Course Project 7 - 5ME20 "Ubiquitous Computing and Activity Recognition"

Voice-based Human Recognition

Dijk, E.T. van – 0749866
Jagannathan, S.R. – 0786488
Wang, D. – 0758165
Eindhoven University of Technology

Abstract

In this project we have tried to come up with a practical and widely applicable system for speaker identification which is as robust as possible against influences of naturally occurring sources of within-speaker variability of the speech signal. As a basis for our system we have used an existing system, implemented in MATLAB, which uses either Mel Frequency Cepstral Coefficients (MFCC) or wavelet Sub-Band based Cepstral parameters (wavelet SBC) as features and a Gaussian Mixture Model to infer speaker models. Based on literature and initial results, we chose to implement a number of changes that would improve the system's robustness and practical applicability. The final system is quite robust against the investigated sources of variability. Nevertheless, much work needs to be done to truly achieve a text-independent, robust speaker identification system that can be directly applied in a wide variety of practical situations.

1. Introduction

As computers become more embedded in our daily lives and everyday activities, the race is on to find ways to make the interactions with these systems easier, more natural and more tailored to the user. In order to enable a system to adapt to its user, however, it first needs to identify the user. Voice-based human recognition is a promising approach, as it is relatively unobtrusive and does not require explicit, physical interaction with the user like most alternatives do (e.g. traditional keyboard-based identification or fingerprint scanning). Additionally, the only sensor needed for voice-based recognition is a simple microphone, which can already be found on many personal mobile devices (e.g. cell phones, PDA's and laptops).

1.1 Speaker Verification vs. Speaker Identification

Two main types of voice-based human recognition exist: speaker verification and speaker identification. In speaker verification the system attempts to verify that the person currently speaking is the 'right' person (e.g. the owner of the cell phone the program is running on, or the CEO who is permitted access to a certain hard drive). That is, an incoming signal is compared to known characteristics of a particular person's speech. A speaker identification system also tries to establish a speaker's identity, but in this type of system, an incoming signal is compared to multiple speaker models to see which (if any) fits best.

While speaker verification is used mainly for security, speaker identification has a wide variety of applications in fields such as ubiquitous computing, ambient intelligence, robotics and any other technology that requires matching content and operation to the current user.

1.2 Challenges in Speaker Identification

One of the most important issues in speaker identification is the fact that the speech signal differs not only between people, but also between different utterances of the same person. Different circumstances can cause people to speak loudly or in a whisper, slowly or quickly, slurred or clearly etc. Additionally, many people speak more than one different language. The language a person speaks at a given time may depend on context as well. Moreover, certain health issues (e.g. nasal congestion) can change the timbre of the voice. All of these sources of within-speaker variability of the speech signal make it difficult to distinguish one speaker's voice from another.

Speaker identification becomes even more difficult if there are no restrictions to the content of the speech that is to be recognized. Matching one utterance to another utterance of the same text, so called text-dependent recognition, is easier than matching that utterance to an utterance of an entirely different text, so called text-independent recognition. In practice, however, the latter type of system has a much wider range of applications, as it can also operate on naturally uttered speech and does not require explicit interaction of the user with the system.

1.3 Project Goals

As mentioned before, speaker identification can be used in a variety of practical applications. However, different types of within-speaker variability of the speech signal arise in daily life and these cannot be avoided when implementing a

speaker identification system in a real world scenario. Therefore, the goal of this project is to implement a text independent speaker identification system that is as robust as possible against different types of within-speaker variability of the speech signal.

2. Method

The main goal of this project is to examine and exclude the influence of different types of naturally occurring within speaker variability of the speech signal. The sources of variability we have chosen to examine are:

1. Text uttered
2. Speed of the speech
3. Language used
4. Nasal congestion

2.1 Recordings

A Sony ECM-C115 microphone was used for all recordings. All recordings were stored in a Waveform Audio File format (.WAV) with the sampling frequency of 8 kHz. As it was our intention to use only features of the signal that are unaffected by signal duration, the duration of the samples was not controlled. Samples were recorded for seven participants (four male, three female) and one additional participant whose samples were never used for training in order to simulate what would happen if the system were to encounter samples from an unknown speaker. Each participant was first asked to say out loud the following three sentences, as they would in a normal conversation:

1. My name is [name].
2. I come from [country].
3. The weather is fine today.

Participants were then asked to repeat the same sentences, but more quickly, as if they were in a hurry. Thirdly, participants repeated the sentences once again, but much slower than they naturally would. This condition simulates the type of speech that most people naturally use when a person (or computer system for that matter) fails to understand them at first.

