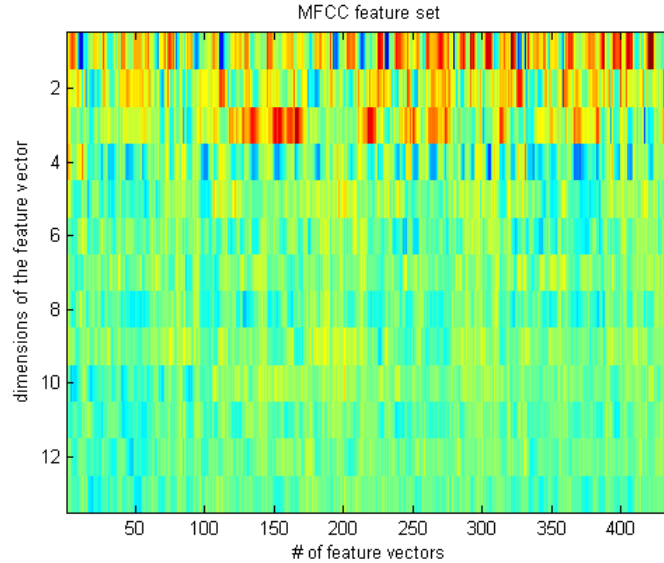


Figure 2.1: MFCC feature set of a voice file

2.1.2 Relative Spectral Feature Set

The Relative Spectral (RASTA) [6] feature set is based on the Mel filters already known from the MFCC feature set. In addition to the idea of human perception of sound the RASTA feature set tries to suppress slow changing components of the human voice.

The RASTA feature set generation works the same as the first steps of MFCC. The input signal is split into overlapping windows with utilization of a hamming window function. Then the Fourier Transform is applied and the resulting data is filtered with 30 Mel filters. A special RASTA filter function (see equation below) is applied to the signal and finally a Discrete Cosine Transform is used to reduce the dimensionality of the dataset (see picture 2.2).

$$H(z) = 0.1z^4 \cdot \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{1 - 0.98z^{-1}} \quad (2.1)$$

2.1.3 9 Band Filter Cross Correlation

The 9 Band Filter Cross Correlation (9BAND) [7] feature set as the name implies utilizes the principal that certain frequency ranges of speech correlate with each

other. This is an essential difference between a speech signal and a noise signal. 9BAND utilizes this to discriminate between voice and noise.

Transformation of data into this feature space is done by performing a number of steps. First the input signal is divided into smaller windows. Then a Fourier Transform is applied to those windows. In the frequency domain nine filters from the Mel domain are applied in the range from 50-6500 Hz. This frequency range has proven to include most of the human voice. After the filters are applied each filters data is transformed with an inverse Fourier Transform. Finally the correlation matrix is calculated between all filters and the square rooted sum is returned as a feature vector (see picture 2.3).

2.2 Classification

In the previous chapter the generation of the feature sets was detailed. Because these feature vectors compromise a labelled data set with each vector labelled, it's a natural choice to classify on a feature per feature basis. Each testing sample will be divided into frames similar to the known data and feature vectors will be calculated using the same feature set generator as used on the know data. These test vectors are tested against the known data. For this classification we applied a K-nearest neighbour and a Kohonen self organizing map. Finally a K-means clustering algorithm is used for data reduction. These algorithms are discussed in greater detail in the following paragraphs. Evaluation of complete test samples is discussed in the final paragraphs.

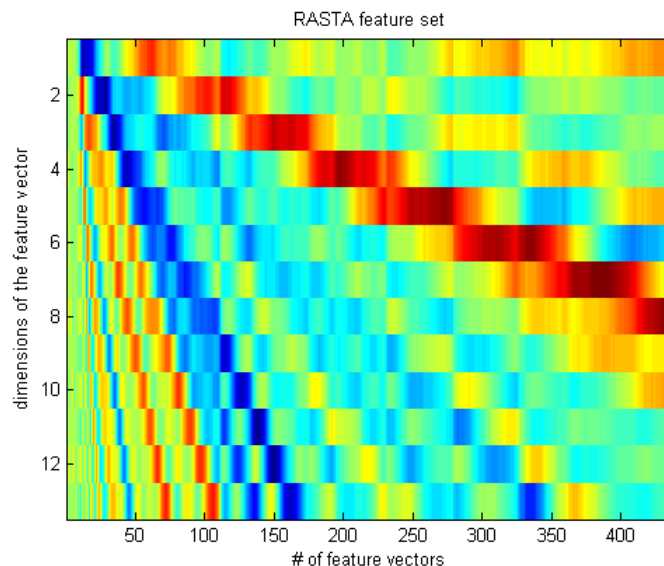


Figure 2.2: RASTA feature set of a voice file

2.2.1 K-nearest neighbour

The first algorithm used is the K-nearest neighbour classifier (KNN). This algorithm calculates the Euclidean distance to the known data points and returns the K closest data points. Despite its simplicity this algorithm is expected to yield good results. The downside of this algorithm is that is computationally intensive and the computation is done at classification time. Another problem with this algorithm is that noisy trainings data is likely to affect the classification result. The inductive bias of this algorithm is that points close to each other in feature space are likely of similar origin.

2.2.2 Kohonen's Self organizing feature maps

The self organizing feature map (SOM) evolved from early research in neural networks, specifically adaptive learning and associative memory. The original aim was to develop an auto associative neural network structure which would explain the spatial organization of certain brain areas, especially the cerebral cortex where the visual and auditory functions of the brain reside [11]. This area of research started with the spatially ordered line detectors of von der Malsburg (1973) and the neural field model of Amari in 1980. These first forays had rather weak results. The Breakthrough in this area was made by Kohonen [9] who created the first well performing auto associative network by combining competitive neural network with a winner-takes-all strategy, with a system to controls the synaptic plasticity of the neurons in the aforementioned network (the neighbourhood function).

