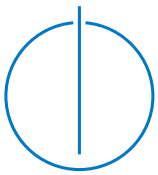
FAKULTÄT FÜR INFORMATIK

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis

Deriving Conversational Social Contexts from Audio-Data

Hanna Jasmin Schäfer



FAKULTÄT FÜR INFORMATIK

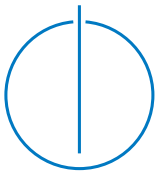
TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis

Deriving Conversational Social Contexts from Audio-Data

Erschließen von gesprächsbezogenen sozialen Kontexten aus Audio-Daten

Author:	Hanna Jasmin Schäfer
Supervisor:	Dr. Georg Groh
Advisor:	Florian Schulze
Submission Date:	15.09.2015



I assure the single handed composition of this master's thesis only supported by declared resources.

Munich, 15.09.2015 Hanna Jasmin Schäfer

Acknowledgments

First of all I would like to express my great appreciation to my supervisor Dr. Georg Groh for giving me the opportunity and resources to work on this thesis. I would also like to give my sincere gratitude to my advisor Florian Schulze for his continuous support during this work. His guidance, motivation and expertise lead me through all phases of this thesis.

Abstract

This thesis aims at processing recordings of human conversation and characterizing them regarding their social context. The social characterization is part of a larger system predicting the interruptibility of smart phone users by recording and classifying their conversations. The aspects used as characteristics are speaker emotions on the one hand and speech affects on the other hand. All predictions are based solely on auditive reception without processing the spoken text.
The first part of the thesis focuses on studying different emotion classification methods on the benchmark datasets Berlin-EMO, FAU Aibo and SEMAINE. Although all datasets have been processed and classified by previous research, the focus of this approach is to include mayor disturbances such as noise and different smart phone channel effects. Also the experiments are limited to the use of Mel Frequency cepstral coefficients (MFCCs) as input data in order to reduce the data transfer from the smart phone. The goal of imposing these limitations is to test the known classification methods within a real life environment. The algorithms applied are Gaussian Mixture Models (GMMs), iVectors, Anchor Models and Random Forests.
The second part of the thesis focuses on predicting lower level features, namely speech affects, such as valence, arousal, power and expectation. These affects could be used for direct prediction of the interruptibility or as an intermediate step towards predicting the emotion. For applying and testing the regression of these affects, the SEMAINE dataset is used as a benchmark. Again the data is disturbed by noise and channel effects as well as reduced to MFCC features. The algorithms applied are Support Vector Regressors, Random Forest Regressors and Long short-term memory recurrent neural networks (LSTM-RNNs).
For the emotion recognition the GMM algorithm showed the best performance with 29.75% UAR in average over all datasets and audio types. The overall low quality of results is caused by the limitations of a real life application. For the affect regression, the best performance was reached by the Random Forest Regressor with R2 values of -1.60 for valence, -0.65 for arousal, -2.84 for expectation and -1,36 for power. Although these correlations seem very low, a transformation into a tenaer classification task (-1,0,1) reaches average recall values of 52.84 for valence, 56.04 for arousal, 53.02 for expectation and 54.010739 for power averaged over all audio types.

Contents

1 Introduction

1.1 Objective

The most common social interaction between people is a dialog. Thus research within the social computing domain has long taken efforts to analyze and classify human dialogs regarding their speech, the interacting speakers and the overall dialog situation. Besides the general features of speech such as speaker and text recognition, a social analysis requires further information about the dialog. The most common way to socially classify a dialog is by the emotions detected within the speech. Additionally those emotions can be broken down into lower level features such as whether emotion is positive or negative and how agitated the speaker is. Those features are called the affects of speech. The recognition of either or both descriptors allows for machine learning approaches to classify the communication situation. The specific use case regarded in this research is the detection of whether a person should be interrupted by a smart phone in the current dialog situation. Figure 1.1 shows the overall scenario pipeline [35].

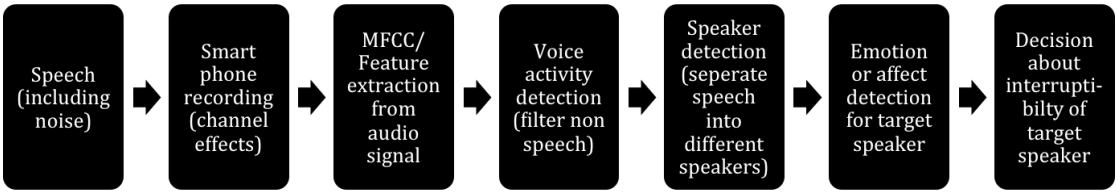


Figure 1.1: Processing pipeline for the interruptibility study.

In this pipeline, the filtered and annotated data can be used for example to detect emotions and affects of a certain speaker. Other parts of this pipeline have already been researched and developed [42][4][36]. In order to simulate the steps of this pipeline the data needs to have the same disturbances as a real dialog recorded with a smart phone

would have. Surrounding noise, the recording channel effects of different smart phones and the sound distortion due to the position of the phone within a pocket deform the audio signal. The disturbances then influence the correlation between the audio signal and the social characteristics. For a clear definition of the given circumstances, this chapter will shortly define the work process and the considered emotions, affects and disturbances. Finally a short overview of the structure of this document is provided to the reader.

1.2 Definition of process

In order to reach the final step of detecting the emotion or affect of a speaker, a process needs to be followed, that simulates the given pipeline (figure 1.1). Since this research should test various different approaches before being integrated into the overall pipeline, the process is reduced to testing on given benchmark datasets. The following steps are necessary in order to classify the characteristics of a pseudo-real dialog retrieved from a benchmark dataset:

1. Read signals from sound files
2. Mix signals with background noise
3. Convolve noised files with channel effects
4. Retrieve features from the convoluted sound signals
5. Assemble features in windows of 2,5 seconds
6. Filter non-speech parts from all windows
7. Remove windows with less than 50% speech
8. Read benchmark annotation of emotion or affect
9. Average annotations over each assembled window
10. Split windows and annotations into training and test set
11. Train classifier or regressor between windows and their annotations
12. Test predictions of classifier and regressor against real annotations

1.3 Definition of affect and emotion

Emotions in general are a complex combination of psychological, biological and social influences. For the purpose of characterizing dialogs this definition needs to be reduced to its relevant components. The object of our investigation is human speech in the form of recorded audio signals. Any human speech can be split into two different lines of information [2]. The first message is the explicit information transferred in the form of language. Given a natural language processing algorithm, this information may be transformed into text form and analyzed for emotional key words. The second line of information is the implicit message the speech sound contains about its speaker such as gender, age, health and the current emotions. The correlation between speech and emotion is always subjective and may be received differently by the conversational partner than the speaker intended to. The bases of a decision about the emotional state in a human receiver are features such as the speech rate, intensity and pitch [2] of the speech. Those observations then lead to the perception of different affects of speech. The four most common affects are valence, arousal, expectation/novelty and power. For example both happy speech and angry speech show a high value of arousal, which can be detected by the speech rate and the variation of the frequencies. To distinguish happy from angry, one needs the valence affect, which defines, whether an emotion is positive or negative. In an automated system all affects and emotions need to be learned from human made labels, since there is only an implicit correlation to the speech itself. Such a system can be trained on discrete emotion categories, or on the different affect dimension values. The most common way to interpret the lower level affect values into common emotions is the valence-arousal-scheme shown in figure 1.2.

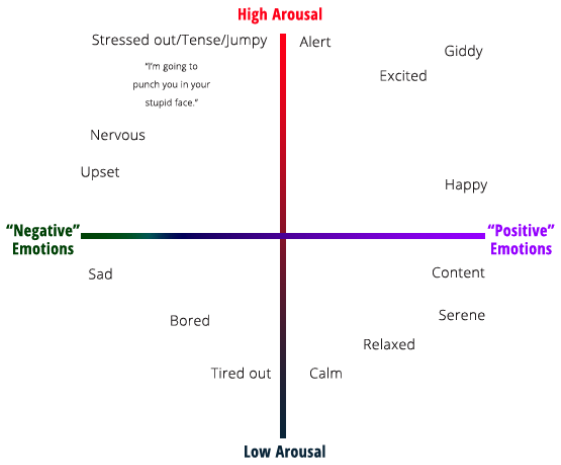


Figure 1.2: Correlation of emotion with valence and arousal values. Source:[44]

This scheme, first introduced by Russel [31], shows which emotions can be deduced from the values of valence and arousal. A simplified version only considers whether those values are positive or negative and thus splits all values into 4 emotions: Happy, Angry, Sad, Relaxed/Serene. The five standard emotions [2] also include Happy, Angry and Sad, but replace Serene with Fear and Disgust. Those are emotions easily detected from a human perspective and thus of high interest in automatic detection. To recognize all those emotions by the affect values the valence-arousal-scheme is not sufficient. Fontaine criticizes in his work [13], that a two dimensional model misses important notions in human emotions. For example fear and anger are within the same area of the valence-arousal-scheme, but are easily distinguished by the power affect value. In the scope of this thesis the valence-arousal-scheme will be sufficient for labeling emotional categories of the SEMAINE dataset. Both other datasets are already annotated with emotional classes and thus need no transformation from affects to emotions. The names of all recognized emotions are: Neutral, Happy, Angry, Sad, Relaxed/Serene, Fear and Disgust.

1.4 Definition of disturbances

To transform human speech into readable data, the digital audio signal needs to be extracted from the audio recording. There are different disturbances that can influence this sound signal. One part of the disturbances origins from the data itself. This part mostly consists of background noise, sound interference and low quality speech

itself. The other part of disturbances is introduced during the transformation process from speech to audio signal. The recording device itself and its surroundings like transportation medium, obstacles and interferences create those disturbances.
For the data disturbance, this research uses recordings of background noise and overlays the clean speech data with this noise. In order to simulate a varying noise strength the signals are mixed with a random signal to noise ratio (SNR) between -2dB and 20dB. Values greater than 0 indicate more signal than noise, while negative values indicate a very strong noise influence. Since the speech part of this mixing is a clean studio production, the low speech quality is not an issue.
The channel effects mainly simulate the processing disturbances. In real life a channel effect is caused by the distortion of the received sound signal by the channel it is transported through. Since the distortion caused by the channel is the same for any signal transported by it, its impulse response can be filtered and reused on other impulses. On the digital side this means we need to convolute the noised sound with the channel effect in order to reproduce what would have happened if this were the real transportation medium. The channel effects used in this thesis consider both the recording smart phone as well as different positions of the phone, like inside a pocket. In this way the data disturbances of noise, the processing disturbances of channel effects and sound distortion by the device position are reproduced for the clean benchmark datasets.

1.5 Structure of content

This thesis is structured into 7 main chapters. First this chapter (1) provides an overview of the content discussed in the thesis and of the structure it is presented with. After that chapter 2 provides insight into the state of the art concerning sound processing, emotion recognition and affect regression approaches as well as the influence of disturbances. In chapter 3 the different benchmark datasets that were used during the experiments are presented. Following this, the approaches for emotion recognition used in this thesis (chapter 4) are explained more closely. The approaches of the second part of this research, the affect regression, are shown in chapter 5. After the discussion of all circumstances the results of the conducted experiments are presented in chapter 6. Finally all content is summarized in chapter 7 and an outlook into possible future research is given.

2 State of the art

2.1 Sound features

The first step in emotion recognition is to extract features from the sound signals that can be used as input during the classifications step. In general there are two main types of features one can extract from sound signal [4], namely spectral and prosodic features. Short-term spectral features like linear prediction cepstral coefficients and the features used in this research, Mel Frequency cepstral coefficients (MFCCs) are calculated on small frames of around 0,02 seconds per frame. Prosodic features on the other hand rely on longer-term analysis and represent features like the pitch, stress, energy and tempo on a word or phrase level. In current research the openSMILE tool [12] was often used to extract all speech related features. It supports prosodic low-level features such as pitch contour and loudness. It also provides many audio low-level descriptors like MFCCs and LPCCs and many others. Successful research [11], [41] in recent years used mostly the following low-level descriptor features (LLDs), extracted with the help of openSMILE:

- Raw Signal: Zero-crossing-rate
- Signal energy: Logarithmic
- Pitch: Fundamental frequency, Cepstrum and Autocorrelation, Probability of voicing
- Voice Quality: Probability of voicing
- Spectral: Energy in bands, roll-off point, centroid, flux, and rel. pos. of spectrum max. and min
- Mel-spectrum: Band 1-26
- Cepstral: MFCC 0-12
- FFT-Spectrum
- LPC Coefficients and Perceptual Linear Predictive Coefficients

In contrast to the variety of input features in [11], [41] this research is restricted to the usage of MFCCs, which will later be relevant to understanding the results of the speech analysis.

2.2 Emotion recognition

Older approaches to emotion recognition used Support vector Machines and Hidden Markov Models. The most common approach in early and recent times is the UBM GMM from which MAP adapted GMMs are extracted [30]. From this model the variation of using iVectors was derived in more recent research. For example [47] describes how iVectors are derived from MAP adapted GMMs as a special type of supervector [24] in the total variability (TV) space. Another model [3] only recently published are the anchor models, which map a training or test set onto all other models and then compare the similarity between these performances. The specific results of these state of the art approaches are compared to the results of this thesis in the discussion section (6).

2.3 Affect regression

As for many fields of research also the affect regression has been currently researched with neural networks. Specifically the Long-Short-Term-Memory Recurrent neural network [20][15] has been used by Schuller and Eyben [10] to predict the affect dimensions of the SEMAINE dataset with very good results. These forms of neural networks have been a special focus for sequence forecasting [16][17] in recent years and thus introduce the new possible feature aspect of time dependency in affect regression.
Another area of interest is the learning with deep neural networks, which was applied by Stuhlsatz, Eyben and Schuller [41] for improving emotion recognition.

2.4 Channel compensation

For compensating channel effects multiple new normalization forms have been introduced. First of all the transformation of models, especially iVectors, into the LDA and PLDA space [22] and the combination of this with a WCCN normalization [9] have been used as variability reduction techniques on the feature side. With regard to channel compensation by score normalization there are the standard cosine distance, the t- and z- norm [46] and more recent score normalization techniques such as the s-norm [23] and the modified cosine kernel [8]. A comparison of these new norms with the previous ones will be shown in the discussion section (6).

3 Datasets

3.1 Universal background model - TimitBuckeye

There are two different datasets used for the universal background model in this thesis. The first one is a mixture of the TIMIT [14] dataset and the Buckeye [29] dataset, which has been used as a standard dataset for many years. The TIMIT (Texas Instruments/Massachusetts Institute of Technology) dataset was recorded with a Sennheiser close-talking microphone at 16 kHz rate with 16-bit sample resolution. It contains 10 sentences spoken by 630 speakers from 8 mayor dialects. The total set contains 5.4 hours of data and is split into 30% female and 70% male speakers [14].
The Buckeye [29] corpus recorded 40 people for the duration of 1 hour per person. The speakers were evenly distributed against gender and age. The recordings were done in a room clear of noise and recorded with a close talking head-mounted microphone. The true purpose was only conveyed to the participants after their session guarantying natural speech behavior. The audio files provided have sample rates of 16 kHz and 16 bits per sample. The TIMIT Buckeye variant of UBM data was also used to predict the speech and non-speech parts of all other datasets, since it contained voice activity detection (VAD) annotations that could be used to train a VAD classifier.