The participants were then asked to pronounce translations of the three given sentences in the participant's native language (Mandarin for 5 of the participants, Dutch for one and Tamil for one), at normal speed. Lastly, participants repeated the English sentences again, but were asked to do so while they held their nose to simulate the sound of their voice if they were to suffer from nasal congestion. All of this resulted in a total of 15 utterances per participant, which are summarized in Table 1.

Table 1: Data Samples

The five conditions – each containing three different sentences – used for the recordings of each participant. Note: ID represents the code used for the recording; Content represents the semantic content of the utterance; Speed represents speaking speed; Language represents the language in which the participant speaks; NC (nasal congestion) represents whether or not the participant held their nose during speaking.

ID	Content	Speed	Language	NC
1_1	My name is [name].	normal	English	no
1_2	I come from [country].	normal	English	no
1_3	The weather is fine today.	normal	English	no
2_1	My name is [name].	normal	Native	no
2_2	I come from [country].	normal	Native	no
2_3	The weather is fine today.	normal	Native	no
3_1	My name is [name].	fast	English	no
3_2	I come from [country].	fast	English	no
3_3	The weather is fine today.	fast	English	no
4_1	My name is [name].	slow	English	no
4_2	I come from [country].	slow	English	no
4_3	The weather is fine today.	slow	English	no
5_1	My name is [name].	normal	English	yes
5_2	I come from [country].	normal	English	yes
5_3	The weather is fine today.	normal	English	yes

The recordings were carried out using the voice recognition software developed in MATLAB. To know how to record voice samples, have a look at the flash video developed for the system.

2.2 Recognition Algorithm

In order to identify speakers, a system is needed that (a) is capable of extracting important features from the audio signal, (b) can be trained to model in a way what these important features are for different speakers and (c) can, after training, match a new audio signal to the appropriate speaker model. As much research has been aimed at building such systems, we have opted to spend the time available for this project on trying to improve an existing

system, rather than reinventing the wheel by building a system from scratch. The basic system we have chosen to use is a freely available system which has already been implemented in MATLAB [2]. The general workings of the already implemented system are described below.

2.2.1 Feature Extraction

The basic speaker recognition system provides the user with two options in terms of feature extraction: Mel-Frequency Cepstral Coefficients (or MFCC for short) or wavelet sub-band based Cepstral parameters (wavelet SBC). The former is a set of features widely used for speaker identification. The basis of this type of feature extraction is a basic Fourier transform. However, the resulting spectrum is adjusted using among others the Mel scale, which indicates the actual amplitudes of each frequency necessary in order for them to be perceived as being of equal loudness. In a sense, this scale indicates the 'importance' of each frequency for human sound perception.

As in MFCC, the derivation of parameters for the SBC is performed in two stages. The first stage is the computation of bank energies, and the second stage would be de-correlation of the log bank energies with a Discrete Cosine Transform (DCT) to obtain the MFCC. The derivation of the SBC parameters also follows the same process except that the bank energies are derived from the Wavelet packet transform method rather than the short-time Fourier transform. SBC parameters are then derived from sub-band energies using the DCT. Although MFCC is most commonly used in speaker identification, systems that use wavelet SBC have been shown to give superior results under noisy conditions [1].

2.2.2 Classification.

The basic system uses Gaussian mixture models (henceforth GMM) for speaker modeling. In GMM, data is assumed to consist of a mixture of several Gaussians. Each Gaussian represents a certain source of variation. In our case, the combined variation in the extracted features of the training samples can be seen as a combination of the variation in the voices of participant one, participant two, etc. to participant seven. In GMM, we try to find a number of Gaussians (in our case 12 for each of the participant) with a certain mean and variance that, when combined, resemble the total variation in the data as well as possible. An incoming sample is then compared to all of these separate Gaussians to determine which of these distributions is most likely to produce the sample in question.

Others have used GMM for speaker identification with impressive results, achieving recognition rates of 96.8% and 80.8% for clean and telephone speech respectively [3]. Additionally, these authors compared performance of GMM to other speaker modeling techniques and found that GMM performed best.

3. Proposed Alterations

In order to gain a better understanding of what it takes to reliably identify a speaker, we have chosen to add a number of alterations to this basic system which will, hopefully, improve the system's usefulness in practical applications.