The SOM is defined as an ordered mapping of a set of input vectors to a regular grid, usually 2 dimensional. Each input is mapped to the most similar (near-

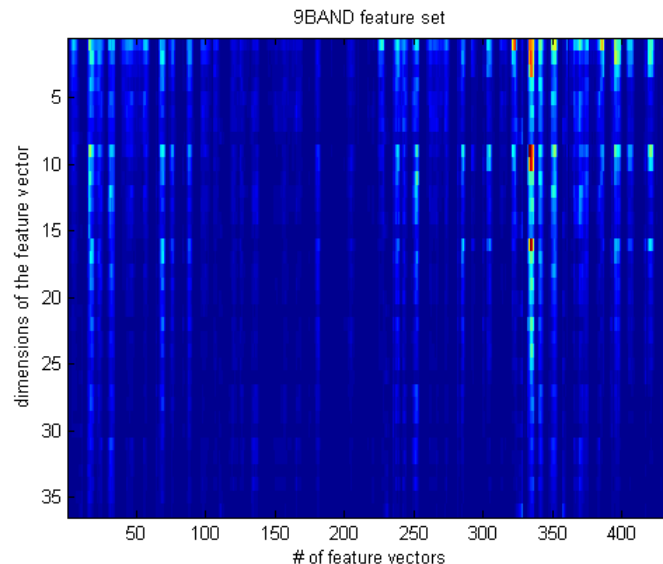


Figure 2.3: 9BAND feature set of a voice file

est) map unit (neuron). By training the neurons move through the data space adapting to the inherent structure of the data generating a sort of skeleton of the data. While the grid resides in the same space as the input, it's inherent dimensionality is much lower. One can visualize this by imagining the grid as a crumpled piece of paper. While it occupies a 3-D space the piece of paper is still intrinsically 2-D. The empirical bias of this map is that it assumes similar data are in close proximity of each other and distance is a good measure of similarity. The SOM-toolbox is used as a general purpose Matlab implementation of the SOM algorithms¹. The toolbox allows to generate and train SOMs. It also contains several related algorithms such as LVQ [9], several data pre-processing and visualization tools. For the research done the supervised training algorithm was used instead of the standard unsupervised one because the data has the class information readily available. Supervised learning is chosen since it improves quality, reduces noise influence and generates easier to interpret results. Two possible lattices can be used with the toolbox, either hexagonal or rectangular as can be seen in figure 2.4.

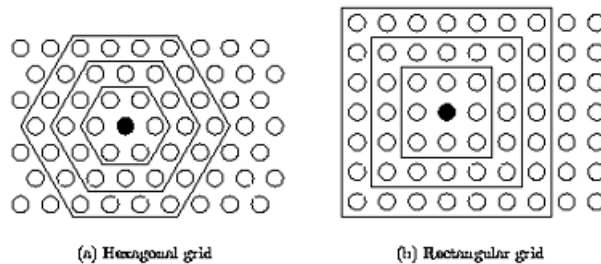


Figure 2.4: Two kohonen grid forms

The size of the generated map can also be specified and since classifying a feature on the SOM is a quick procedure, and training a non-recurring event in classification we chose to use the a 'big' map as this yielded better results in the classification phase. The default size for the map structure as implemented in the SOM Toolbox is generated by the heuristic formula:

$$\text{neurons} = 5 * \text{dlen}^{0.54321} \quad (2.2)$$

Specification of a 'Big' map increases this number of neurons 4-fold. The actual classification is done by presenting the network with a feature vector and looking up the label of the Best Matching Unit (BMU). The feature is then classified in accordance with the BMU's label. For the multi-speaker detection a vector of BMUs and their corresponding labels is returned to the multi-speaker detection algorithm

2.2.3 K-means

The KNN algorithm is computationally intensive as the dataset increases in size. Therefore the K-means clustering algorithm is applied to reduce the size of the

¹<http://www.cis.hut.fi/projects/somtoolbox/>, freely available under the GNU license

known data. Reducing the dataset increases the speed of classification by the KNN algorithm. However reducing the known data will make the feature space it spans more general, which is likely to affect the test results. Finally the effect of noise and artefacts in the data is likely reduced. The K-means algorithm is a clustering algorithm based on the attributes of the data. The K in K-means represents the number of clusters the algorithm should return in the end. As the algorithm starts K points known as centroids are added to the data space. To achieve a 1:10 reduction on the original data K is set to 10 percent of the database. The next step is to assign all points in the data to the nearest of these centroids. After this assignment each of the centroids finds the centre of this data. The centroid places itself in this centre, hence the name centroid. This process is iterated for a number of steps reducing the intra cluster distance. In the end the centroids represent the general characteristics of the data and are used as input for the classification.

2.2.4 Evaluation

After the frame by frame classification the test sample as a whole is evaluated. For single speaker samples the number of correctly classified frames per sample is divided by the total number of frames in the test sample. For a multiple speaker sample a different evaluation is needed. To achieve this, a matrix with not only the best, but also the second, the third classification and so on is created. From this matrix a vector is generated for each speaker. This matrix and its derivative vectors are used to detect different speakers even if both speakers speak at the same time. Another advantage of this method is that it makes it more robust against noise and false positives. These vectors are generated and evaluated in the following ways.

The naive linear evaluation generates the vector counting the occurrences of a specific speaker in each frame. This vector is treated as a binary vector. The linear evaluator has a specified block threshold, if it is reached the speaker is considered to be present in this sample. The algorithm continues to evaluate the vector and increases the resulting output. At the end this resulting value is normalized and returned.

The weighted linear evaluation is a weighted version of the naive linear evaluation. The difference is in the way the vector is evaluated. While the linear evaluation treats the vector as binary, the weighted linear evaluator takes the quotient of the vector and the number of positive speakers. Furthermore while the linear version uses a block threshold, the weighted version uses a weight threshold. When evaluating the vector the weight variable is increased with the quotient till either the weight threshold is reached, or a zero is encountered. If a zero is encountered the weight variable is reset to zero. If the weight threshold is reached the speaker is considered to be present in the sample. If a speaker is found the weight variable is reset and the rest of the vector is evaluated as well. Similar to the linear evaluator, every time the weight threshold is reached the result count is increased. At the end, this result is normalised as well. The advantage of the weighted linear classifier is its increased flexibility. It can detect speakers in a shorter span if their voice is highly prevalent and puts more emphasis on the quality of the classification.