3.2 Universal background model - VoxForge

For the second UBM we chose the VoxForge [45] corpus as data source. VoxForge is a more recently initiated open source speaker detection dataset that consists of multiple languages and was originally created for use in Open Source Speech Recognition Engines such as such as ISIP, HTK, Julius and Sphinx. Due to the large amount of data available, we only used 6 million frames of the dataset to build our UBM.

3.3 Emotion Data - Berlin EMO

The first emotional dataset [7] is called the Berlin-EMO set. Created in 1997-1999 it contains 10 German sentences spoken by actors with multiple emotions. The recordings were done by the Technical University of Berlin, Institute of Speech and Communication,

department of communication science. 20 listeners evaluated each enactment for the emotional content and naturalness. Overall the dataset consists of 535 sentence recordings done by 10 speakers and is available as 16 kHz, 16 bits, mono audio wave files. The Berlin-EMO dataset has the largest variety of emotions containing 7 different annotations. The emotions classified are namely: Happy, Bored, Neutral, Sad, Disgusted, Angry and Anxious. Although the dataset offers very clear and precise emotion examples due to the enactment, the size prevents it from providing optimal models for new incoming data.

3.4 Emotion Data - FAU Aibo

In contrast to the Berlin-EMO dataset the FAU Aibo [39] set is large enough to build sufficiently based models. On the other hand, it is not completely representative for emotional speech, since children did all recordings. The dataset contains overall 9 hours of German sentences spoken by 51 speakers of an age between 10 and 13 years. It was created in a project of the Friedrich-Alexander-Universitaet where children interacted with the Sony pet robot Aibo. The displayed spontaneous emotions where split into syntactically meaningful chunks and into words. Human labellers then annotated the chunks and words with 2, 4, 5 or 11 classes of different emotions. In case of this thesis, the four-class annotation was used, which contains the following emotions: Motherese, Neutral, Empathic and Angry. The data is available as 16 kHz, 16 bit, mono audio wave files.

3.5 Affect Data - SEMAINE

The third emotional dataset in this research is the SEMAINE [28] [27] dataset. In contrast to the other two datasets, it provides annotations about five affect dimensions: valence, arousal, expectation, power and intensity. This dataset was created by recording different dialogs of users with an enacted operator, who adopted one of 4 roles designed to emotionally influence the dialog. 3-7 raters annotated the dialog sessions for every 0.02 seconds. For the purpose of regression and classification, those labels were averaged over all raters. In order to also use the dataset for emotional classification this thesis used a mapping of valence and arousal values onto the 5 emotions: Happy, Relaxed, Neutral, Sad and Angry. Differences between the emotions were only differentiated by the binary value of valence and arousal. The neutral emotion was selected as the area around 0 with a maximal absolute value of 0.15 in any dimension. Since the dataset provides multiple recording channels an additional step of mixing them into one mono channel was required.

3.6 Noise Data

For mixing the benchmark datasets with real background noise, a separate set of noise recordings is required. The noise data used needs to meet several criteria. It should not contain any understandable speech, it should not be limited to a single sound, but represent a real life situation and it needs to integrate noise occurring due to the pocket movement of the recording device. These criteria were met by using the QUT-NOISE corpus, created at the Speech and Audio Laboratory at Queensland University of Technology, for different ambient noises such as a cafe, the street or at home. The dataset contains 10 hours of noise divided into different categories. Additionally further real life noises were recorded using devices like iPad mini fall 2013, LG G2, Nexus S, Samsung S3 and Samsung S3 mini. Those noises were mixed together by category and then combined with the situation specific pocket movement, such as a backpack during cycling for the outdoors noises. The combined noises used in the experiments were namely: foot_steps, pocketMovement, public_transportation, super_market_mall, bar_cafe_crowd, car, indoor_kitchen_office_living, outdoor_street_nature

3.7 Channel Data

Since the overall project considers recording dialogs between different speakers with a smart phone, the noised speech also needs to be convoluted with a smart phone channel effect. For this purpose the impulse responses of four different smart phones held in each of four different pocket positions were retrieved. The impulse retrieval took place in a dammed studio-recording environment without ambient noise using a Dynaudio BM6A MKII box. The sound submitted to the channel effect was a sine sweep from 40 Hz to 20000 Hz. A reference microphone was used to calculate the difference in sound created by the channel effect. The four positions used during recording are: on the desk, inside a leather jacket and inside a pocket upward or downward facing. The cell phones used were a HTC, a LG, a S1 and a S3.

3.8 Data preparation

3.8.1 Audio preparation

The first step in the processing of audio recording into input features is manipulating the audio files themselves. In order to not include differences in loudness during the recording, all files are peak normalized. Additionally, for the SEMAINE dataset, all sound channels need to be combined into one channel by adding them together. After this step, all sound files are normalized and represented by only one mono channel.

3.8.2 Mixing

Before mixing the sound with the noise files, all noise files are transformed into integer signals, to match the sound files. Afterwards both files need to be gain normalized in order to have them at the same gain level. The reference file for standard gain is "pink_ref". This is necessary to later use the Signal-Noise-Ratio (SNR) in the mixing. A snippet, matching the sound files length, is chosen from a random noise file. In order to avoid sharp sound changes, the noise snippets need to be faded in and faded out. Afterwards both sound and noise can be mixed. The SNR ratio at which they are mixed is chosen at random $\in$ [-2dB; +20dB]

3.8.3 Convolution

After mixing signal and noise, the result needs to be convoluted with a channel effect. To avoid any clipping effects, the file is normalized and converted to float before the convolution. The resulting signal is then converted back to integer before being saved. To speed up the convolution process, both files are translated into the frequency domain, where the time domain convolution becomes a simple multiplication. The multiplied signal is then converted back into the time domain and saved for the data extraction.

3.8.4 Data extraction

In case of this research the only input feature used are Mel Frequency Cepstral Coefficients (MFCCs), since they provide detailed information in a compact form, which is crucial for the smart phone transmission, and they give clear insight into the frequencies that also used by humans to detect emotion. Those MFCCs are retrieved in the following way [43]:

1. Cut the sound file into snippets of length 0.025s with a step between two windows of 0.01s, in order to create the appropriate overlap
2. Since the audio signal is supposed to be statistically stationary within the window, the snippet is converted in to the frequency domain using a discrete Fourier transformation
3. Calculate the logarithm of the amplitude spectrum in order to simulate the perceived loudness
4. Use the nonlinear Mel-scale filter bank in order to emphasize lower frequencies of the speech, which contain information more relevant to perception than higher frequencies

5. Properly spaced by the Mel-scale, the filter-bank averages frequency regions and thus indicates their specific energy, which results in a smoothed amplitude spectrum

6. In order to de-correlate the overlapping snippets, the discrete cosine transformation is computed

7. Finally the results are transferred back into the time domain in order to reverse the side-effects of the discrete cosine transformation

The extracted MFCCs still contain the c_0 coefficient, which represents the average energy of the snippet. Since this information has little emotion or speaker related content, it is excluded from the results. For this research the MFCC coefficients 1-36 were used.

3.8.5 Voice activity detection

Voice activity detection (VAD is a processing step that aims at differentiating not only the time of silence, but also time of pure noise, from those parts of the record that actually contain human speech. In this thesis the VAD was done using a binary classifier, instead of a signal analysis. For this binary classification training a voice activity annotated dataset is needed. For this purpose a Random Forest classifier with 20 trees was trained on the TIMIT Buckeye datasets MFCCs. Since each MFCC represent a snippet of speech, the annotations are averaged over that snippet accordingly. The resulting VAD predictions reached an accuracy of 94%.

3.8.6 Windowing

After retrieving the MFCCs, the small data snippets need to be rearranged into meaningful packages. In order to do that windows of 256 frames, around 2,5 seconds, are aggregated. Those windows are then tested on whether they contain more that 50% speech and if they do they are stored for further processing. A majority voting over all frames retrieves the emotional labels of each window.
In addition to the packaging the windows are used for a short-term normalization such as the Cepstral Mean and Variance Normalization [43]. This normalization step subtracts the mean of all feature vectors in a window and additionally divides the data by its standard deviation. This step results in a uniform zero mean and unit variance in each window. Alternatively a long-term normalization would do those steps before windowing by using the mean and standard deviation of the whole dataset. In case of this research CMNV was only used on the short-term windows.

3.8.7 Cross validation

For the results to be objectively evaluated a cross validation over different test and training sets is needed. This cross validation also has to fulfill some task specific requirements regarding the splitting of the data. In case of this research a four fold cross validation was used. This number originated from the number of different phones and positions for the channel effects. Since a cross validation over all possible combinations would take too much time, a diagonal approach is used. That way each phone and each position was part of the test set once. In addition to excluding the phone and position that is tested from the training set, the speakers need to be divided between training and test, in order to achieve a speaker independent emotion classification. The emotions contained in training and test were split together with their speakers and thus not balanced, such as is proposed in [19], but left in their original distribution. This is in agreement with the simulation of a real life situation that would not provide evenly distributed emotions for each speaker.

3.8.8 Test set creation

As a final step all test files are mixed and accumulated into one pseudo real dialog and then split into windows again. These results in a simulation of the emotional switches between speakers of a dialog and in windows of mixed emotions, that represents a real life transition during any dialog. Again the majority vote decides about the emotion label of each window.

4 Emotion recognition

In this chapter the different approaches and methods used for emotion classification are shortly described. For a more detailed view into the concepts, please refer to the previous work on this project in [42] and [4]

4.1 Universal Background Model

The concept of a universal background model (UBM) [30] is to represent the average speech of any speaker without any emotional coloring. The UBM can have two different functionalities. One way is, that it is used in emotion detection as the impostor model that is compared to the model of the emotion. In that case, the test data is scored against both models and the emotion counts as detected if the score of the emotion model is higher that the UBM score.
The other possibility is to use the UBM as a baseline from which other speaker or emotion specific models can be adapted. Additionally the UBM is used in this thesis to initialize the vector spaces needed for different models. This task could also be performed on the emotional data, but might then create only a limited space, that is not applicable to all emotional datasets. The UBM, on the other hand, ranges over a lot of different speech samples, thus it should provide a sufficient space model. In case of this thesis, the UBM consists of a Gaussian mixture model or in as an impostor in a form matching the model type it is compared to.

4.2 GMM emotion detection with MAP adaption

The Gaussian mixture model is one of the most used models for speaker recognition and emotion recognition. The GMM can either be calculated directly from the data of each emotion, or the UBM-GMM [30] can be transformed into an emotion specific model by a maximum a posteriori adaption (MAP adaption)(chapter 4.2.2). The MAP adaption step reforms the model according to the new input but also keeps the models basis. That way, a MAP adapted UBM is more general than the GMM from scratch and thus prevents over fitting.
The GMM is a probabilistic model that consists of multiple Gaussian probability

distributions. The components of a Gaussian Model are the means, weights and covariances of each mixture element. In case of this thesis 256 mixtures were computed for each GMM. Each of these mixtures defines one cluster in which data points are distributed more densely. The more components are used, the more specific data features can be represented. GMMs with a low number of components on the other hand can be used for more general classification. During the training of a GMM, the means, weights and covariances are optimized by an expectation maximization algorithm (chapter 4.2.1). Due to the complexity of this algorithm the model is often initialized with the faster k-means clustering before training. When testing, the log-likelihood of the test data fitting the GMM is computed and compared with the other log-likelihoods.

4.2.1 Expectation maximization

The Expectation Maximization algorithm loops through the following two steps, expectation and maximization:

- E-Step: Compute the log-likelihood of the training data with respect to the current model, also known as the posterior probabilities
- M-Step: Update the models parameters, mean, weights, and covariance, to maximize the expected log-likelihood calculated in the E-Step

Since the MFCCs are highly uncorrelated, a diagonal covariance matrix is usually sufficient. The algorithm stops either after a defined number of iterations or after the log-likelihood reaches a threshold.

4.2.2 Maximum A Posteriori Adaption

The MAP adaption [30] is a single EM step used with the UBM model as the prior and the training data of a specific emotion as the maximum likelihood solution. The following steps roughly calculate it:

1. Match the new data to the mixtures of the UBM
2. Get likelihood statistics of the new probabilistic alignment
3. Update the UBM parameters such as mean, weights and covariance

Furthermore the relevance factor r, decides how strongly the model should be adapted to the new data.

4.3 iVector based emotion recognition

The concept of iVectors was derived from the Joint Factor Analysis [9]. Instead of making a distinction between emotion and channel subspaces it was proposed to only extract the factors and use them in one single Total Variability (TV) space. The iVector classification for emotion recognition [47] is build upon the MAP adapted GMM models. It uses the given models and acquires statistics about its relation to the training data. These statistics are aggregated in a supevector [24]. These iVectors of 100 features each are then projected into a Total Variability space that has previously been initialized by the iVectors build from UBM data samples. After this projection the iVector is normalized and whitened. Optionally the normalization techniques LDA and WCCN were used in this thesis. Both need to have initial space definitions acquired by the UBM data samples. Then the iVectors are projected into those two spaces.
There are multiple options for creating the final WCCN iVector model for each emotion. Either all emotional data is directly used to acquire one emotional iVector, or the data is windowed and after all transformations the window WCCN vectors are averaged for each emotion. In case of this thesis, the windowed approach was used, since numerical instabilities occurred when only operating with 4 iVectors for the LDA and WCCN transformation. For testing, the test data is also transformed step by step into a WCCN iVector and then tested for the maximal cosine similarity to one of the emotion iVectors.

4.3.1 LDA

The LDA projection used on iVectors [9] tries to improve the discrimination between classes by maximizing the variance between classes and minimize the variance within classes. It does so by projecting onto a new vector basis, which is based on the eigenvectors of the highest eigenvalues, thus reducing the variance within the dimensions.

4.3.2 WCCN

The idea of using WCCN normalization [18] on iVectors [9] is to scale the iVector space by the inverse of its within-class covariance matrix. To do that it used the class information from the training data and finds orthonormal directions in that feature space. It then projects the iVectors into that newly created vector basis.

4.4 iVector score normalization

For scoring without normalization a cosine similarity [8] is used. Additionally the score can be normalized in different ways, in order to reduce channel and noise effects. In this experiments the s-, z-norm and the modified cosine kernel were used.