3.1 Alteration 1: Hybrid Model & 'Unknown' class

The basic system has two options for feature extraction: MFCC, which is generally considered to give the best results and wavelet SBC, which has been found to be superior when dealing with noisy data [1]. In order to optimally exploit the characteristics of both feature extraction methods, we have chosen to implement a 'hybrid model' which uses both features. Additionally, we want to add an 'unknown speaker' class as a possible match. If a sample of a previously unknown speaker is presented to the system, it will likely be a poor match to all of the known speakers. However, the 'best' match, although poor will still be presented as the identity of the speaker. In real-world applications, it is quite likely that strangers will be heard by the system from time to time; the system should respond by saying the speaker is unknown rather than guessing, as the basic system essentially does. To implement this feature, we have added a clause to the hybrid model which states that if both the MFCC model and the wavelet SBC model cannot find a good enough match, the sample is classified as 'unknown'.

For the hybrid model, two sets of speaker models are trained, one with MFCC and one with wavelet SBC. Both the MFCC-based and the wavelet SBC-based model give results as a set of numbers representing the 'quality' of the match to the current sample. Two thresholds exist for both models: a strict threshold (i.e. if a match is higher than this threshold, it is considered to be really good) and one that is more lenient (i.e. if the match is below this, the match is really poor and if it is in between both thresholds it is moderately good). The values used for these thresholds were derived and improved upon using the ones already given in the basic system.

See Figure 1 for a flowchart of the decision process in the hybrid model. For each to-be-classified sample, the classification process can operate in one of several ways. If the MFCC model has a match that is above the strict threshold (i.e. a

really good match), this model's result is given regardless of the results of the wavelet SBC model. This is done because we assume that if the MFCC model has a good match, the sample is not noisy and can be classified better by using MFCC. If the SBC model has a really good match and the MFCC model only a moderate match, the wavelet SBC model's result is used. Here we assume that if the MFCC model is 'unsure', the sample may be noisy and should be classified using the wavelet SBC model. If both models have only moderate results, the result used depends on the relative quality of both matches. Finally, if one or both of the models has a really poor match (i.e. below the lenient threshold) the sample is classified as coming from an unknown person.

3.2 Alteration 2: Additional Training

In the basic system, it is not possible to use more than one training sample for each speaker model the system is to infer. However, we have assumed from the start that there can be major differences between different samples, especially if they are recorded under different conditions. It may therefore be beneficial to train the system not on one specific sample, but on a number of samples from different conditions. We believe this will make the resulting model more generalizable and as such better suited to recognize samples recorded under different conditions.

To train the altered system, each Gaussian is first modeled on one sample to determine the Gaussian's mean and variance. This model is then used as a base and adjusted according to the second sample, etc.

3.3 Alteration 3: Multiple Models for Each Speaker

When we train the system on different samples from different conditions, as described above, we simply take 'the average' of one person's voice under different conditions. This may not be the most appropriate approach. As we are trying to describe each person's voice with a cluster of a Gaussian distribution, we assume that the distribution of 'normal' (English, normal speed, no nasal congestion) samples is no different than that of samples that are spoken more quickly, slowly, in a different language and with nasal congestion. It may well be the mean and/or variance of the distribution that best describes one voice's data is very different for each condition.