The resulting value of these evaluators is a ratio between the identified speakers. It should be noted that every non zero value indicates the presence of the re-

spective speaker. The values of these ratios are purely for comparing prevalence between the speakers.

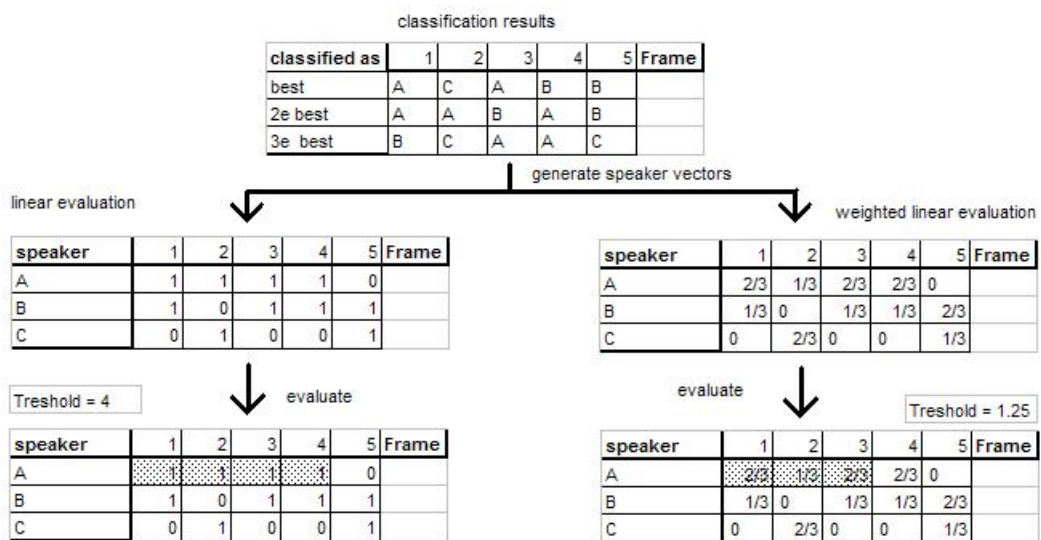


Figure 2.5: Visualization of the evaluation process

Chapter 3

Testing

The following chapter will describe how the testing environment was set up, what it is composed of, what tests were run, and its results.

3.1 Recording of Samples

For the recording of the voice samples two female and two male speakers have been used. It was decided to take both female and male recordings, so gender specific differences could be analyzed. The recordings were done in the same environment so similar conditions were provided for all recordings. This was done to prevent the influence of background noise and have comparable recordings. For the recording a Plantronics Headset connected to a Creative Soundblaster Live! 5.1 Player were used. The recording settings are a sample rate of 22.050 kHz, 8 bit depth, and mono. With each speaker fifteen specified sentences (see B.1) were recorded. All samples were edited for leading and trailing noise, silences, long parts of noise and silence in between the sentences. Noise samples were recorded as well.

3.2 Setup of the tests

For the test a modest Personal Computer has been used (see table 3.1 for details).

CPU	AMD Athlon 64 3200+
RAM	1 GB
OS	Fedora Linux 6
Software	wine-0.9.29, Matlab-2006b

Table 3.1: testing hardware

3.3 1. Test: Calibrating Parameters

The first tests were used to find optimized parameters for the algorithms. These preliminary tests gave an indication on the quality of the databases derived from the recorded samples, specifically completeness and over fitting. This was done by implementing a cross validation and a batch testing module with different combinations of the algorithm parameters. The obtained results were analyzed and used as configuration for the later tests.

3.4 2. Test: Single Speaker Recognition

The second test was conducted to test the quality of single speaker detection. For this setup the database was composed of only fourteen samples per speaker to obtain results from known data as well as results from unknown data. The tests were run with all possible combinations of algorithms, feature sets, datasets, sample files and speakers.

3.5 3. Test: Multi Speaker Recognition

The final test was done to see how the speaker detection works in a multi speaker environment. For this purpose multi speaker samples were created by concatenating samples from different speakers. For this reason, measurement of the results was possible. The test was run with all different multi speaker samples for all combinations of algorithms, feature sets and datasets. In addition it was run multiple times to find the optimal parameters for the evaluation algorithms.

Chapter 4

Results

4.1 Test 1: Calibrating Parameters

The first calibration test was run with the standard number of dimensions, 13 [7]. At these settings different settings for the SOM were tested. The size of the SOM's grid has been tested at the settings small and big, which are default options for it. The results from these tests show a 25 percent increase in performance figure 4.1.

The other parameter tested for the SOM was the setting of either a hexagonal

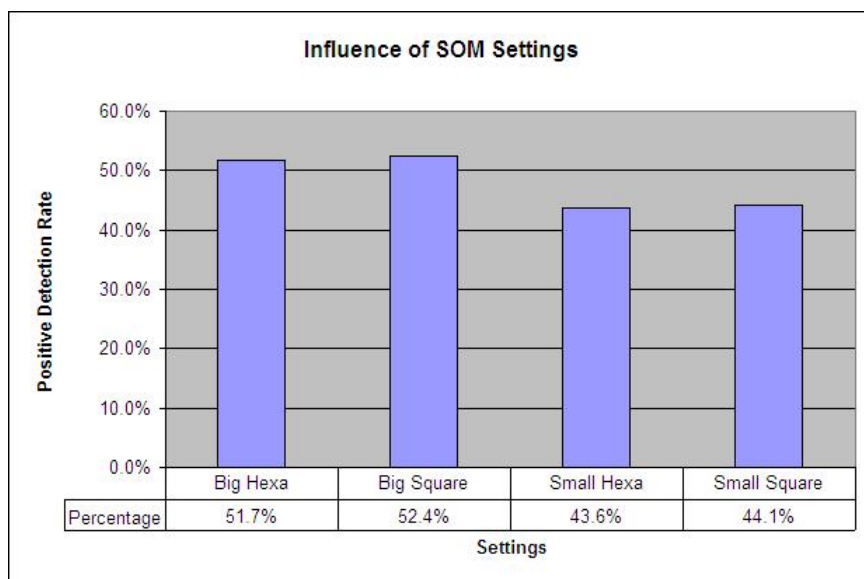


Figure 4.1: Influence of SOM settings

or square lattice. In this test small differences were found but differences are too small to be significant and are likely due to sample size. As a consequence of these results the SOM was set to use a 'big' hexagonal lattice for the remaining tests.