The z-norm [46], or zero normalization [32], compensates inter-speaker score variation. For estimating the normalization statistics, an impostor set (from the UBM) is scored against the target model and the mean and the standard deviation of the scores are computed:

1. $IM = impostorScoresOnTargetWCCN$
2. $score(testWCCN, targetWCCN) = \frac{Loglikelihood(testWCCN,targetWCCN) - \mu(IM)}{\sigma(IM)}$

The s-norm [23], or symmetric normalization, is reducing within session variability leading to improved performance, better calibration, and more reliable threshold setting. For estimating the normalization statistics, again an impostor set (from the UBM) is scored against the target model and the mean and the standard deviation of the scores are computed. Additionally, the same procedure is used for scoring the impostor against the testWCCN. The final normalized score is then calculated as follows:

1. $IT = impostorScoresOnTestWCCN$
2. $IM = impostorScoresOnTargetWCCN$
3. $score(testWCCN, targetWCCN) = \frac{Loglikelihood(testWCCN,targetWCCN) - \mu(IM)}{\sigma(IM)} + \frac{Loglikelihood(testWCCN,targetWCCN) - \mu(IT)}{\sigma(IT)}$

The modified cosine kernel norm [8] is a new cosine similarity scoring combining the effects of z- and t-norm. For doing that it is necessary to first compute the diagonal matrix that contains the square root of the diagonal covariance matrix of the impostor iVectors and the mean impostor iVector. Then the score can be computed as follows:

1. $norm\_speaker = norm(impostor\_sqrt\_diag\_covar\_mat, targetWCCN)$
2. $norm\_test = norm(impostor\_sqrt\_diag\_covar\_mat, testWCCN)$
3. $dist\_target = targetWCCN - mean\_impostor\_ivector$
4. $dist\_test = testWCCN - mean\_impostor\_ivector$
5. $score\_norm = (dist\_target * dist\_test) / (norm\_speaker * norm\_test)$

4.5 Anchor models

Anchor models [3] are an approach to improve the performance of another model. Both anchor models based on the MAP adapted GMMs and anchor models based on the WCCN projected iVectors were used. In an anchor model the performance of the training data for each emotion is scored (log-likelihood) on all models. This results in an Emotion Characterization Vector (ECV) that summarizes, how a certain emotional dataset behaves against all models. The resulting EVC is additionally projected into the WCCN space for both GMMs and iVectors. This procedure is used on each window and then averaged over all resulting EVCs of one emotion. For testing the test window is scored against all emotion models, resulting in a test EVC, which is then compared to the anchor models of each emotion. For the comparison the cosine similarity is used.

4.6 Random Forest

The Random Forest algorithm was included into the classification as a comparison between a standard general classification and the emotion recognition specific/ proven algorithms. A Random Forest ensembles multiple decision trees. It then classifies the test data with each of the decision trees in order to receive a majority vote as the predicted value. Random Forest can also be used for regression, as later explained in chapter 5.

4.7 Evaluation metrics

To evaluate the results of this classification approaches, three different metrics are used. All of these metrics can be deduced from the confusion matrix.

4.7.1 Confusion matrix

When testing the performance of a classification algorithm, one compares the actual class with the predicted class. A confusion matrix is a table that is always built with the scheme seen in table 4.1. The values in the cells will later be used to calculate the accuracy, recall and precision.

Table 4.1: Comparison of predicted classification and real classification in a confusion matrix

	Actual value positive	Actual value negative
Predicted value positive	true positives	false positives
Predicted value negative	false negatives	true negatives

4.7.2 Accuracy

Accuracy is the proportion of correct predictions compared to all predictions. In case of this thesis, accuracy and recall mostly have the same values.
Accuracy =(P(true positive) + P(true negative)) / (P(true positive) + P(false negative) + P(false positive) + P(true negative))

4.7.3 Recall

The recall metric tells us, how many of the predicted values for each class originated in that same class. In this thesis only the unweighted average recall (UAR) is used as a measurement, since the unbalance of the real life data should be preserved. UAR is also called the Balanced Error Rate.
Recall= P(true positive)/ P(true positive)+P(false negative)

4.7.4 Precision

The precision metric tells us, how many of the predictions for each class were classified as that class. The value is also called the positive predictive value (PPV), since it evaluates, how good the classifier can predict positive values:
Precision=P(true positive) / P(true positive) +P(false positive)

5 Affect regression

5.1 Random Forest Regressor

The Random Forest Regressor [25] is an ensemble algorithm. It consists of multiple individual regression trees. Each regression tree is tested separately for each test sample and the regression result is the aggregated over all trees.
In this thesis two configurations of Random Forest have been implemented. One has a maximal number of 10 trees and a maximal tree depth of 10, in order to see the performance within a minimal setting. The other Random Forest was given a set of 100 trees with a maximal depth of 20. Both versions were tested on the raw input data, on an average input sample over the elements in each window and on an iVector created with the samples from each window.

5.2 Support Vector Regressor

Support Vector Regression [38] works very similar to a classification with a Support Vector Machine. Instead of minimizing the distance of the target coordinated to the decision surface, a Support Vector Regressor optimizes a function to have a lower than ϵ deviation from the target points. At the same time it tries to maximize the flatness of the function, which means minimizing the norm of the functions weights. Given both conditions, the Support Vector Regressor seeks the solution for a convex optimization problem.
In this thesis two configurations of a RBF (radial basis function kernel) Support Vector Regressor have been implemented. One has an error penalty of 1.0 and only a maximum of 200 iterations. The other one has an error penalty of 10.0 and a maximum of 400 iterations. Both versions were tested on the raw input data, on an average input sample over the elements in each window and on an iVector created with the samples from each window.

5.3 Long short-term memory, Recurrent Neural Networks

Long short-term memory, Recurrent Neural Networks are a new form of recurrent neural networks preferably used to predict sequences. The conception of the LSTM used in this thesis is based on Graves [16] the implementation is done using Theano [5], [6] and is mostly based on an open Source LSTM implementation (`https://github.com/kastnerkyle/net/blob/master/net.py`).
The network is configured to use steepest gradient descent to optimize the network. It treats each input frame as a sequence with only one time step. The regression values are transformed into integers of a fixed interval and given as target variables. In the first layer, the input is read and interpreted. Then two hidden LSTM layers with 10 units each follow. The last layer consists of a softmax, which produces a probability distribution over all possible target values as an output.
When predicting the regression values, the target value with the highest probability is chosen as the resulting value and transformed back to its original value range between -1 and 1. Again two variants of models were calculated. Both have a learning rate of 0.1 and a momentum of 0.6. Since the direct processing of all the input is too time consuming, the variants differ in the size of the averaged windows. The first variant uses windows with 256 frames and the second variant only uses 128 frames, resulting in a more precise network.

5.4 Evaluation metrics

For the regression algorithms, there are two different evaluation methods. One is to use the direct regression value, which will be qualified by the explained variance, the mean absolute error and the R2 score. The other is to transform the values into a tenaer classification of the values -1,0 and 1 according to their meaning in the valence-arousal-scheme. The classification metrics will be accuracy, precision and recall, as defined in chapter 4.7. The regression metrics will be shortly explained here.

5.4.1 Explained Variance

The explained variance is a measure for the variance of prediction values that is explained by the model used to do the prediction. in contrast the unexplained variance would be the variance caused by external influences.

5.4.2 Mean absolute error

The mean absolute error takes the absolute value of difference between prediction and target and averages it over all samples.

5.4.3 R2 score

The R2 score also known as R-squared is a statistical measure of how close the target values are to the predicted regression line. It is also called the coefficient of determination. In case of a linear regression is can be seen as equivalent with the squared correlation coefficient.

6 Discussion of Results

6.1 Emotion Recognition - Comparable results of literature

First of all the best results achieved for emotion classifications in this thesis are compared with the results of literature. An overview of the best results reached in this work is given in table 6.1. For comparison the average results of a random forest classifier as a general classificator are given. It can easily be seen, that the results are quite low and the random forest outperforms all classification results of the SEMAINE dataset and all except the normal dataset of Berlin-EMO. Only the FAU dataset reached performance that outdid the random forest classification.

Table 6.1: Comparison of best results in emotion recognition specific classifications with the average Random Forest result

Audiotype	Berlin best	Fau best	Semaine best	Random Forest average
Normal	0,473	0,513	0,292	0,325
Noised	0,193	0,312	0,217	0,288
Channeled	0,178	0,325	0,239	0,267

For comparison with previous research results, Iliou and Anagnostopoulos [21], reached an average accuracy of 48% for speaker independent emotion recognition. This signifies, that even the Random forest Regression results reached and compared are lower that the state of the art. The reason for this difference even when comparing with the results of un-noised un-channeled data, is the lack of input features, which were reduced to only MFCC vectors.

Another way of comparing the results is by looking at the researches state on each algorithms performance. For example Attabi [3] reached 44.19 UAR with his newly introduced anchor model on the FAU Aibo dataset. With this he outperformed the best performance of the 2009 INTERSPEECH [40], where 44% were reached. In comparison this project only reached a maximum of 23.39% UAR when using anchor models.

Looking at the performances of the Berlin-EMO one can see an even stronger impact of the real life restrictions on the results. In the 2009 INTERSPEECH [40] a maximum

of 90% accuracy was reached for the Berlin Dataset, whereas only a maximum of 32.25% accuracy was reached here. This difference is especially prominent for the Berlin-EMO dataset, since it is quite small and can easily be optimized with enough input parameters. Other research reached the following results for the Berlin-EMO dataset using the GMM algorithm, on which a maximum of 32.25% accuracy was reached in this thesis (listed as cited by Anagnostopoulos [1]):

- 81% in Berlin EMO database (speaker independent) Atassi and Esposito (2008)
- 74.6% in Berlin EMO database (speaker independent) Lugger and Yang (2007a)
- 63% in Berlin EMO database Mishra and Sekhar (2009)

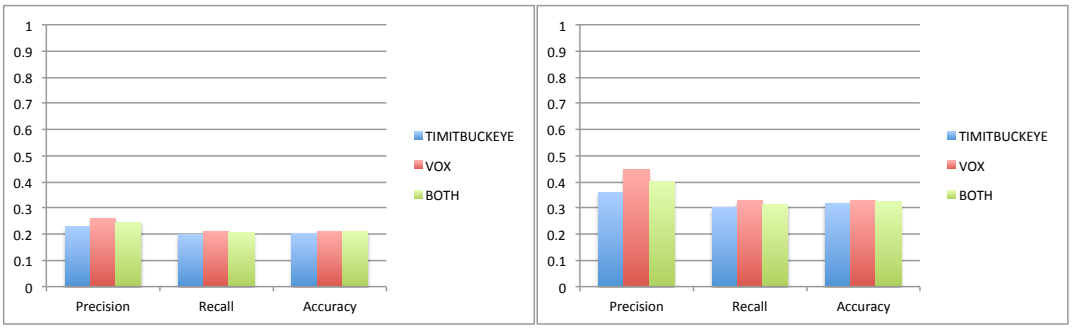
Even better results were reached when using SVMs, which were not part of this thesis and can thus not be compared (listed as cited by Anagnostopoulos [1]):

- 87.5% in Berlin EMO database Schuller et al. (2005a)
- ca. 90% in Berlin EMO database Vlasenko et al. (2007)
- 78% in Berlin EMO database Luengo et al. (2010)
- 88.6% in Berlin EMO database Wu et al. (2009)
- 89% in Berlin EMO Yang et al. (2009a)

More recent research of the team of Florian Eyben reached 84.6% with SVMs [41] 79.1% with Generalized Discriminant Analysis based on Deep Neural Networks, 88.8% UAR with Support-Vector Machines [11] and 73.2% with HMMs and GMMs [34].
None of those results were below 60%, which indicates a loss of 30% when using only MFCCs.
To rule out the influence of the dataset, also the FAU Aibo set, which performed best in all categories, should be compared to the state of the art. In the 2009 INTERSPEECH [40] a maximum of 69% UAR was reached for the 4-class problem of FAU Aibo. Compared to that only 51.3% UAR are reached with the best algorithm of this research. The difference is only 20% in this case, but still the heavy impact of limitations is visible.
The abundance of good emotion recognition results in all areas supposes that this is a solved problem. But all these results have in common, that they used clean data without noise or channel effects as well as a large number of spectral and prosodic features, while this thesis solely relied on MFCCS.

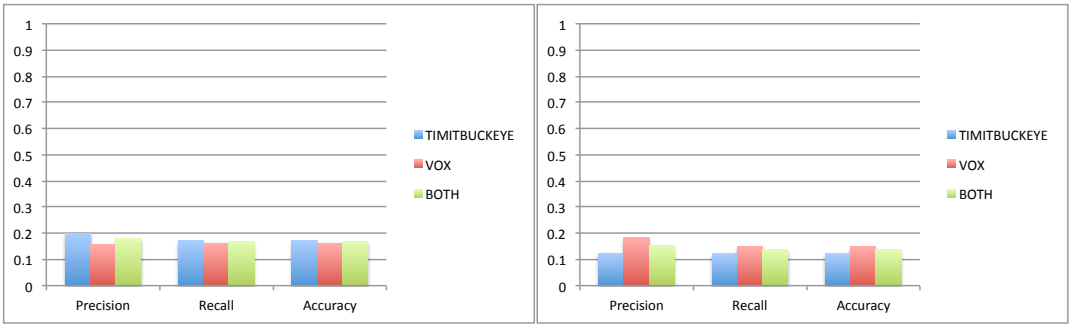
6.2 Emotion Recognition - Differences between UBMs

In figure 6.1, figure 6.2 and figure 6.3 the influence of using different data for the universal background model is shown with regard to each dataset. All algorithms are used with the help of the UBM, since the GMMs are MAP adapted based on it and the iVectors are projected from the emotion specific GMMs. Even the anchor models build on GMMs or iVectors giving them an indirect connection to the UBM.
All four comparisons show, that the UBM choice little if any impact. In general the VoxForge UBM shows better results than the TIMIT Buckeye UBM.



(a) Difference between universal background models averaged over all audio types of the Berlin dataset

(b) Difference between universal background models averaged over all normalized audio files of the Berlin dataset

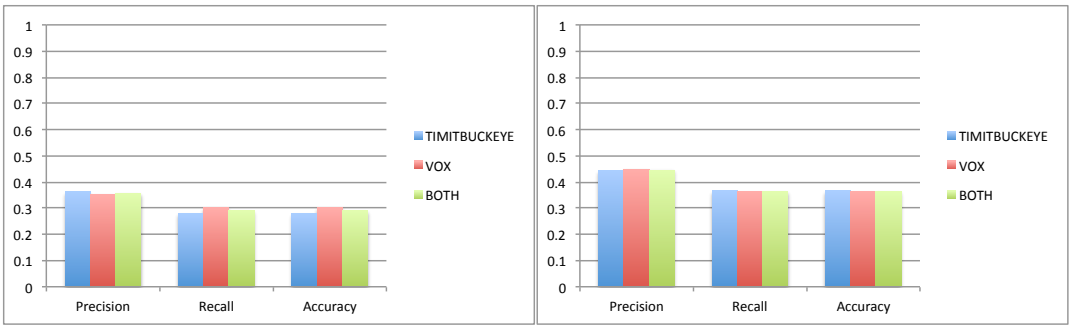


(c) Difference between universal background models averaged over all noised audio files of the Berlin dataset

(d) Difference between universal background models averaged over all channeled audio files of the Berlin dataset

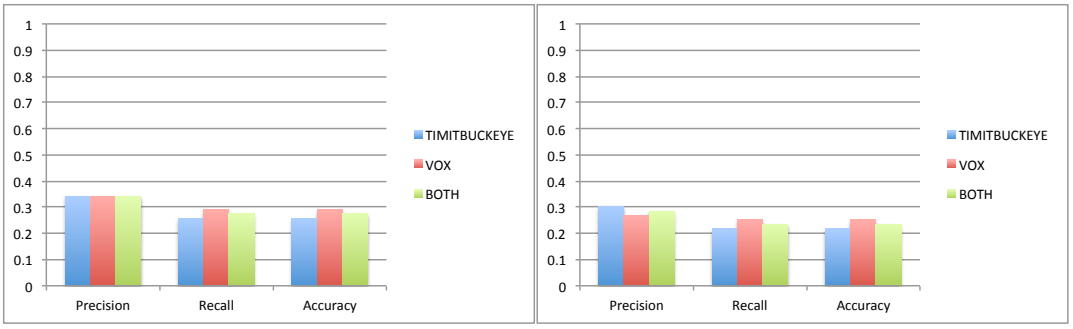
Figure 6.1: Illustration of differences between the universal background models for the Berlin dataset

For the Berlin-EMO dataset, only the noised audio files showed a better performance with the TIMIT Buckeye data. All other performances differ around 3% to the benefit of the VoxForge UBM. This might be a slight indication that the TIMIT Buckeye dataset is more noised than the VoxForge. Which would have to be confirmed by the other two datasets.