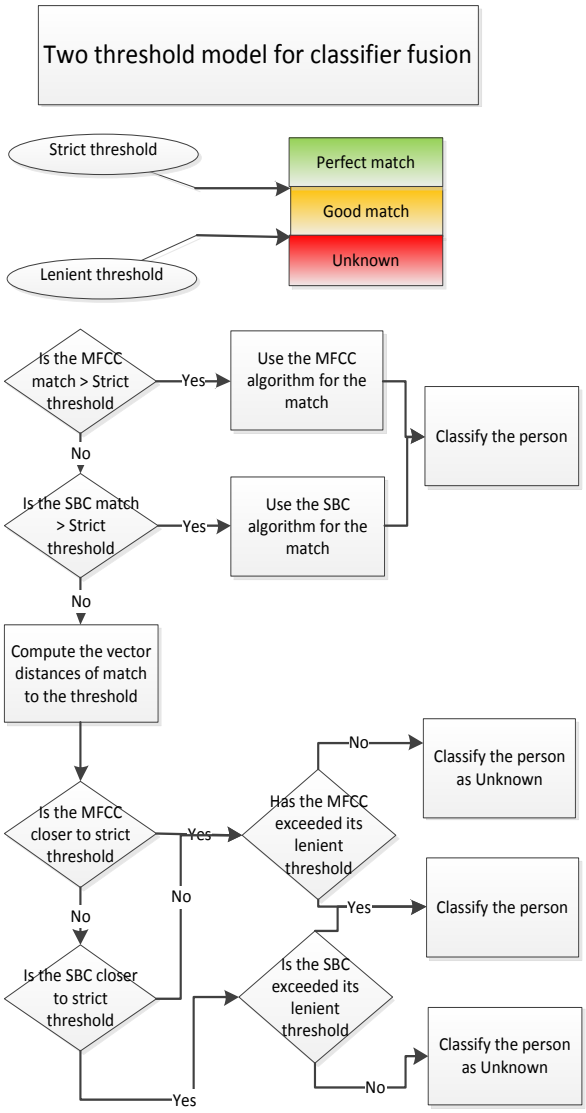


Figure1. Flowchart of the Decision Process in the Hybrid Model

For example, when we speak more slowly, the pitch of our voice almost automatically becomes lower as well, which would translate to a different mean of the Gaussian model that describes the samples.

In order to overcome this issue, the system should infer not only one model per speaker, but rather one model per condition/speaker combination. In our system, this would mean inferring five models for each speaker: one for the speaker speaking normally, one for the speaker speaking quickly, one for the speaker speaking slowly, one for the speaker speaking in their native language and one for the speaker speaking with a congested nose.

3.2 Alteration 4: Borda Count implementation

In addition to the threshold classifier fusion method, the Borda count method was also implemented. The Borda count is a group consensus function which works on the basis of majority vote. For each class, the Borda count is the sum of the number of classes ranked below it in each classifier. The final output is taken from ranking the magnitude of the sums. The highest Borda count is assigned the highest rank [4]. However an additional threshold is required for detecting the unknown samples. The threshold for detecting the unknown samples in this case is same as the lenient threshold in Figure 1.

The Borda count method was implemented on the system S3, which has been trained on multiple models for each speaker.

4. Data Analysis

Two kinds of results are of importance to this project: recognition performance of the system and ease of use of the system. Although the latter is difficult to measure objectively, the former can be put into numbers with relative ease. In order to track changes in recognition performance as we adjust the system, we will test the original system, as well as each intermediate step in our alteration process. This results in a set of five systems to tests, which are summarized in Table 2.

For each system, we will determine the recognition rate (percentage of correctly identified samples), both overall and under the different conditions separately. Additionally, confusion matrices will be obtained, both for the overall recognition of each system, as well as for recognition of 'normal' samples.

In the confusion matrix the number in each cell depicts the percentage of samples of the class on the horizontal axis that was predicted by the system to be of the class on the vertical axis. For ease of interpretation of the confusion matrices, each cell is shaded depending on the number in it: a higher number means a darker shade. A diagonal of dark cells with 100's in them, then, would indicate perfect performance; any high numbers outside the bottom-left to top-right diagonal indicate the system's weak points.

The samples from each participant that are used for training and for testing is not the same across all the systems. The sections below summarize which samples are used for training and testing in each system.

Table 2:
Definition of the Six Different Systems to be Tested.

System	ID Description
S0.1	Basic system with MFCC
S0.2	Basic system with wavelet SBC
S1	Altered system with alteration 1
S2	Altered system with alterations 1 & 2
S3	Altered system with alterations 1,2 & 3
S4	Borda count implemented on S3

S0.1 & S0.2. In both S0.1 and S0.2, only one training sample is used to infer each subject's speaker model; we have opted for sample 1_1 (see Table 1). For each subject, the remaining 14 samples (i.e. samples 1_2, through 5_3) are used to test the system. Overall, then, the basic system uses 7 samples to infer seven speaker models, while $14 * 7 = 98$ samples are used to test the system's performance. Note that for each speaker; only two samples from the 'normal' condition (normal speed, no nasal congestion, English) are used for testing, whereas three test samples from all other conditions are used.

S1. For easy comparability of results, this version of our altered system is tested in the same way as the basic system: to train each subject's speaker model in the basic system only one sample was used per subject; we have again opted for sample 1_1 (see Table 1). For each subject, the remaining 14 samples (i.e. samples 1_2, through 5_3) are used to test the system. Additionally, S1 is tested using 15 samples from a previously unknown speaker (participant 8). As the system is not trained on any of this subject's samples, all 15 samples are used for testing. Overall, then, 7 samples are used to infer seven speaker models, while $14 * 7 + 15 = 113$ samples are used to test the system's performance. Again note that, for each of the 'known' speakers, only two samples from the 'normal' condition are used for testing, whereas three test samples from all other conditions are used.