4.2 Dimensionality

After the initial calibration of the SOM, the performance of the SOM was tested with feature sets of different dimensionality. In these test the following settings for the dimensionality of the feature sets were used 9, 11, 13, 15, 17 and 29. The results of this test are visualized in figure 4.2.

As can be seen the increase of performance between the settings between nine

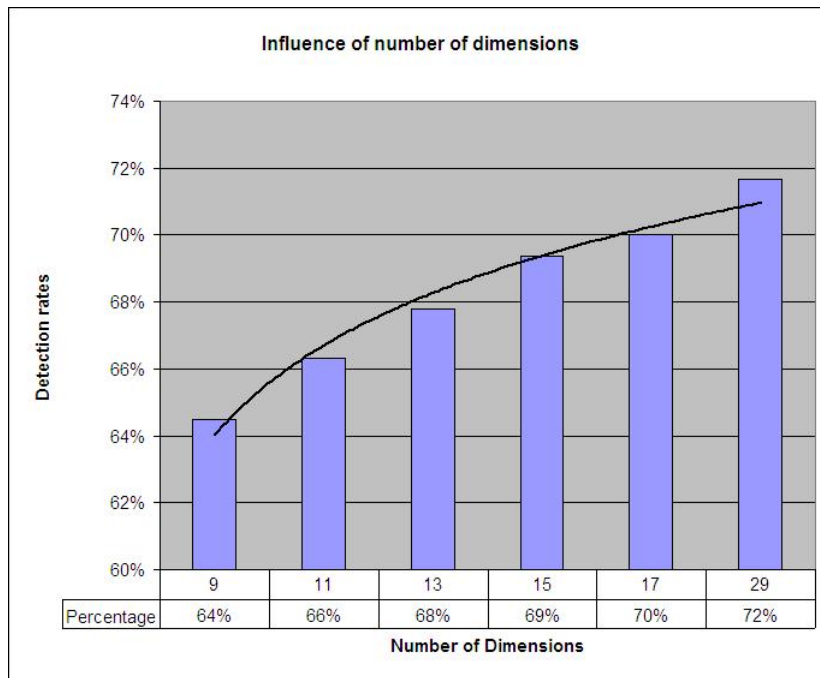


Figure 4.2: Influence of number of dimensions

and 13 are still significant in relation to the increase of dimensionality. Even if the dimension of the feature set is more than doubled (13 to 29) there is only a 4 percent increase in detection rates. The trend line in the graph clearly shows the declining rate of growth after the 13 dimensions. This corresponds with the used literature [7].

The results of the cross validation, with 13 dimensions and the above mentioned Kohonen settings the next graph is summarized in figure 4.3.

With the Nearest Neighbour classifier RASTA yields slightly better results. However, with the SOM Rasta has only 40 percent detection. On Average MFCC yields the best results with a detection rate of above 65 percent, Rasta has a mixed result based on the classifier and finally 9BAND performs mediocre with results averaging 50 percent.

4.3 Test 2: Single Speaker Recognition

The results of single user recognition are visualised in (4.4). The average results,