(a) Difference between universal background models averaged over all audio types of the FAU dataset

(b) Difference between universal background models averaged over all normalized audio files of the FAU dataset



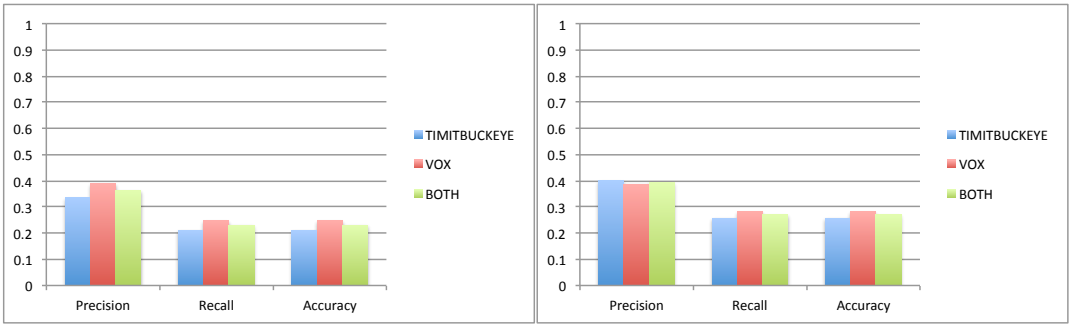
(c) Difference between universal background models averaged over all noised audio files of the FAU dataset

(d) Difference between universal background models averaged over all channeled audio files of the FAU dataset

Figure 6.2: Illustration of differences between the universal background models for the FAU dataset

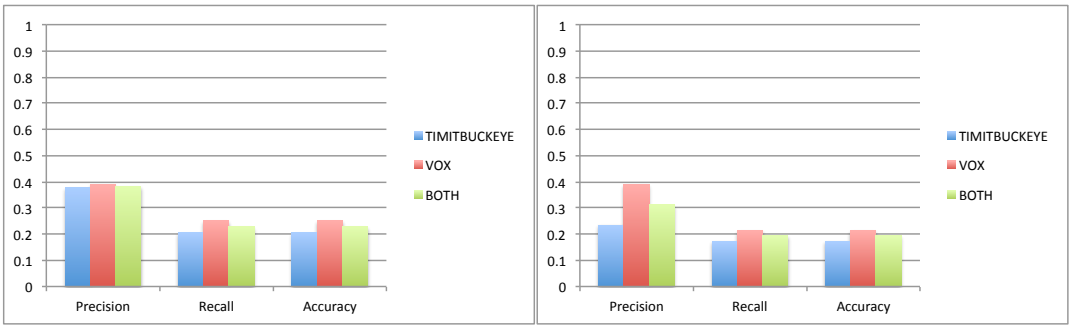
When looking at the FAU Aibo dataset (figure 6.2) the noised data shows a clear indication for a better VoxForge UBM. Also all other audio types show that VoxForge provides the same or better results. This supposes the switched relation observed in the Berlin dataset was an outlier.
Overall the FAU Aibo dataset is less influenced by the choice of the UBM. In case of

the normal dataset the performances are even identical. This observation shows, that FAU Aibo not only provides the best results, but also the most robust results.



(a) Difference between universal background models averaged over all audio types of the SEMAINE dataset

(b) Difference between universal background models averaged over all normalized audio files of the SEMAINE dataset



(c) Difference between universal background models averaged over all noised audio files of the SEMAINE dataset

(d) Difference between universal background models averaged over all channeled audio files of the SEMAINE dataset

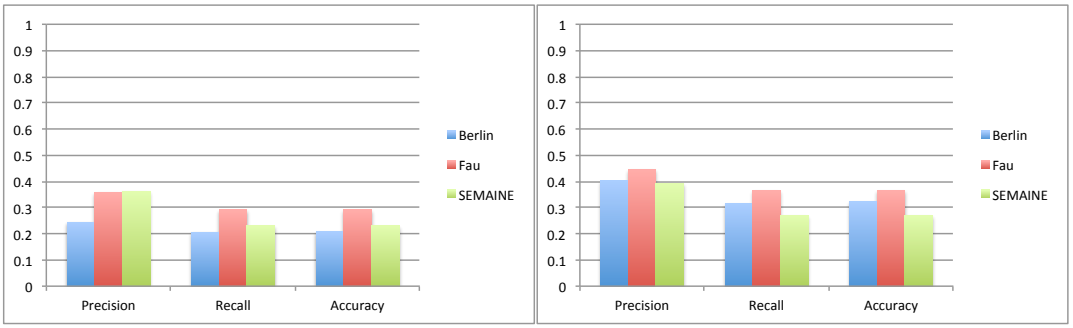
Figure 6.3: Illustration of differences between the universal background models for the SEMAINE dataset

In the SEMAINE dataset (figure 6.3) the performance of VoxForge is also better in all cases. No further outliers or special correlations are visible for this dataset.

6.3 Emotion Recognition - Differences between datasets

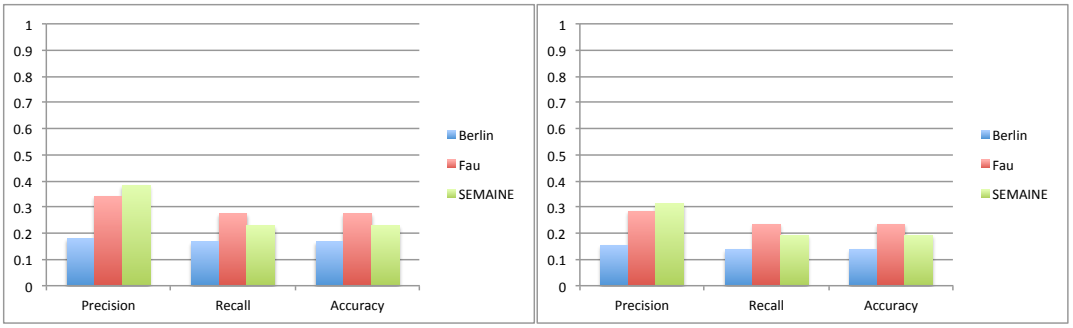
Figure: 6.4 shows a comparison of how the datasets performed compared to each other. The charts immediately support the previous notion of FAU outperforming all the

other datasets. This is consistent with FAU having the largest available dataset and thus the best basis for the building models
Also Berlin-EMO only reaches a good performance when using the undisturbed data. Here its benefit of enacted and very clear emotions is strong, while in the disturbed data the disadvantage of having only a small number of samples overweighs. In contrast SEMAINE is even less disturbed by noise and channel effects that the FAU dataset, thus slowly minimizes the difference to FAUs performance. Except for the special performance of Berlin on the undisturbed data, SEMAINE holds the second place in terms of the accuracy and recall.



(a) Difference between datasets averaged over all audio types

(b) Difference between datasets averaged over all results of normalized audio files



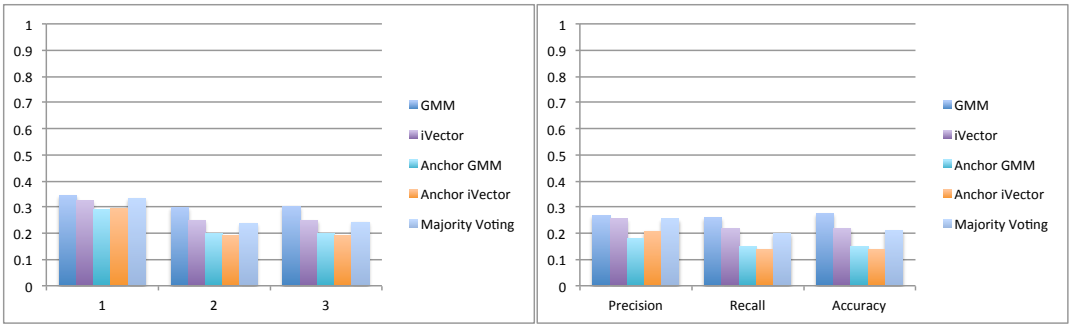
(c) Difference between datasets averaged over all results of noised audio files

(d) Difference between datasets averaged over all results of channeled audio files

Figure 6.4: Illustration of differences between the benchmark datasets for each audio type

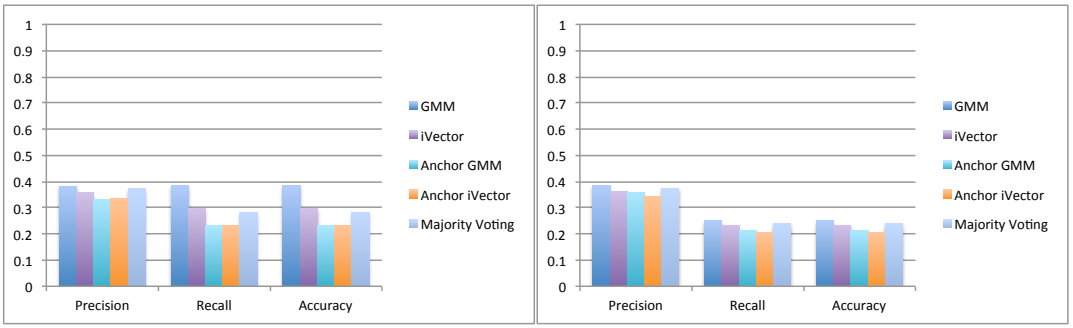
6.4 Emotion Recognition - Differences between algorithms

In figure 6.5 and figure 6.6 the performances of different classification algorithms are compared. It is obvious, that the GMM algorithm outperforms the other algorithms in all settings.
When looking at the differences between datasets in figure 6.5 , it can be observed, that all three datasets show the same ranking of algorithms. SEMAINE is the only dataset showing only little influence between the chosen approaches. This behavioral difference can be explained by the general low performance of all algorithms on this dataset.



(a) Difference between algorithms averaged over all datasets

(b) Difference between algorithms averaged over the Berlin dataset



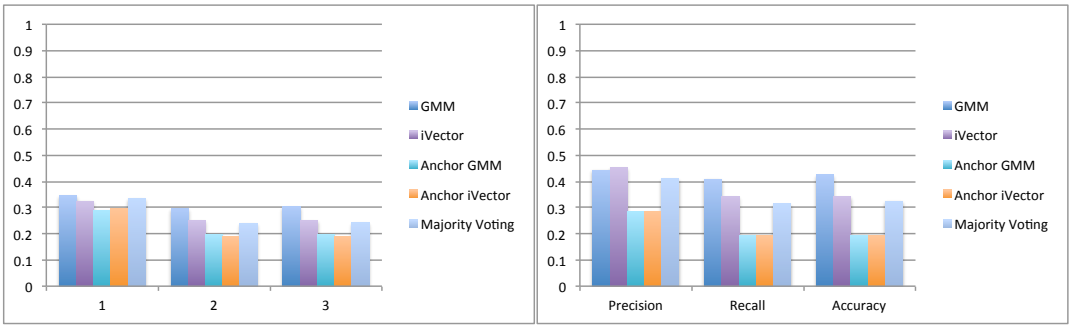
(c) Difference between algorithms averaged over the FAU dataset

(d) Difference between algorithms averaged over the SEMAINE dataset

Figure 6.5: Illustration of differences between algorithms in each benchmark dataset

When looking at the influence of disturbances in figure 6.6 the iVector algorithms are second place for all configurations except for the channeled audio files. It seems that the disturbances have a very strong effect on this algorithm. This is somewhat surprising

because they contain channel compensation norms, LDA and WCCN transformation, which should make them more robust to channel effects than most other algorithms. The Anchor models are always below the other algorithms except again in the channeled audio case, where they even outperform the iVector algorithm. In general the differences become smaller when adding more disturbances, like noise and channel effects. They seem to effect especially on the GMM and iVector performance, while the anchor models stay almost the same in all configurations. This could indicate, that although having a lower performance, the anchor version of both algorithms is more robust against outside disturbances.



(a) Difference between algorithms averaged over all audio types

(b) Difference between algorithms averaged over all results of normalized audio files

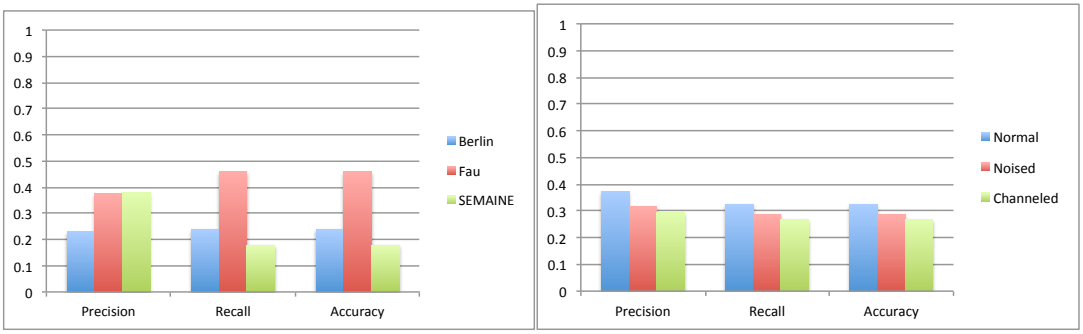
(c) Difference between algorithms averaged over all results of noised audio files

(d) Difference between algorithms averaged over all results of channeled audio files

Figure 6.6: Illustration of differences between algorithms for each audio type

In addition to the emotion recognition algorithms a Random Forest classification is included in these results to show the correlation to a general approach. The best results over the above shown algorithms are reached by the FAU dataset with a GMM

algorithm. This combination reaches an accuracy of 45.64%. Compared to this the random forest algorithm reached 46.06% on the same dataset. It also shows a much more distinct difference between the FAU results and the other datasets, as depicted in figure 6.7a. Considering this, the Random Forest might be only a favorable algorithm for the FAU Aibo dataset, leading to a better performance than the GMM on that specific set.