S2. To train each subject's speaker model in S2, ten samples per subject are used: two from each of the five conditions (i.e. samples 1_1, 1_2, 2_1, 2_2, 3_1, 3_2, 4_1, 4_2, 5_1 and 5_2; see Table 1). For each subject, the remaining five samples (i.e. Samples

1_3, 2_3, 3_3, 4_3 and 5_3) are used to test the system. Additionally, system 2 is tested on all 15 samples from the previously unknown speaker. Overall, then, S2 uses $10 * 7 = 70$ samples to infer seven speaker models, while $5 * 7 + 15 = 50$ samples are used to test the system's performance

S3. In S3, five speaker models are inferred for each participant. For each of these models, two samples from the relevant set of samples are used for training (i.e. samples 1_1 & 1_2 for the 'normal' model, samples 2_1 & 2_2 for the 'fast' model, samples 3_1 & 3_2 for the 'slow' model, samples 4_1 & 4_2 for the 'native language' model and samples 5_1 & 5_2 for the 'nasal congestion' model; see Table 1). For each participant, the remaining sample from each condition is used to test the system (i.e. samples 1_3, 2_3, 3_3, 4_3 and 5_3). Here, too, the system is tested using 15 samples from a previously unknown speaker. Overall, then, the final system uses $10 * 7 = 70$ samples to infer $5 * 7 = 35$ speaker models, while $5 * 7 + 15 = 50$ samples are used to test the system's performance.

S4. In S4, the training and testing sequences are same as S3, except the classifier fusion works on Borda count.

5. Results

See Table 3 for recognition rates of the five different systems on different conditions. Overall confusion matrices for each of the five systems can be found in Figure 2; confusion matrices for 'normal' samples are shown in Figure 4.

Table3:
Recognition rates of the five different systems on different conditions.

Condition	S0.1	S0.2	S1	S2	S3	S4
All	58%	34%	33%	52%	62%	60%
Normal	64%	50%	41%	80%	80%	80%
Fast	67%	38%	38%	40%	50%	70%
Slow	52%	29%	17%	30%	60%	50%
Native language	71%	38%	46%	50%	50%	60%
Nasal congestion	38%	24%	25%	60%	80%	40%

6. Discussion

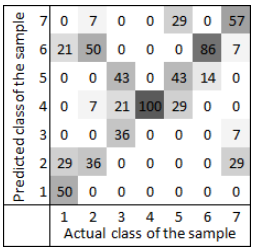
6.1 Effects of Within-Speaker Variability of the Speech Signal

Our results confirm that 'normal' samples (normal speed, English, no nasal congestion) are easiest to classify. For systems S0.1, S0.2 and S1, this comes as no surprise, as the system was trained on one of the 'normal' samples. However, even with multiple training samples from different conditions (as in S2), recognition rates are highest for 'normal' samples. Additionally, even when specific models are trained to deal with the different conditions (as in S3), 'normal' samples are classified most successfully. This needed further investigation.

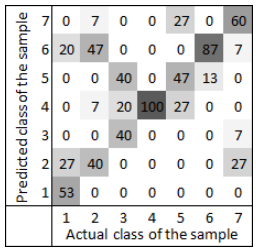
Statistical modeling done by the GMM uses Expectation Maximization (EM) algorithm. In the EM algorithm an initial model is obtained by the estimation of parameters from the clustered feature vectors. The proportions of vectors in each cluster serve as mixture weights. Means and covariance are estimated from the vectors in each cluster. After the estimation, the feature vectors are re-clustered using component densities (likelihoods) from the estimated mixture model and then model parameters are recalculated. This process is iterated until model parameters converge. So, if the samples have overlapping clusters, then the algorithm would not be able to differentiate clearly between different speakers. This can also be inferred from the scatter plots shown in Fig 3. The samples within this area have high probability of getting classified incorrectly. The cross clustering area is comparatively less in S3 for 'normal' samples. This is the main reason behind the high accuracy of the 'normal' samples.

6.2 Effects of Between-Speaker Variability of the Speech Signal

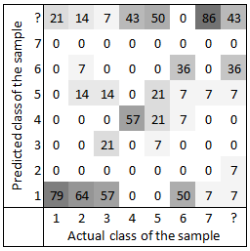
Note that participants ('classes') 3, 4 and 5 in the confusion matrices are female, whereas participants 1, 2, 6 and 7 are male. The confusion matrices show that male voices are rarely (only in the difficult-to-classify nasal congestion samples) confused with female voices or vice versa. Female voices are, however, likely to be confused with one another (which shows in the confusion matrices as darker cells in the nine center squares), as is the case for male voices (which shows in the confusion matrices as darker areas in all four corners).