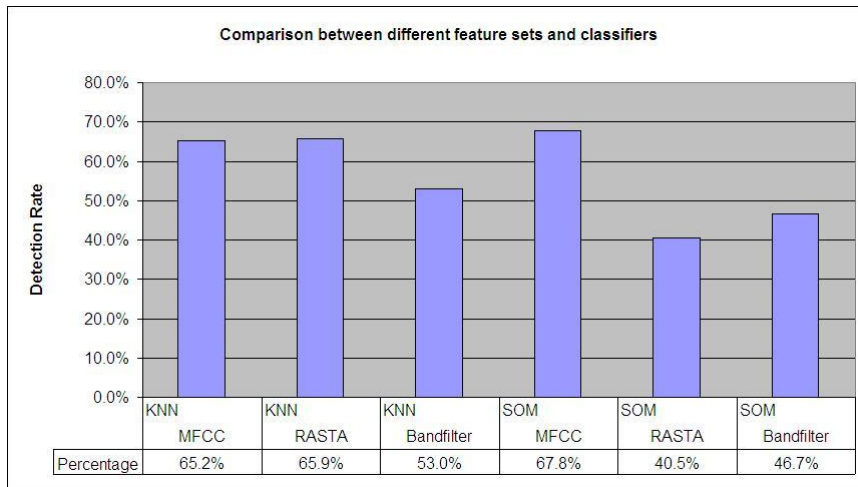


Figure 4.3: Comparison between feature sets and classifiers

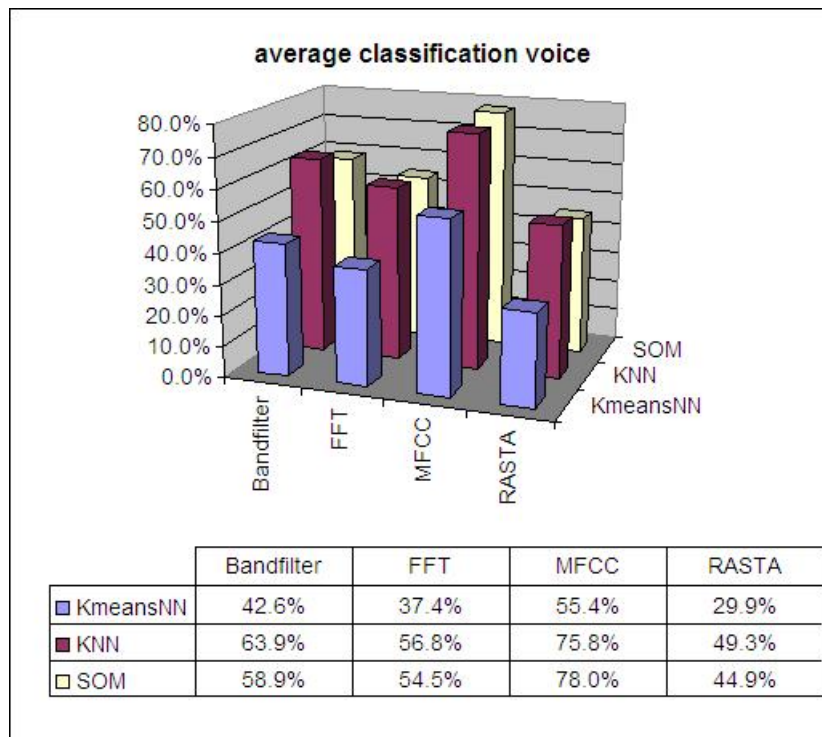


Figure 4.4: Average classification voice

show that the difference between NN and SOM is only marginal. Further more SOM and KNN perform very similar if noise is added (see figure 4.5). While

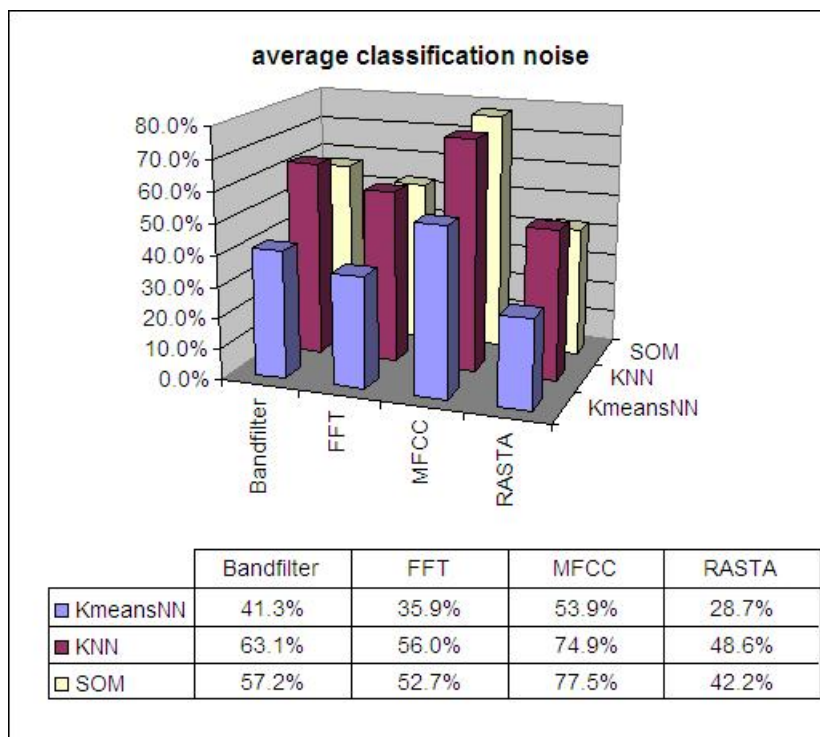


Figure 4.5: Average classification noise

the amounts of correct classifications are similar SOM performs slightly worse on eliminating false positives (see figures 4.6 and 4.7).

The different feature sets do result in fairly different results. RASTA performs even worse then the base line FFT, 9BAND performs only marginally better, but MFCC clearly yield superior results to the other three feature set. Finally the K-means reduced KNN yield worse results then KNN but the running time is greatly reduced (see figure 4.8). However, the performance and running time are still worse then the SOM.

4.4 Test 3: Multi Speaker Recognition

After running tests for several values of the threshold, the speakers present in the samples where all correctly detected. However speakers not present in the samples where also detected. These results indicate the correct threshold has not been reached yet. The evaluation layer combines information from multiple hits produced by the classification algorithm and therefore should provide more accurate results than without it. When looking at the detection of the noise this effect is already present, most samples are classified as containing voices, not noise (see figure 4.9).