(a) Results of Random Forest classification for each dataset

(b) Results of Random Forest classification for each audio type

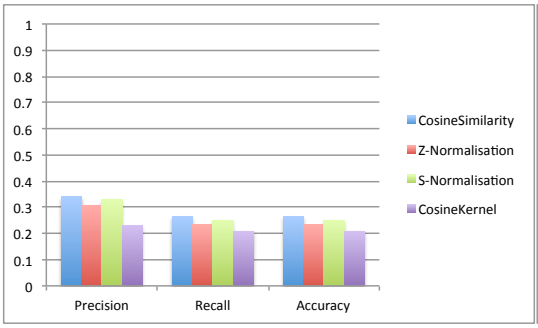
Figure 6.7: Illustration of random forest results as a general classifier compared to state of the art emotion classifiers

On the comparison between audio types (figure 6.7b) the disturbances have only little effect of around 5% while those same disturbances cause around 20% loss for the GMM and iVector algorithm results. This indicates a relatively low robustness for those two best algorithms, as already observed in the algorithm comparison 6.6 with regards to the iVector performance.

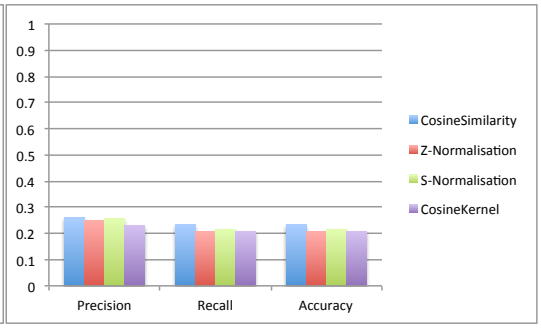
6.5 Emotion Recognition - Differences between channel compensating norms

In figure 6.8 and figure 6.9 the different channel compensating norms are shown for each dataset and for each level of disturbances.
In each of the datasets (figure 6.8) the cosine similarity outperforms the other three norms, but only by around 5% percent. The second best norm is the s-norm, without regard to the dataset. The only difference from the dataset perspective is the FAU dataset, which has a very stable distribution over all four norms. This seems to confirm

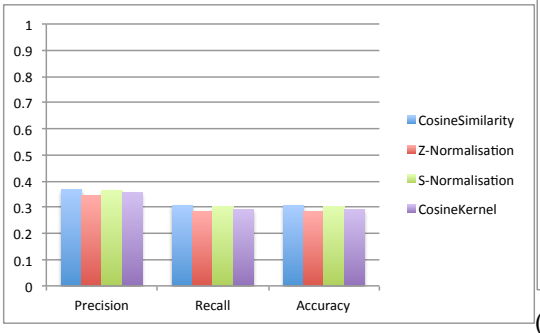
with the previously observed stability against UBM types, which both indicate the FAU dataset has not only higher performance but also high robustness against some external influences.



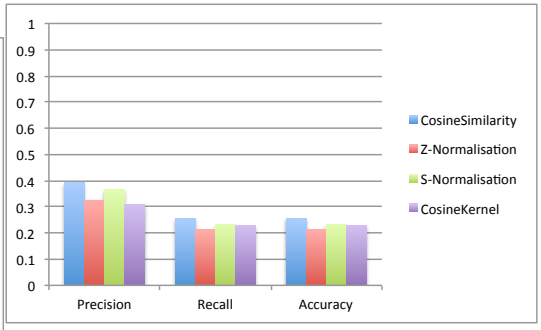
(a) Difference between channel compensating norms averaged over all datasets



(b) Difference between channel compensating norms averaged over the Berlin dataset



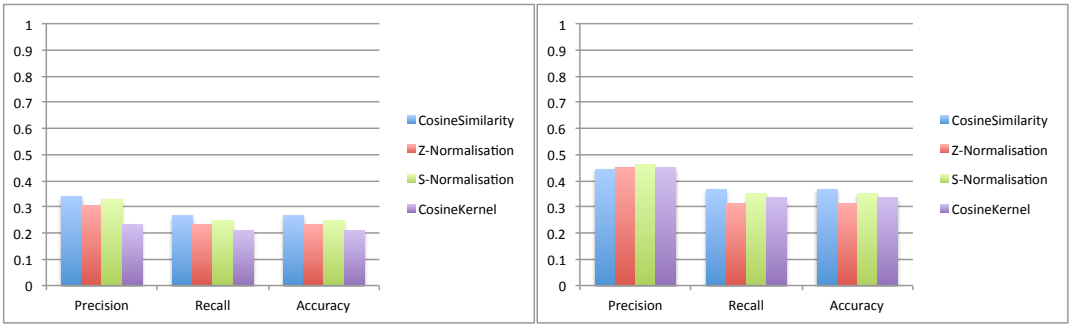
(c) Difference between channel compensating norms averaged over the FAU dataset



(d) Difference between channel compensating norms averaged over the SEMAINE dataset

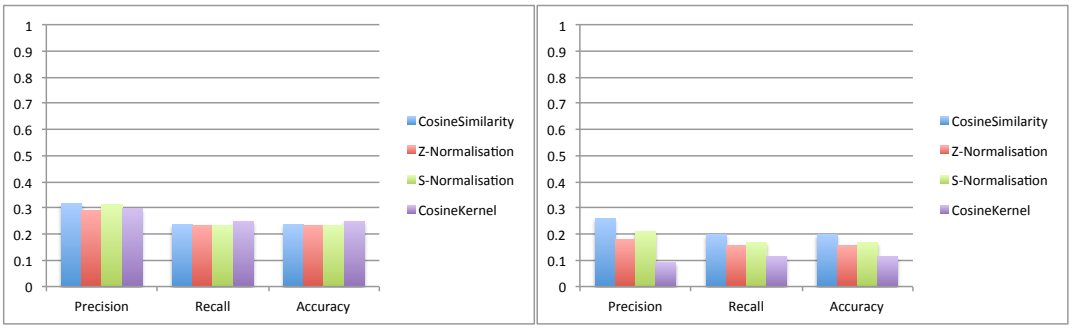
Figure 6.8: Illustration of differences between channel compensating norms in each benchmark dataset

When comparing the different audio types (figure 6.9) the cosine similarity and s-norm keep their positions for the normal and channeled files. For the noised case on the other hand the cosine kernel as defined by Dehak [8] copes better with the external influences than all other norms. It is curious, that the norm seems to handle noise, but not channel disturbances. This difference might be part of future experiments.



(a) Difference between channel compensating norms averaged over all audio types

(b) Difference between channel compensating norms averaged over all normalized audio files



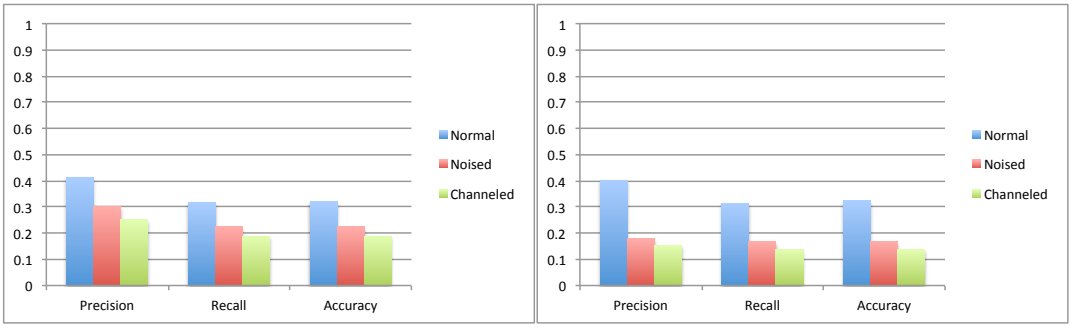
(c) Difference between channel compensating norms averaged over all noised audio files

(d) Difference between channel compensating norms averaged over all channeled audio files

Figure 6.9: Illustration of differences between channel compensating norms for each audio type

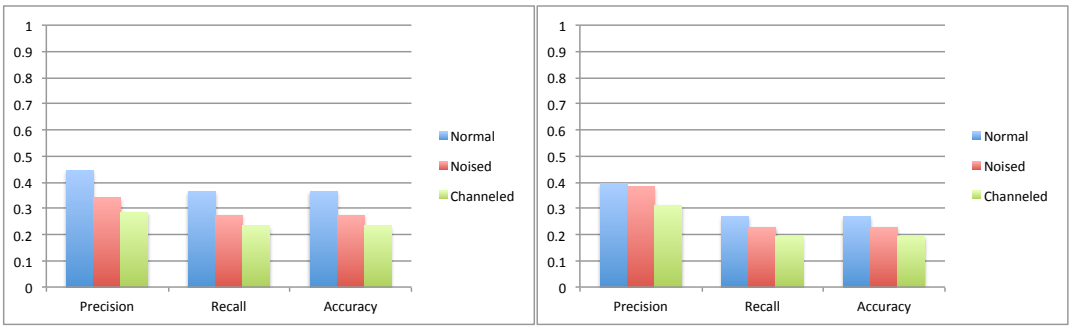
6.6 Emotion Recognition - Influence of noise and channel effects

Figure 6.10 and figure 6.11 show how strongly the external disturbances influence the outcome of the classification results. Along the datasets (figure 6.10) is can be observed, that the Berlin dataset is much less robust against noise and channel effects than the other two datasets. It can also be notices, that the noise influence reduces the results for FAU and Berlin stronger, than the channel effect, while for SEMAINE both influences have a very similar impact.



(a) Influence of noise and channel effects averaged over all datasets

(b) Influence of noise and channel effects averaged over all the Berlin dataset

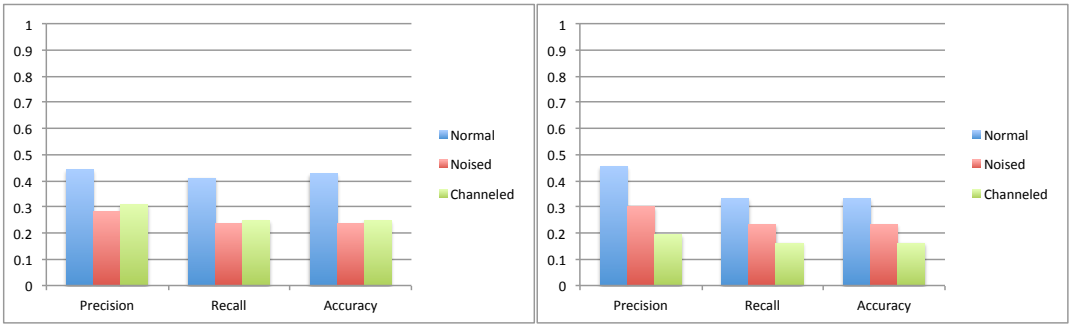


(c) Influence of noise and channel effects averaged over all the FAU dataset

(d) Influence of noise and channel effects averaged over all the SEMAINE dataset

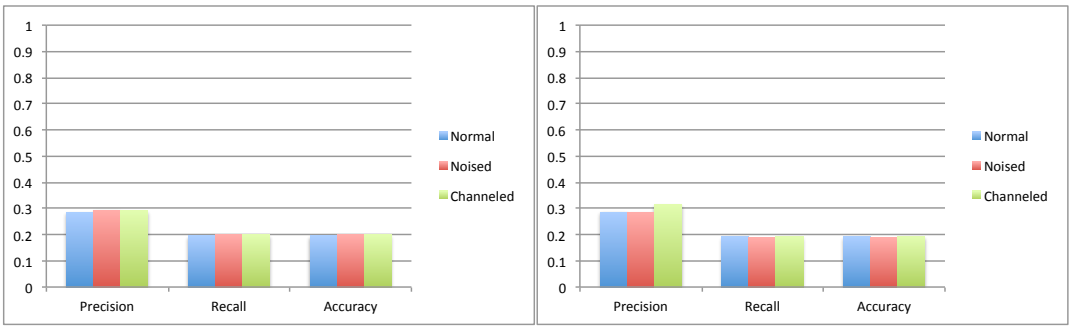
Figure 6.10: Illustration of the influence of noise and channel effects in each benchmark dataset

When looking at the impact along the different algorithms (figure 6.11), it is again confirmed, that both anchor models are more robust against the disturbances than the GMM and iVector models. Another surprising effect is, that the channeled files perform slightly better than the only noised files for the GMM while the iVectors, which are supposed to cancel out channel effects by normalization, receive another fallback when adding channels. As these distinction between noise and channel effects also occurred earlier in the results comparisons, it stands to reason, that the noise actually has a stronger impact on the results than the channel effects. A future experiment comparing noised data with un-noised but channeled data could confirm that indication.



(a) Influence of noise and channel effects on the results of a GMM algorithm

(b) Influence of noise and channel effects on the results of an iVector algorithm



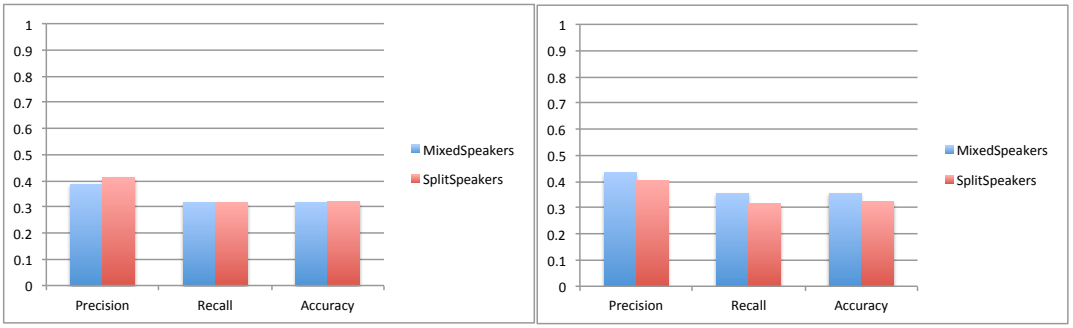
(c) Influence of noise and channel effects on the results of a GMM Anchor algorithm

(d) Influence of noise and channel effects on the results of an iVector Anchor algorithm

Figure 6.11: Illustration of the influence of noise and channel effects in each algorithm

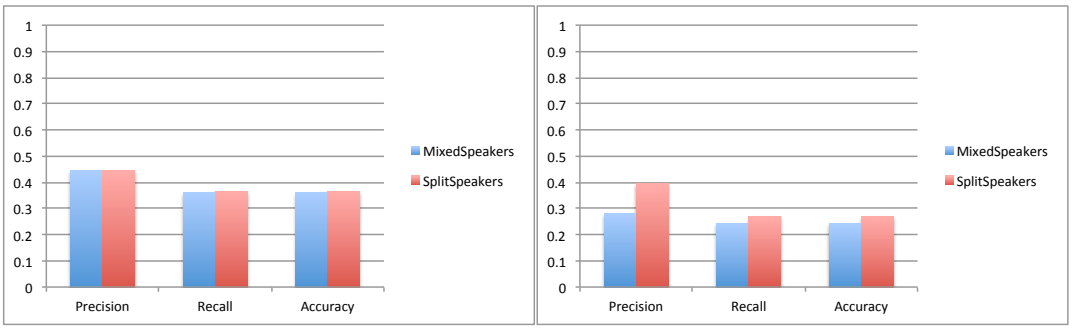
6.7 Emotion Recognition - Influence of speaker independence

Figure 6.12 and figure 6.14 present the influence of speaker independence on the classification results. Although previous research ([21], [26]) show a clear improvement of results for speaker dependent classification, the results only reflect this effect for the Berlin dataset.