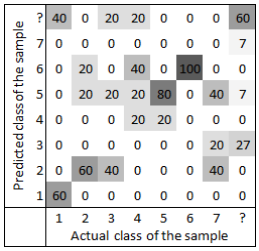
(a) S0.1 (basic with MFCC)



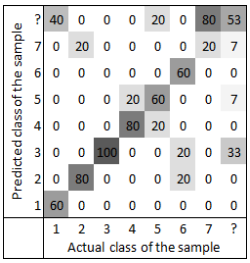
(b) S0.2 (basic with wavelet SBC)



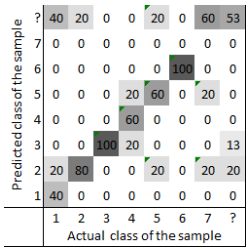
(c) S1 (hybrid model)



(d) S2 (additional training)



(e) S3 (multiple gaussians)



(f) S4 (Borda Count)

Figure 2. Overall Confusion Matrices of the Different Versions of the System.

6.3 Effects of Different System Versions

S0.1 & S0.2. Others have suggested that wavelet SBC may be superior to MFCC under noisy conditions [1]. However, we have found no evidence to support this. S0.1 has a higher recognition rate under all conditions than S0.2. Moreover, S0.2 seems less robust against influences of the differences between conditions than S0.1.

S1. As can be seen from the recognition rates in Table 3, the results we have achieved with our hybrid model (S1) are quite close to those of the original system using wavelet SBC (S0.2). Closer investigation of the individual classification decisions in the hybrid model confirms that in most cases, results of the wavelet SBC model are used. Unfortunately, this means that results in S1 are actually poorer than those achieved by the basic system with MFCC (S0.1). This indicates that the double-threshold method we use to fuse the outcomes of the MFCC-based model and the wavelet SBC-based model in the hybrid model is not optimal. This is mainly because the Hybrid model uses the SBC results 64% of the time for arriving at the decision, and uses the reliable MFCC only 36% of the time. Alternative options for this problem are explored in the Limitations and Future Work section.

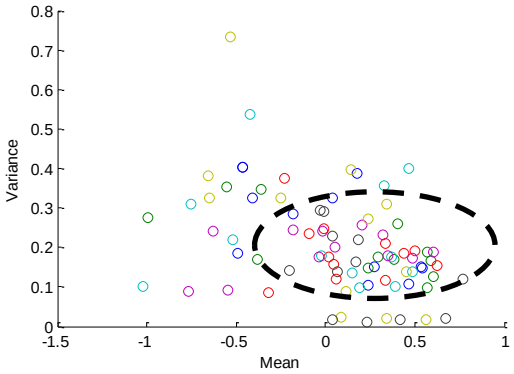


Figure 3a. Cross clustering area of a single Gaussian model (S1) for 'normal' samples. The probability of error is comparatively high

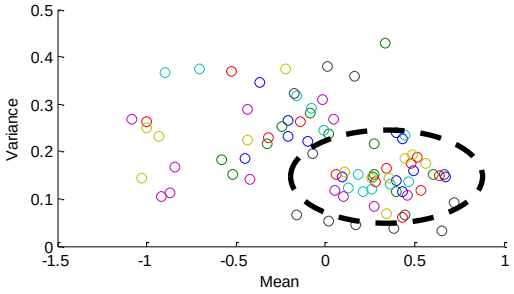
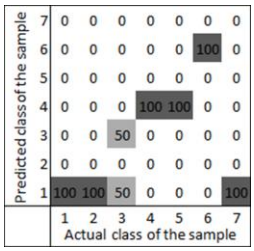


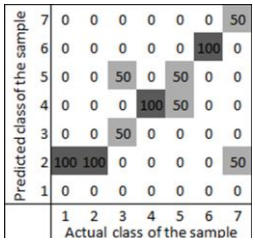
Figure 3b. Cross clustering area of a multiple Gaussian model (S3) for 'normal' samples. The probability of error is comparatively less

S2. Although the overall recognition rate of S2 is lower than that of S0.1 (52% versus 58%), the recognition rate for 'normal' samples is quite a bit higher: 80% versus 64%. Especially in comparison to the results of S1, the advantage of using multiple training samples is clear: the recognition rate for 'normal' samples has almost doubled (80% versus 41%). Interestingly, additional training did not – contrary to what we expected help the system to form a more generalizable model. In fact, is seems it has only helped the model to recognize 'normal' samples, while recognition rates for samples from other conditions are only marginally better. This

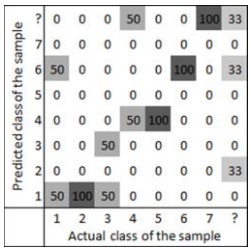
contradictory result is likely caused by the problem described in the section on Alteration 3: Multiple Models for Each Speaker.