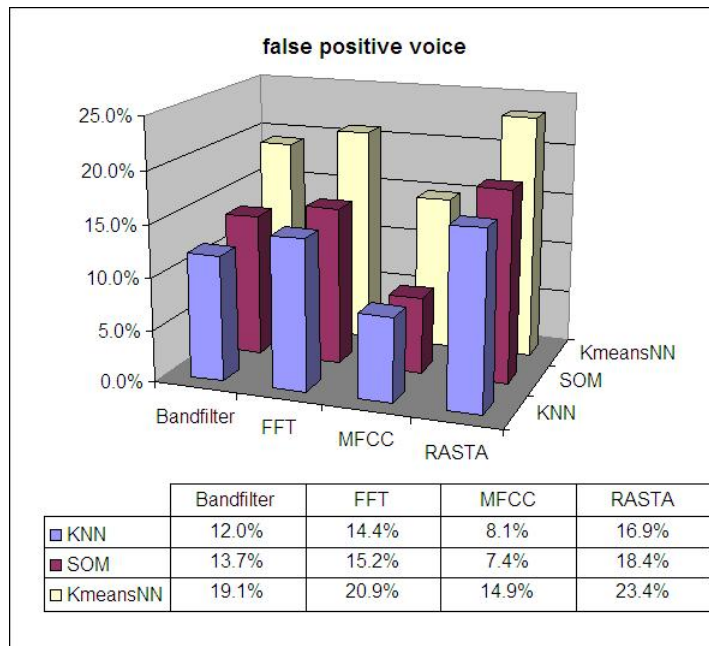


Figure 4.6: False positive voice

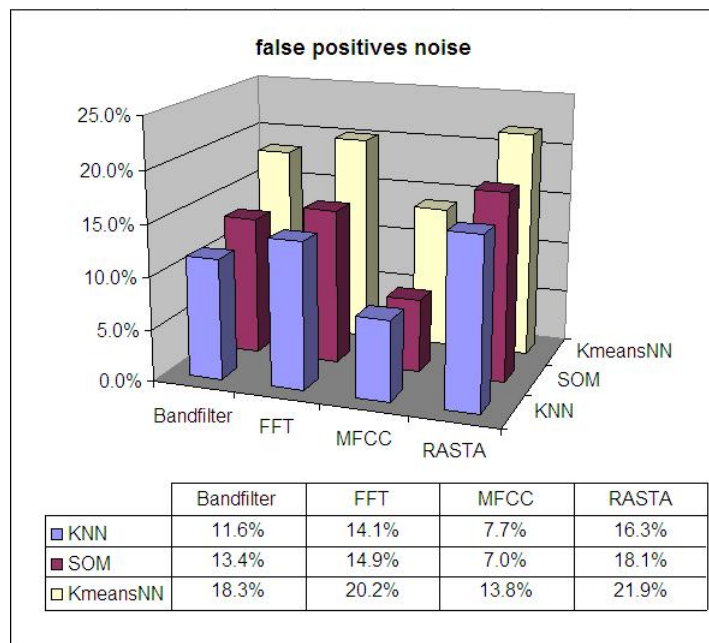


Figure 4.7: False positive noise

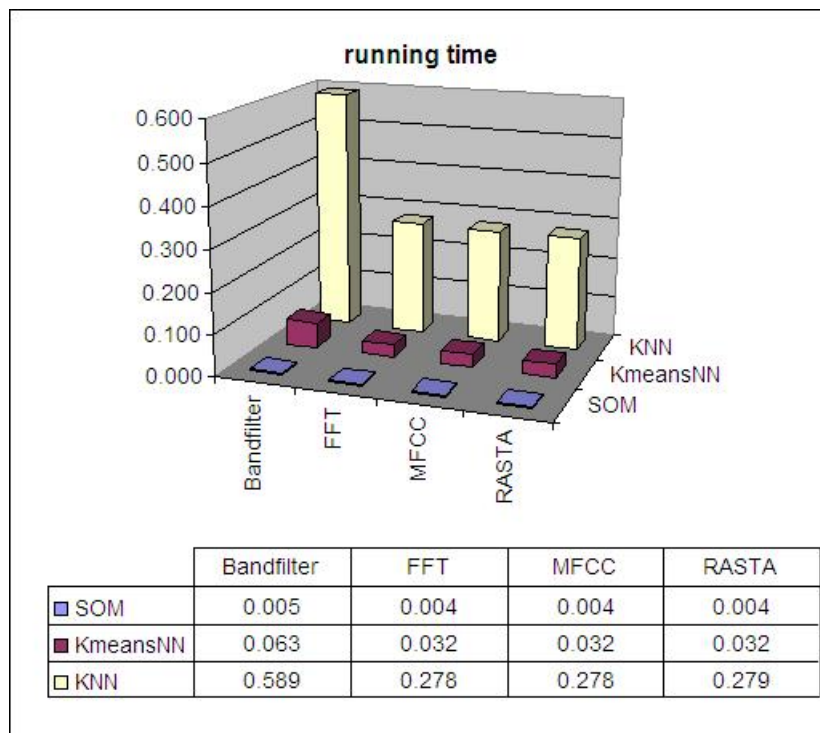


Figure 4.8: Running Time

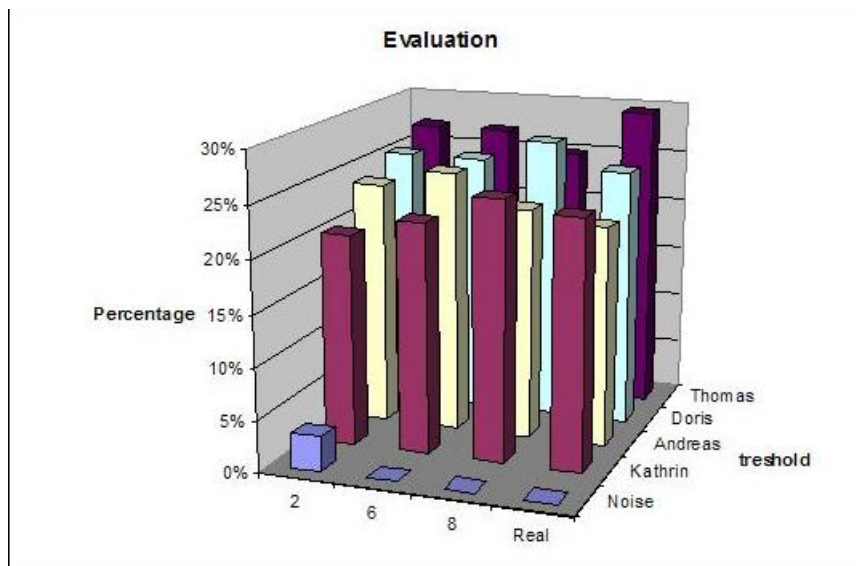


Figure 4.9: Evaluation multi-speaker

Chapter 5

Conclusion

5.1 Summary

The results from the first two tests have shown that some feature set algorithms generally work better than others. Some of the results are as expected, e.g. 9BAND works better on finding differences between voice and noise than inter voice difference. It is interesting to see that the RASTA feature set did not work out as well as expected. One reason might be that the filter function used is not optimal for preserving the important differences between. On the other hand the MFCC feature set which is similar to RASTA performed well. The naive Fourier Transform feature set as seen in the result chapter performed better than expected. For detection of different speakers as in a single- or multi speaker environment as well as for voice and noise distinction MFCC proved to be the best feature set tested.