(a) Influence of speaker independence averaged over all datasets

(b) Influence of speaker independence averaged over all the Berlin dataset



(c) Influence of speaker independence averaged over all the FAU dataset

(d) Influence of speaker independence averaged over all the SEMAINE dataset

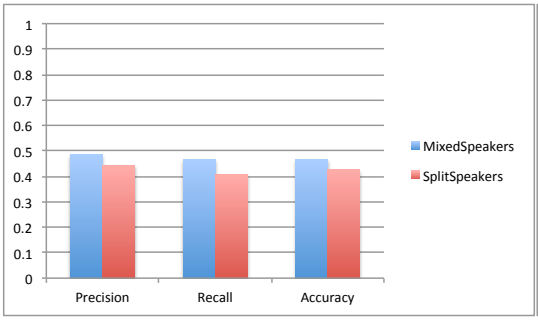
Figure 6.12: Illustration of the influence of speaker independence in each benchmark dataset

For Iliou the better performance might be explained, since they are also classifying the Berlin dataset, but Yang Liu is using the USC-IEMOCAP database and still has a positive influence of speaker dependence as shown in figure 6.13. In conclusion this supposes, that speaker dependency should have a positive effect, but does so only for certain datasets and algorithms. Whether that anomaly is caused by the real life limitations or by other side effects cannot be determined at this stage.

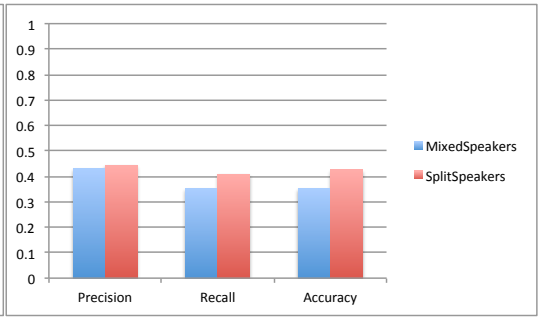
		Static
Speaker dependent		65.5
Speaker independent [Controlled size: same as speaker dependent]	Train spkrs K=2	49.1
	K=4	50.5
	K=6	53.2
	K=8	52.8
Speaker independent	All data	57.4

Figure 6.13: Illustration of the influence of speaker independence according to Yang Liu [26]

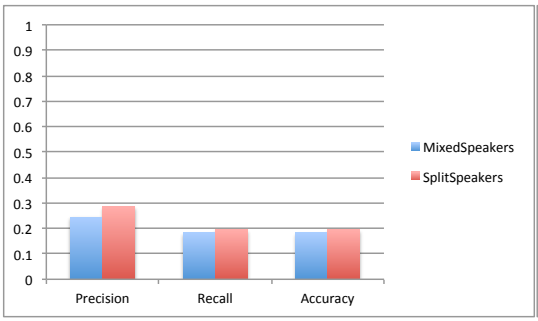
While the speaker dependence has almost no influence on the FAU dataset, the difference is clear on the SEMAINE dataset. For FAU the indifference to speaker dependence might be explained by the usage of children voices, which are not yet that distinctive. For SEMAINE the impact might be caused by the dataset only consisting of only four speakers overall, and when splitting them along the speakers instead of along the emotions, some emotions might not be trained well enough.



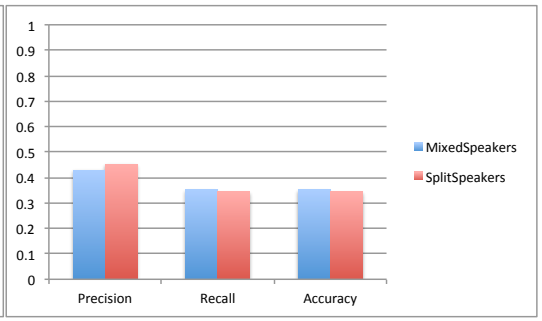
(a) Influence of speaker independence on the results of a GMM algorithm



(b) Influence of speaker independence on the results of an iVector algorithm



(c) Influence of speaker independence on the results of a GMM Anchor algorithm



(d) Influence of speaker independence on the results of an iVector Anchor algorithm

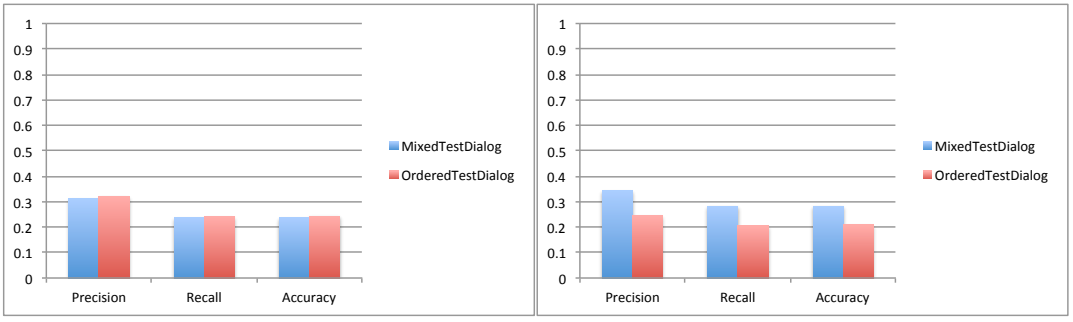
Figure 6.14: Illustration of the influence of speaker independence in each algorithm

Figure 6.14 shows that for GMM and iVector anchors the speaker dependence improves the results, while for pure iVectors or GMM anchors it worsens them. This unreliable behavior presents the question of whether there is a relation at all or only different algorithmic reactions to other effects.

6.8 Emotion Recognition - Influence of mixed test settings

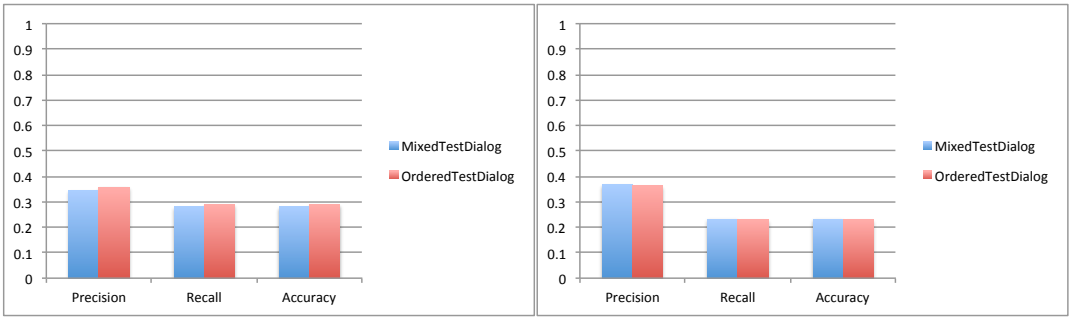
Figure 6.15 and figure 6.16 show how simulating a mixed dialog during the test influences the results of the classification.
In the first figure (6.15) the influence on different datasets is shown. Only the Berlin dataset seems to improve with a mixed dialog while all others are mostly stable, but slightly lower than with an ordered test dataset. As the Berlin dataset is enacted and thus clearly distinctive between the emotions, a mixing might not affect this set as

much, explaining the better performance. The results of the other algorithms are in line with the expected drawback, but since they shown only little influence they confirm, that this real life component is not part of the reason for low performances.



(a) Influence of mixed test settings averaged over all datasets

(b) Influence of mixed test settings averaged over all the Berlin dataset

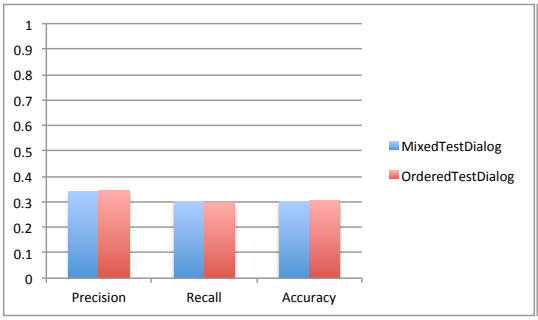


(c) Influence of mixed test settings averaged over all the FAU dataset

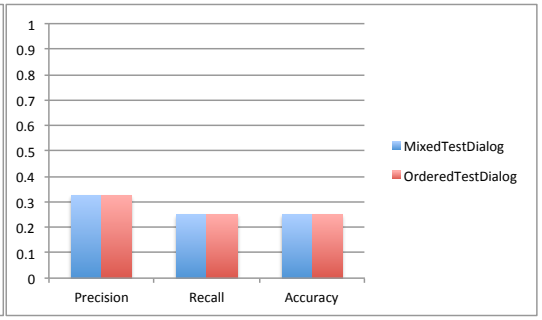
(d) Influence of mixed test settings averaged over all the SEMAINE dataset

Figure 6.15: Illustration of the influence of mixed test settings in each benchmark dataset

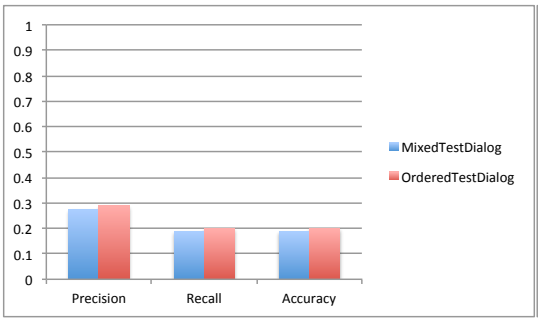
Figure 6.16 shows that the mixed test set has almost no influence on the algorithms performance. When influencing, the results are again negatively influenced. This confirms, that handling mixed windows of different speakers/ emotions is an influence but does not need to be considered a major disturbance.



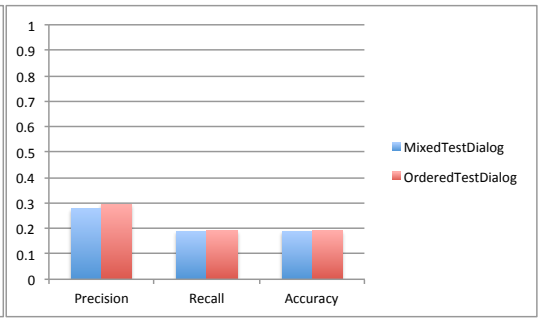
(a) Influence of mixed test settings on the results of a GMM algorithm



(b) Influence of mixed test settings on the results of an iVector algorithm



(c) Influence of mixed test settings on the results of a GMM Anchor algorithm



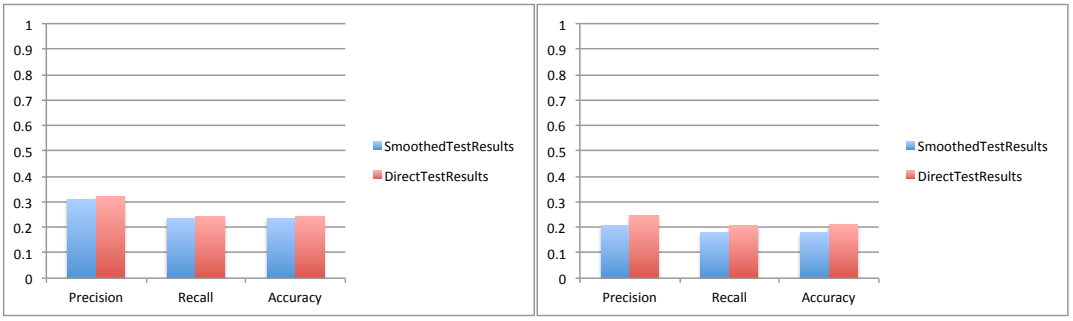
(d) Influence of mixed test settings on the results of an iVector Anchor algorithm

Figure 6.16: Illustration of the influence of mixed test settings in each algorithm

6.9 Emotion Recognition - Influence of smoothing the test results

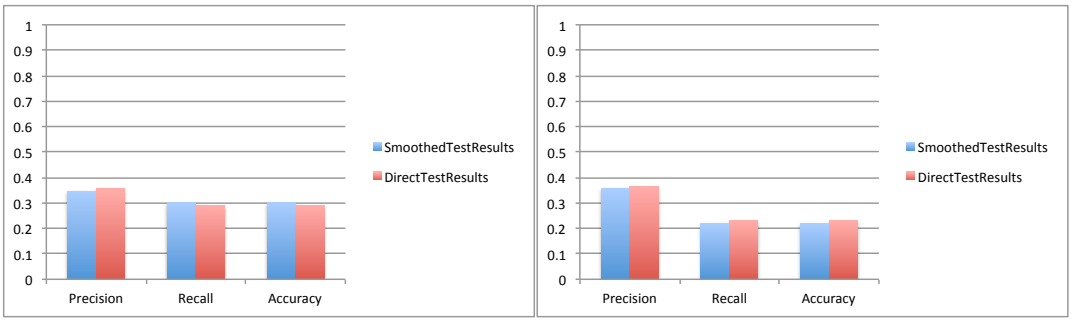
Figure 6.17 and figure 6.18 show how smoothing the test results even with a small span of 2 frames influences the quality of predictions.

.



(a) Influence of smoothing the test results averaged over all datasets

(b) Influence of smoothing the test results averaged over all the Berlin dataset

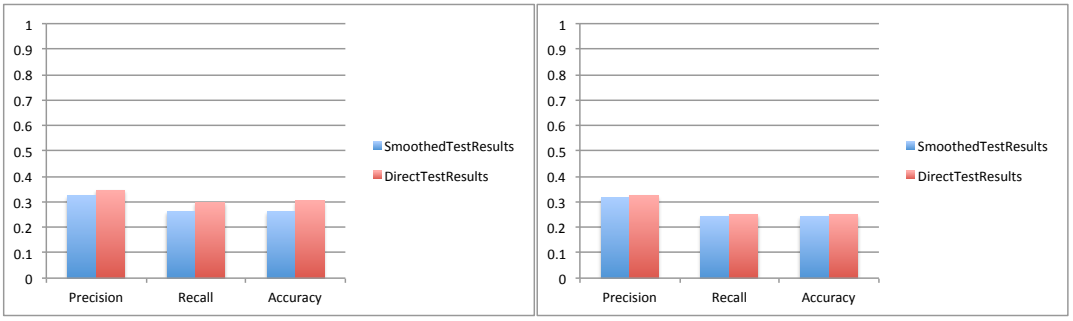


(c) Influence of smoothing the test results averaged over all the FAU dataset

(d) Influence of smoothing the test results averaged over all the SEMAINE dataset

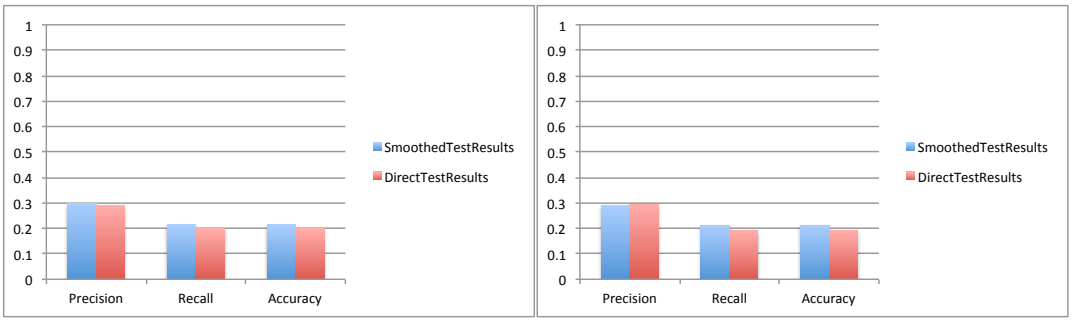
Figure 6.17: Illustration of the influence of smoothing the test results in each benchmark dataset

Along the sets (6.17) as well as along the algorithms (6.18) the smoothing has a negative effect. A look at the real and predicted data confirmed this correlation, since after the windowing of 256 frames two consecutive windows only rarely have similar/ the same emotion value. Thus smoothing any two windows would always result in less performance.