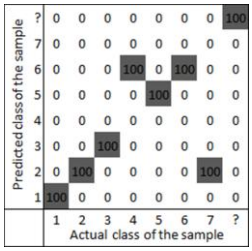
(a) S0.1 (basic with MFCC)



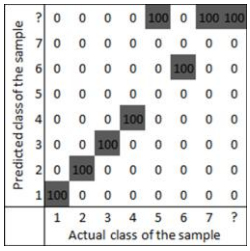
(b) S0.2 (basic with wavelet SBC)



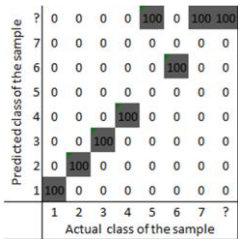
(c) S1 (Hybrid model)



(d) S2 (additional training)



(e) S3 (multiple gaussians)



(f) S4 (Borda count)

Figure 4. Confusion Matrices of the Different Versions of the System for Normal Samples.

Using only a single Gaussian distribution cluster to describe all the variation in the data from different conditions is conceptually odd, since we actually expect the MFCC or wavelet SBC characteristics to be very different under different conditions. For this reason, we have also implemented S3, to investigate whether tackling this issue – by using a specific Gaussian model for each condition – can improve recognition results.

S3. The final version of our system, S3, has the highest overall recognition rate of all versions of our system: 62% of all test samples are recognized correctly. Although the recognition rate for 'normal' samples is no higher than that of S2 (both 80%), S3 does seem to be more robust against influences of the sources of variability we have examined, as we had hoped. Additionally, as we can see from Figures 2(e) and 4(e), the 'unknown' class is working very well in S3. Most samples that are not correctly classified are seen as 'unknown'. In the confusion matrices, we can see this from the fact that almost all instances are either in the bottom-left to top right diagonal (where correctly identified instances are located) or in the top row (where instances classified as 'unknown' are located).

Overall, we have been able to exclude variations of the text uttered quite well, as we set out to do. Although the quite high recognition rate of 64% for 'normal' samples in S0.1 indicates that the basic system could already generalize over different utterances quite well, we have certainly improved upon this initial performance with our final system. Moreover, where the robustness of the basic system was far from optimal, we have certainly improved the system's ability to exclude the sources of within-speaker variability of the speech signal we have investigated: variations of the speech signal caused by speed, language and nasal congestion are handled more appropriately and with more success.

6.4 Effects of Classifier fusion strategies

The systems S3 and S4 are comparable as they use the same kind of initial training and test samples. Borda count method of fusion is effective in cases where the actual class is ranked higher in either of MFCC or SBC. Though the overall confusion matrix looks better than S2, but S3 based on hybrid model still outperforms Borda count method. This is mainly because Borda count method treats both the MFCC and SBC equally. But we have already observed that the MFCC outperforms the SBC on various samples. Hence the strategy of treating classifiers equally and implementing a Borda count method doesn't create much improvement in the system performance.

7. Limitations and Future Work

Although we have learned much about the possibilities and difficulties of human speech recognition in this project, much work remains to be done before our system can be employed in practical applications. In this section a number of remaining issues and possible alleys for improvement are described.

7.1 Classifier Fusion Strategies

As described in the Proposed Alterations section and Figure 1, we initially used a two-threshold technique for classifier fusion in our hybrid model. The thresholds that are set are highly tuned to the data available. Hence this can give rise to scalability problems when the system is implemented in real world applications. To solve this problem, the Borda count fusion method was implemented. But as the Borda count fusion method treats both classifiers equally, the overall recognition rate decreased marginally. In order to optimally merge the output from both classifiers in a practical way, a logistic regression approach could be used, wherein a regression equation is extracted from the system that describes the extent to which each classifier's output predicts the actual class of the training samples.

In this way, the classifiers are weighted according to their predictive power [4]. Although this approach rids us of one threshold, another would still need to be in place to allow the system to decide to classify a sample as 'unknown'. More research is required to find ways to automate the setting of this threshold, or avoid it altogether. An important feature with the current application that is developed is that, the top 3 ranks available in each approach can be seen via the details tab. Refer to the flash demo for more information. This can be useful in taking any decisions regarding the new classifier fusion techniques to be used.