Using the cross validation test to find the right parameters proved to be a good starting point for testing and finding good parameters to start real world testing. The evaluation layer improved the results and served to improved detection of whether a speaker actually speaks in multi speaker environment or not.

The classifiers both performed very well. The classification with the SOM is fast and as such could even serve in a real time voice recognition application. The training was extremely short and can be disregarded compared to the time needed for the creation of the feature sets. The KNN classifier also delivered good results, but at a cost of running time (28 hours). The positive aspect of not needing training was therefore rendered meaningless. By using the k-means reduced data the running time was significantly reduced, at the expense of reduced accuracy. Because of this the SOM classifier is the right choice here. There are still a lot of improvements that can be made starting by the selection of the sample sentences (B.1) or the technical aspect of recording the samples. The algorithms can undergo more tuning with aspects derived from the results from test two and three.

5.2 Discussion

One step of further research could be to use different classification algorithms besides the KNN and SOM classifiers. Some research was done during the first

part of the project. One alternative could be the SASH [10] algorithm. However this is still experimental, in general this algorithm is not well known yet. Another step could be to extend the multi speaker detection tests, such as finding optimal settings for the threshold functions. Also real world multi speaker samples could be tested. This could give better insight into the performance of the algorithms if different speakers speak at the same time. For this a measurement of 'accuracy' needs to be defined as this test cannot rely on the same knowledge base the engineered samples from test three could. Next other feature sets could be added to the project framework and compared to the existing ones. Finally examples of other approaches and ideas for algorithms can be found in [7].