(a) Influence of smoothing the test results on the results of a GMM algorithm

(b) Influence of smoothing the test results on the results of an iVector algorithm



(c) Influence of smoothing the test results on the results of a GMM Anchor algorithm

(d) Influence of smoothing the test results on the results of an iVector Anchor algorithm

Figure 6.18: Illustration of the influence of smoothing the test results in each algorithm

6.10 Affect Regression - Comparable results of literature

For affect regression we will compare Random Forest and Support Vector regressors as well as LSTM RNNs. Since the LSTM and SVR solution has a recent application to the SEMAINE dataset by Eyben and Schuller [10], we compared those values in table 6.2.

Table 6.2: Comparison of R2 values with CC results of Eyben [10]

Dim	LSTM(CC)	SVR(CC)	LSTM S	LSTM L	SVR S	SVR L	RFR S	RFR L
V	0.454	-0.085	-1406,489	-128,041	-0,987	-0,910	-1,675	-1,604
A	0.757	0.653	-659,802	-122,195	-0,667	-0,695	-0,746	-0,653
E	0.549	0.190	-611,407	-51,367	-3,699	-0,986	-0,322	-2,841
P	0.520	0.367	-561,052	-380,074	-1,982	-1,772	-0,787	-1,367
I	0.579	0.503	-623,133	-97,977	-0,411	-0,380	-0,146	-0,194

The table clearly shows, that the regression results received from the channeled and limited regression are far below the ones in the paper. Additionally the paper [10] indicated that LSTM RNNs perform significantly better than the Support vector regressor, while in this thesis the Support Vector regressor, even with low performance is clearly ahead of the LSTM. The best performance of this research was reached with a Random forest regressor, which unfortunately was not part of the papers research. Differences in the LSTM that would explain the high variation of results are that the paper uses the sequential minimal optimization (SMO) algorithm while this thesis uses the steepest gradient descent (SGD) algorithm. Another strong parameter is again the number of features used (Intensity, Loudness, RMS & LOG energy F0, probability of voicing, MFCC 0–12, RASTA-PLP 0–7, log. Mel-Freq. bands 1–14, 95% spectral roll-off point, Spectral flux, entropy, and variance Zero-crossing rate) compared to the MFCCs 1-36 in this work including the effect of noising and channeling the data.

In addition to the direct regression the results were transformed into a tenaer classification, by mapping onto -1,0 and 1. A similar approach was conducted by Eyben [33] when transforming the regression values to a binary classification values indicating whether the original values were above or below the mean. The results of this binary/ tenaer classification are compared in table6.3.

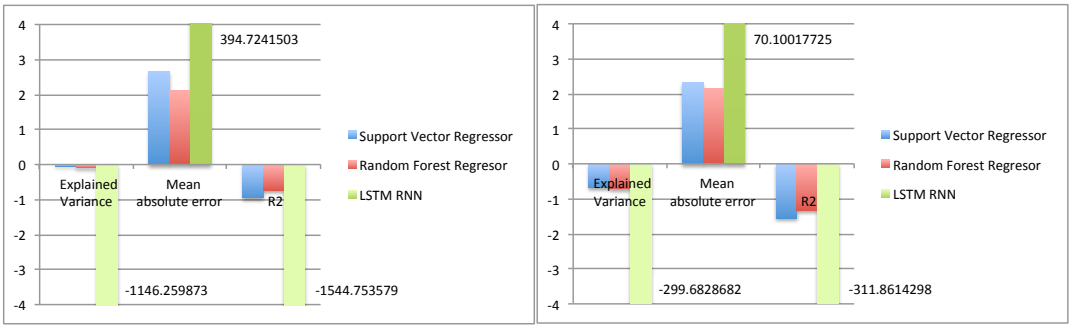
Table 6.3: Comparison of tenaer accuracy with binary accuracy of Eyben [33]

	Activity	Expectation	Power	Valence
Eyben	57.0	54.5	49.1	47.2
Thesis	56.0	53.0	54.0	52.8

The table shows that the results are very similar and the thesis Random Forest regressor even outperforms on the Power and Valence dimension. Showing that this tenaer regression combination is a successful step within this project.

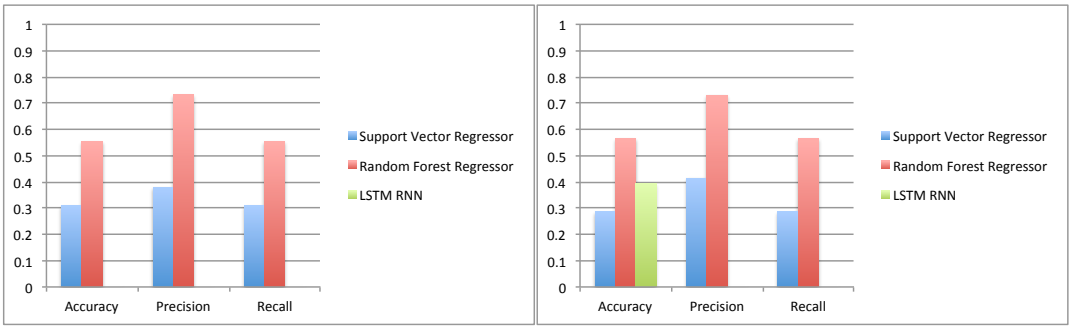
6.11 Affect Regression - Differences between algorithms

Figure 6.19 shows the different performances of each algorithm. It is clearly visible, that the LSTM performance is far worse than that of the other two regressors. Even in tenaer classification the LSTM results are mostly zero. Of the other two algorithms the Random Forest shows higher accuracies and better correlation metrics than the Support Vector regressor. The quality of the models showed almost no influence on the results, except for the LSTM tenaer regression to reach 40% instead if 0% accuracy.



(a) Difference between the regression results of different algorithms of low quality

(b) Difference between the regression results of different algorithms of high quality



(c) Difference between the tenaer results of different algorithms of low quality

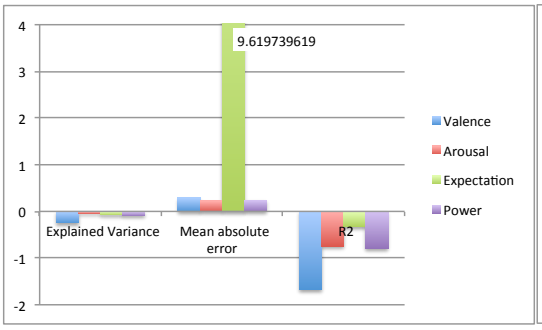
(d) Difference between the tenaer results of different algorithms of high quality

Figure 6.19: Illustration of the difference between the results in different algorithms

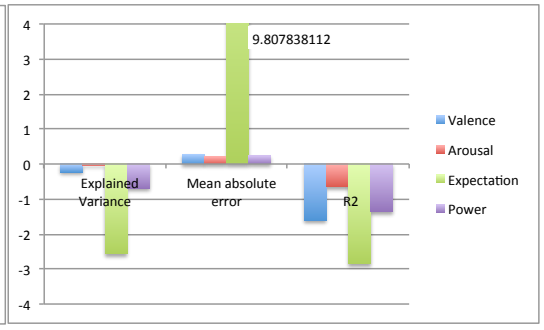
6.12 Affect Regression - Differences between dimensions

Figure 6.20 shows how the different affect dimensions react to the regression on small and large random forests. Expectation and Valence show clearly lower results on the

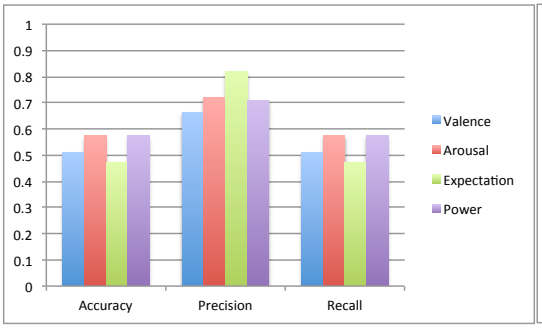
direct regression, but only slightly differ as soon as the tenaer postprocessing is used. For larger models, the tenaer transformation evens out the differences even more than for the smaller models.



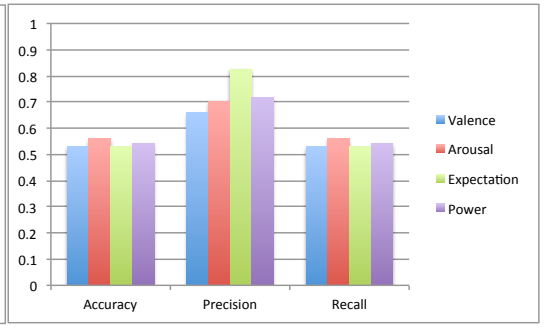
(a) Difference between the regression results of different dimensions in Random Forest Regression with 10 trees and a maximal depth of 10



(b) Difference between the regression results of different dimensions in Random Forest Regression with 100 trees and a maximal depth of 20



(c) Difference between the tenaer results of different dimensions in Random Forest Regression with 10 trees and a maximal depth of 10



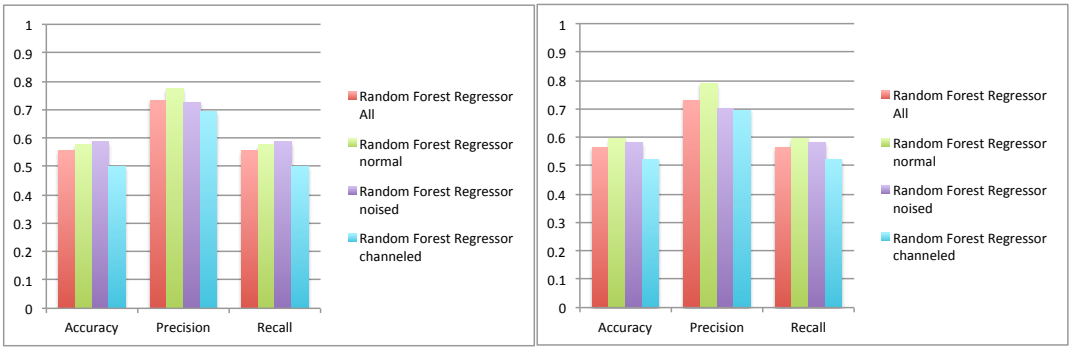
(d) Difference between the tenaer results of different dimensions in Random Forest Regression with 100 trees and a maximal depth of 20

Figure 6.20: Illustration of the difference between the results in different dimensions

6.13 Affect Regression - Influence of audio types

Figure 6.21 shows how the noise and channel effects influence the tenaer regression result. For the smaller model somehow the addition of noise even improves the results, while for the larger model we see the well-known behavior of stepwise decrease. The

glitch in the smaller model shows, how the results are not reliable enough so that the noise effects by coincidence produce better results than without noise.



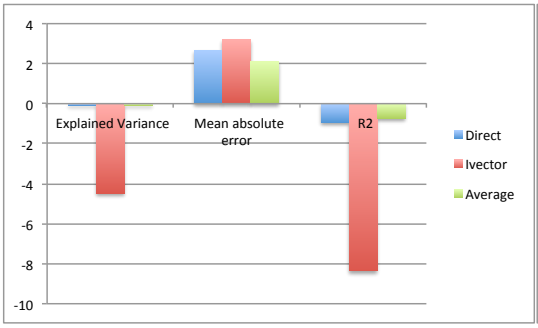
(a) Illustration of the influence of different audio types on the Random Forest Regression with 10 trees and a maximal depth of 10

(b) Illustration of the influence of different audio types on the Random Forest Regression with 100 trees and a maximal depth of 20

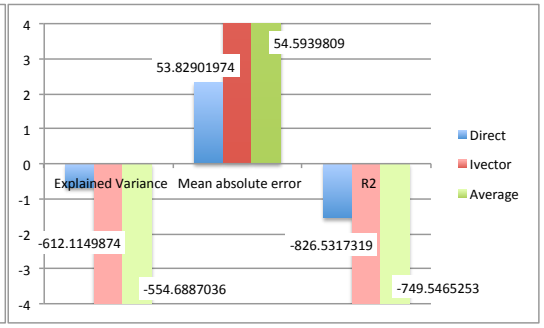
Figure 6.21: Illustration of the influence of different audio types on the Random Forest Regression results

6.14 Affect Regression - Influence of preprocessing steps

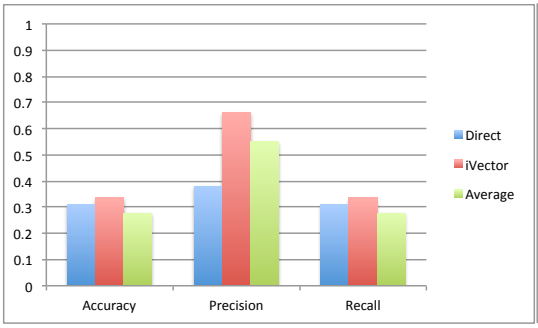
Figure 6.22 and figure 6.23 show the influence of preprocessing steps onto the Regression and tenaer classification results. For the SVR algorithm (figure 6.22) the iVector aggregation is clearly the worst option for the regression results, but it shows best results for the tenaer classification on both small and large models. In contrast the best classification results with direct values receive the lowest accuracies for the tenaer classification. It thus seems reasonable to conclude that this model is still very unstable even with the better parameters.