7.2 Alternative Classifier

We have seen from our results that samples from each separate condition can be described with a Gaussian distribution with some success (as in S3). However, the complete set of samples from different conditions cannot be accurately described with a Gaussian distribution. Although our current approach of using multiple models for each speaker works in theory, there are some issues in practice. First of all, each model needs to be trained separately, which places a large burden on the user. Additionally, we have limited ourselves to five sources of within-speaker variability in this research; quite a few more may exist in real-life speech. Determining how many and which models are needed may well prove a non-trivial task.

Rather than combining multiple simpler models, as we have done, one could also try adopting a modeling technique, such as artificial neural networks, that can handle more complex models.

7.3 Higher Sample Quality

From our results (especially the differences between S1 and S2), the benefit that is to be gained from additional classifier training is clear. However, in practice we do not want to bother the user with repeated requests for training samples. Rather than using more training samples, it makes sense to try to get more out of each individual training sample instead.

By using a higher sample rate (16 KHz), different variations of speech beyond the current 8 KHz can be captured. By increasing the duration of the current samples, continuous speaker variability can be captured in a natural way. Thus we give the system more information to work with, while not putting too much of a burden on the user.

7.4 Streaming Audio and Recognizing Real-Time

An additional option for improvement concerns the system's interface. At the moment, each sample of speech has to be recorded separately manually and then accessed from the MATLAB system. In real-world applications, one may want to incorporate both of these actions into one automatic program, avoiding the need for excessive amounts of user input. By automatically clipping incoming speech signals to samples of, say, 5 seconds and using these as direct input for the MATLAB program, the system can be used to gather data without explicit instructions from a human user.

This does, however, introduce a need for the recognition of speech as a general type of sound, compared to, e.g. silence and natural background noise. If we were to feed any and all sounds to our system, it would try to recognize speakers in silence and all kinds of natural background noise (traffic noise, the sound of people walking, dropping something, putting their shoes on, etc.). Again, more research is necessary to find a workaround to avoid this issue.

8. Conclusion

In this project we have tried to come up with a system for speaker identification. Our main goal has been to make our system more practical and widely applicable as possible. The most important aim therefore was to make the system as robust as possible against influences of naturally occurring sources of within-speaker variability of the speech signal, namely a) the text-uttered b) the speed of the speech c) the language used and d) whether or not the person suffers from nasal congestion.

As a basis for our system we have used an existing system, implemented in MATLAB, which uses either Mel Frequency Cepstral Coefficients (MFCC) or wavelet Sub-Band Cepstral parameters (wavelet SBC) as features and a Gaussian Mixture Model to infer speaker models. Initial results from

this basic system showed that the system was quite vulnerable to influences by a number of sources of within-speaker variability of the speech signal.

Based on literature and initial results, we chose to implement a number of changes that would improve the system's robustness and practical applicability: firstly, we opted to use a hybrid approach to fully exploit the strong points of both MFCC and wavelet SBC as they have been described in literature. Additionally, to make the system more widely applicable, we have added to the system the option of classifying an incoming sample as 'unknown' as real-world scenario's is likely to require this option. To help the system form a more generalizable model of each speaker we trained the system on several samples, recorded under different conditions. To better deal with the differences between samples taken under different conditions, we have implemented an approach where one speaker model per condition/speaker combination is used, rather than one general model per speaker.

The final system does quite well at generalizing over uttering different texts. The recognition rate of 'normal' samples (English, normal speed, no nasal congestion) is higher in the final system and most sources of within-speaker variability of the speech signal are no longer a major issue in our final system.

Overall, much has been learned in this project about the possibilities and problems surrounding text independent, robust speaker identification. Results from our project suggest that results others have described in literature do not always generalize to similar situations. Moreover, many systems de- scribed in literature need to be extended or adjusted to be used in practice and we have learned from our own experiences that these kind of adjustments may take away from the system's performance in recognition. Concluding, we can say that much work needs to be done to truly achieve a text-independent, robust speaker identification system that can be directly applied in a wide variety of practical situations.

References

1. Farooq, O., & Datta, S. (2004). Wavelet based robust sub-band features for phoneme recognition. Vision, Image and Signal Processing, IEEE Proceedings, 151, 187-193.

2. Ram, R. (2010). Speaker recognition system. Available from http://www.mathworks.com/matlabcentral/fileexchange/27059

3. Reynolds, D. A., & Rose, R. C. (1995). Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE Transactions on Speech and Audio Processing, 3, 72-83.

4. Ward, J. A., & Tröster, G. (2006). Activity recognition of assembly tasks using body-worn microphones and accelerometers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28, 1553-1567.