Appendix A

Additional Graphs

A.1 Project Flow

A.2 Results

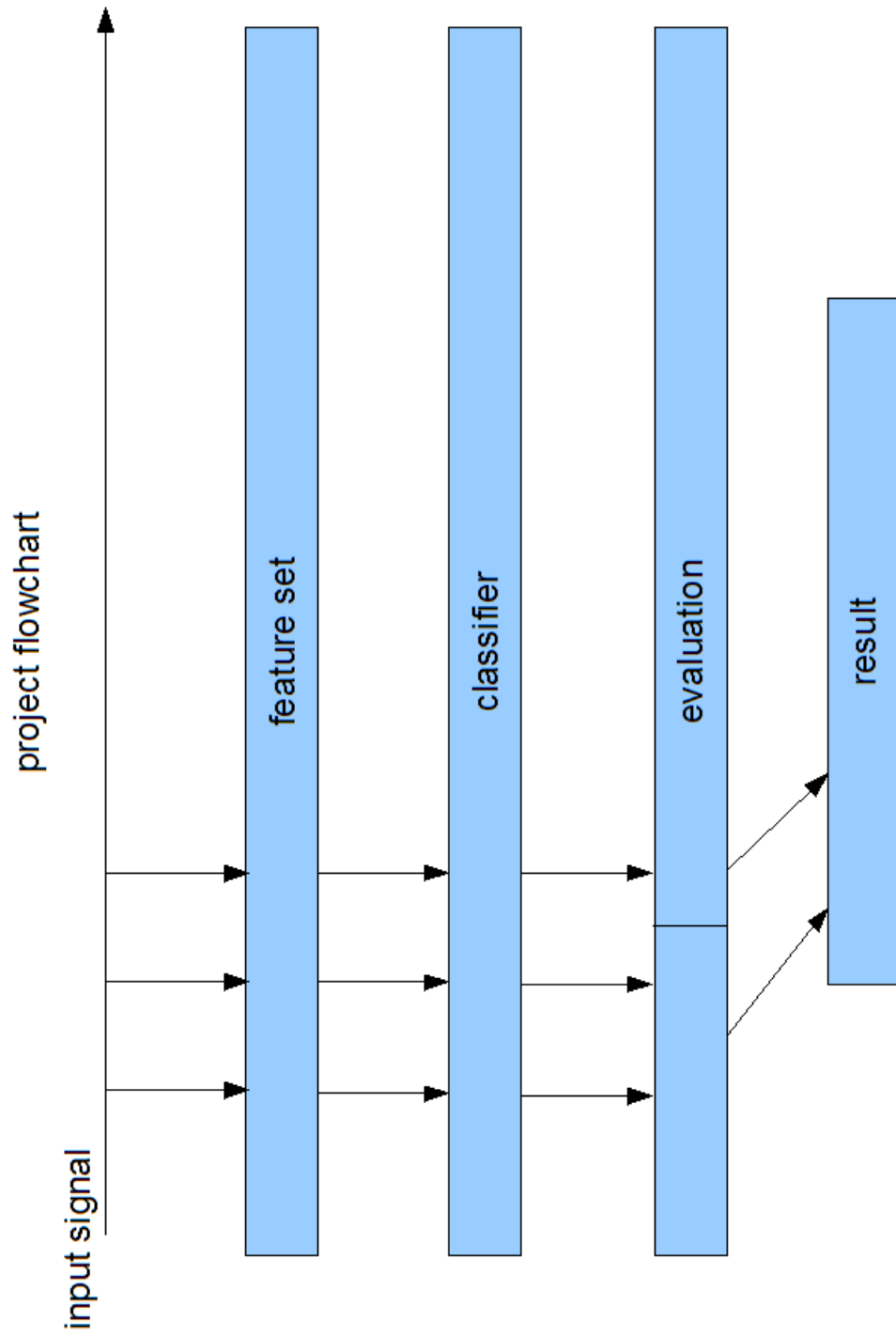


Figure A.1: Flow of the project during speaker recognition

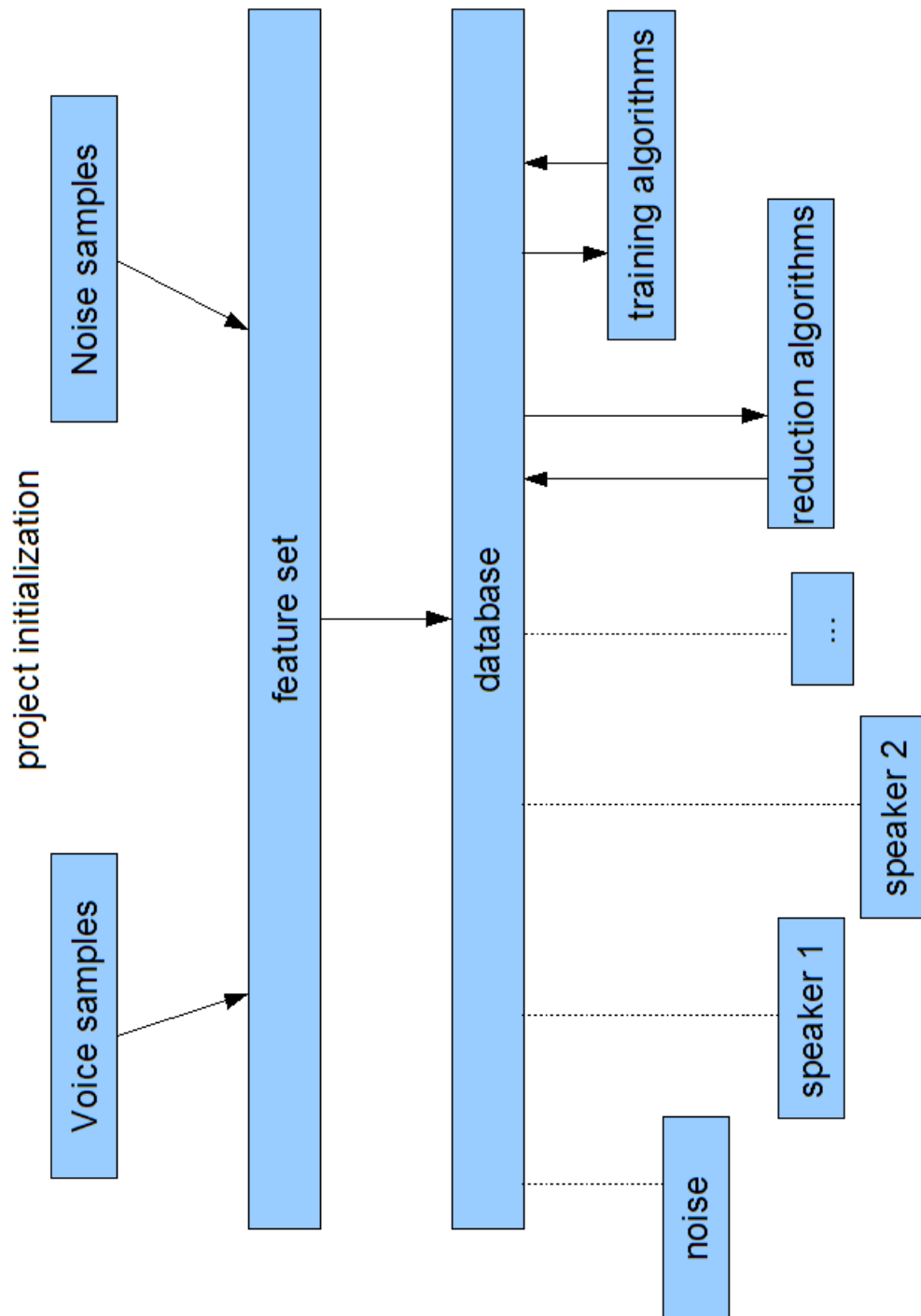


Figure A.2: Flow of the project during the initialization

Appendix B

Additional Material

B.1 Sample Sentences

1. test one two three
2. yes
3. no
4. this is a test for project 3.1
5. hello my name is 'your name'
6. it doesnt matter what you say
7. any sample of voice will do
8. recognise me please.
9. mail those samples
10. who's gonna get me a new cup of coffee
11. houston we have a problem!
12. the quick brown fox jumps over the lazy dog
13. to be or not to be
14. only two more samples
15. and we are done for the day

Bibliography

- [1] Joseph P. Campbell, Jr: Speaker Recognition: A Tutorial, Proceedings of the IEEE, VOL. 85, NO. 9, September 1997
- [2] Chih-Wen Weng, Cheng-Yuan Lin, Jyh-Shing Roger Jang: Music Instrument Identification Using MFCC: Erhu as an Example, Chinese Music Dept, Tainan National College of The Arts, Taiwan, Computer Science Dept, Tsing Hua University, Taiwan, Computer Science Dept, Tsing Hua University, Taiwan
- [3] Thomas Kerwin: Classification of natural language based on character frequency, June 2006
- [4] Beth Logan: Mel Frequency Cepstral Coefficients for Music Modeling, Cambridge Research Laboratory
- [5] Daniel J. Jacobsen: Probabilistic Speech Detection, IMM-THESIS-2003-50
- [6] Hynek Hermansky and Nelson Morgan: RASTA Processing of Speech, IEEE Transactions on Speech and Audio Processing, VOL. 2, NO. 4, October 1994
- [7] Ronald A. Cole: Survey of the State of the Art in Human Language Technology, November 1995
- [8] Timo Honkela: Self-Organizing Maps in Natural Language Processing, Helsinki University of Technology, December 1997
- [9] Teuvo Kohonen: Self-Organizing Maps, Springer, 3rd Edition, May 2006
- [10] Michael E. Houle, Jun Sakuma: Fast Approximate Similarity Search in Extremely High-Dimensional Data Sets, Proceedings of the 21st International Conference on Data Engineering (ICDE 2005), IEEE 2005
- [11] Petra Hendriks: BreinMakers and brein brekers, inleiding in cognitieve wetenschap, Pearson Education, 1997
- [12] Wikimedia Foundation, Inc.: Window Function, 23-01-2007, http://en.wikipedia.org/wiki/Hamming_window