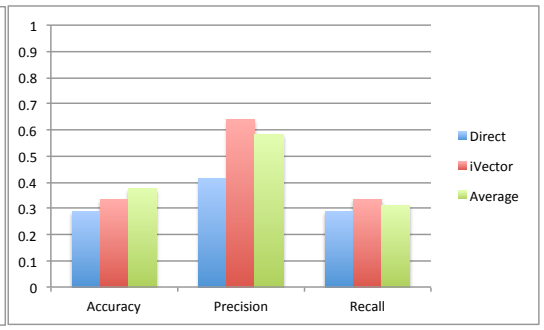
(a) Influence of smoothing on the regression results in Support Vector Regression a C-Penalty of 1, a maximum of 200 iterations and 256 frames per aggregation



(b) Influence of smoothing on the regression results in Support Vector Regression a C-Penalty of 10, a maximum of 400 iterations and 1024 frames per aggregation



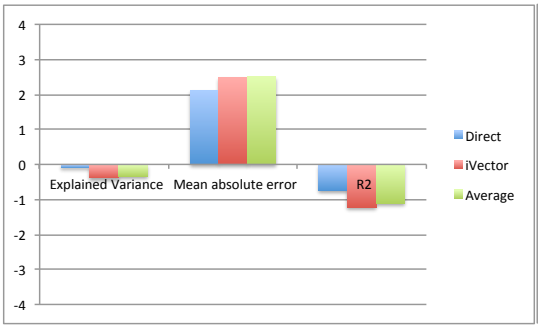
(c) Influence of smoothing on the tenaer results in Support Vector Regression a C-Penalty of 1, a maximum of 200 iterations and 256 frames per aggregation



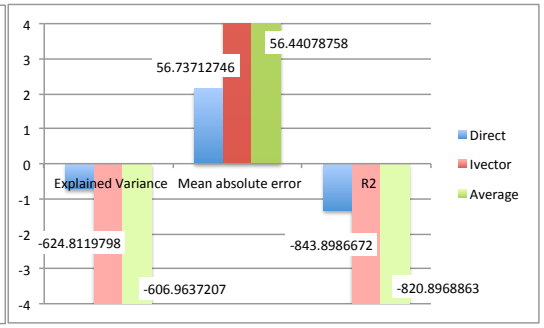
(d) Influence of smoothing on the tenaer results in Support Vector Regression with a C-Penalty of 10, a maximum of 400 iterations and 1024 frames per aggregation

Figure 6.22: Illustration of the influence of different preprocessing steps on the Support Vector Regression results

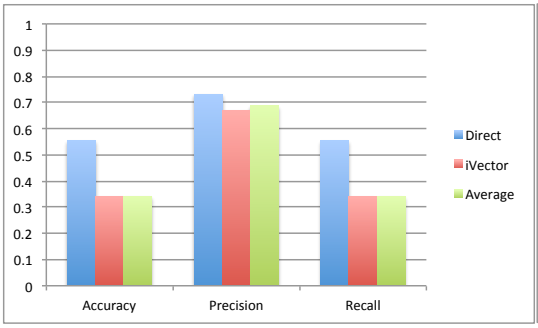
For the Random Forest regressor on the other hand (figure 6.23) we see the expected correlation of good regression results and good tenaer classification results. Also here the average and iVector processing show almost identical performances on both regression and tenaer classification. Another interesting difference is the relatively positive regression results of the smaller model compared to the larger Random Forest. On the other hand, this relation is not depicted in the tenaer results and might not have as much influence in general.



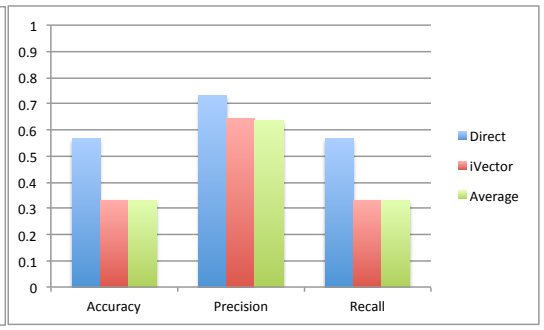
(a) Influence of smoothing on the regression results in Random Forest Regression with 10 trees, a maximal depth of 10 and 256 frames per aggregation



(b) Influence of smoothing on the regression results in Random Forest Regression with 100 trees, a maximal depth of 20 and 1024 frames per aggregation



(c) Influence of smoothing on the tenaer results in Random Forest Regression with 10 trees, a maximal depth of 10 and 256 frames per aggregation



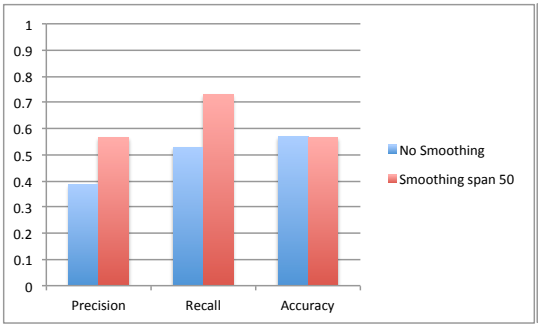
(d) Influence of smoothing on the tenaer results in Random Forest Regression with 100 trees, a maximal depth of 20 and 1024 frames per aggregation

Figure 6.23: Illustration of the influence of different preprocessing steps on the Random Forest Regression results

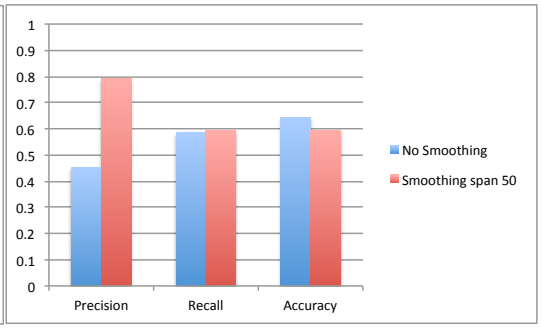
6.15 Affect Regression - Influence of smoothing

Figure 6.24 and figure 6.25 show the effects of smoothing the regression results with a moving average before doing tenaer classification. Between the different disturbances (figure 6.24), only the normal types profit without smoothing. All regressions with noise or channel effects gain performance when the results are smoothed. The explanation for this smoothing being positive, while smoothing the classification results had a clearly negative effect, is that the regression values are continuous and much more diverse in

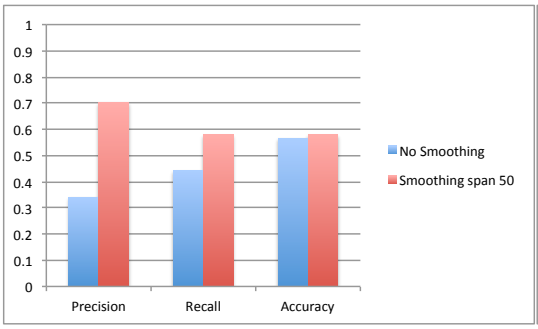
their values.



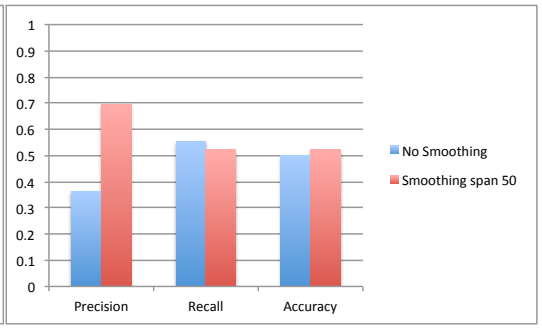
(a) Influence of smoothing the test results averaged over all audio types



(b) Influence of smoothing the test results averaged over all normal audio files



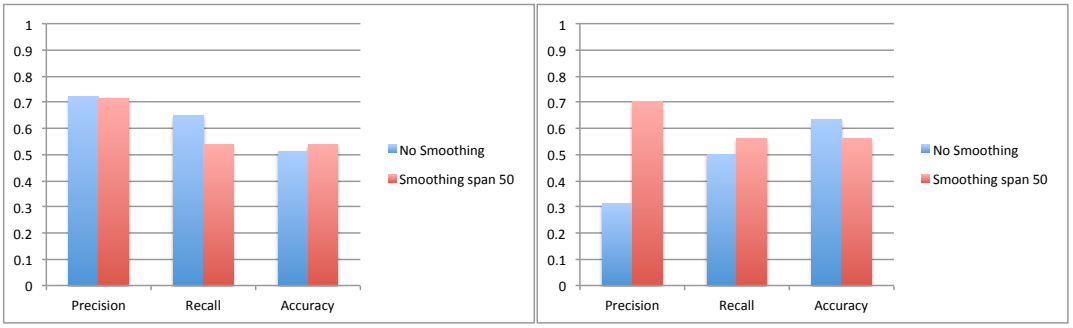
(c) Influence of smoothing the test results averaged over all noised audio files



(d) Influence of smoothing the test results averaged over all channeled audio files

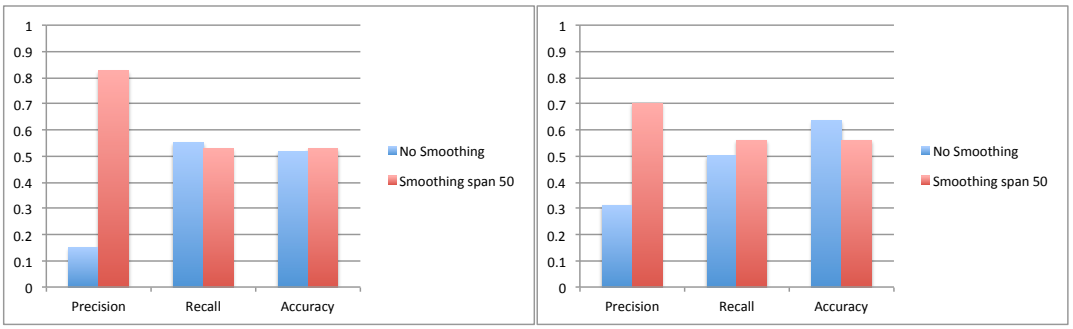
Figure 6.24: Illustration of the influence of smoothing the test results in each audio type

When comparing the smoothing effect between die different dimensions (figure 6.25) Valence and expectation need the smoothing while arousal and power loose performance when being smoothed. Those two groups are the same as in the general difference between dimension shown in figure 6.20 This indicated, that the dimension in general might not be different in their performances, but only under the influence of the smoothing, which supports or hinders some dimensions.



(a) Influence of smoothing the test results on the valence results

(b) Influence of smoothing the test results on the arousal results



(c) Influence of smoothing the test results on the expectation results

(d) Influence of smoothing the test results on the power results

Figure 6.25: Illustration of the influence of smoothing the test results in each dimension

6.16 Affect Regression - Transformation of tenaer classification to emotion

In figure 6.26 the back transformation of the classified tenaer valence and arousal values into a distinct emotion is shown. Even though the channel effects clearly diminish the results, this option might be more performing that the direct classification for both other cases. Thus a comparison table of the SEMAINE classification performance with the back-transformed tenaer results is shown in table 6.4.

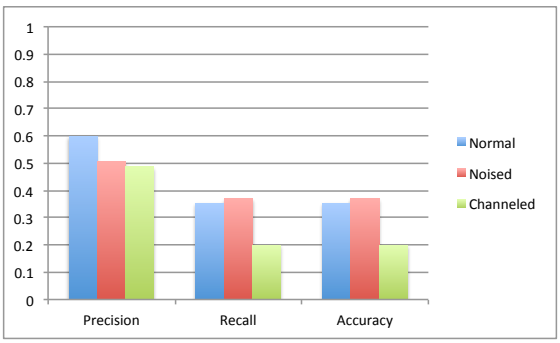


Figure 6.26: Results of combining tenaer valence and arousal predictions back into emotions

Is clear that for normal and noised data the back transformation not only outperforms the average result but also the optimal GMM result. Unfortunately for the focus group of channeled data, the performance only slightly differs from the average and is clearly outperformed by the GMM model.

Table 6.4: Comparison of classification accuracies derived from tenaer regression with accuracies of the direct classification methods

SEMAINE audiotype	Tenaer regression	Classification avg	Classification gmm
Normal	0,353	0,270	0,292
Noised	0,370	0,229	0,217
Channeled	0,199	0,193	0,239

6.17 Affect Regression - LSTM RNN configuration

To improve the performance of the LSTM RNN algorithm, different settings of parameters were tested and are shown in figure 6.27. It is clearly visible, that the selected parameters of 0.1 learning rate and 0.6 momentums are in the optimal area. A major influence on the performance seems to be the size of the aggregated windows. The closer the data comes to a direct approach, the better the performance. Unfortunately, even a widow size of only 64 frames increases the runtime beyond the scope of this thesis. In the future another investigation could be done on the performance of a direct LSTM model.

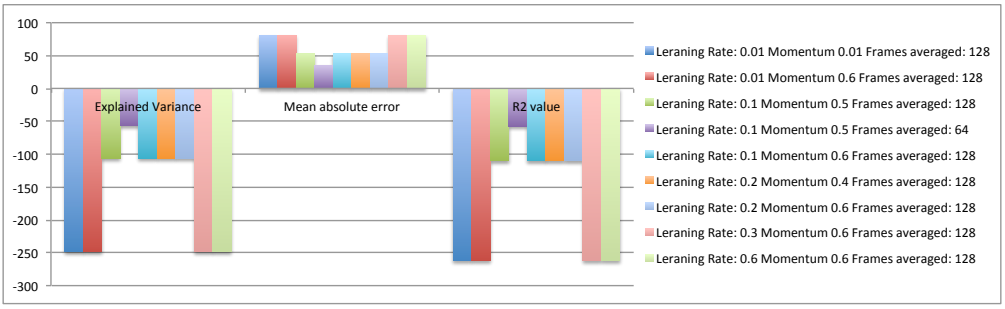


Figure 6.27: Influence of learning rate, momentum and number of aggregated frames on the LSTM results

7 Summary and outlook

7.1 Summary

The goal of this thesis was to test proven emotion recognition and affect regression methods on data that is disturbed by real life influences and additionally restricted to MFCC input features. Three different benchmarks datasets were classified with GMMs, iVectors and anchor models of both types. In summary it can be said, that given the real life disturbances and restrictions, the results for emotion classification and regression stay far behind the ones from literature. On the channeled datasets a maximum of 32.5% accuracy was reached for the FAU Aibo dataset with a GMM classifier. GMMs also seem to be the general best classifier in the different datasets and audio types. For the channel normalization techniques, the unmodified cosine similarity and the s-norm deliver best results. Using speaker dependent classification did not improve the results for this specific research. Also smoothing and shuffling had mostly negative influence on the overall result. Overall the largest dataset, FAU Aibo had the best evaluation metrics compared between the datasets.
Additionally, one of the benchmark datasets was tested for regression results with the same real life disturbances and restrictions, as mentioned above. None of the Regression models reached good R2 scores and most were not even comparable to those values from previous papers. The best model by far was the Random Forest Regressor. An improvement of those results is reached, when processing the output and reassigning tenaer or binary values to them. In that case accuracies between 50% and 60% can be achieved. Also, when retranslating these values with a valence-arousal-scheme, the results are better than the average direct classification results. Only the GMM results for channeled data outperformed them. In difference to the classification results smoothing for regression is vital instead of disturbing.
Overall the emotion recognition GMM algorithm showed the best performance with 29.75% UAR in average over all datasets and audio types. For the affect regression, the best performance was reached by the tenaer Random Forest Regressor with UAR values of 52.84 for valence, 56.04 for arousal, 53.02 for expectation and 54.010739 for power averaged over all audio types.

7.2 Outlook

One major difference to the systems other researches use for emotion recognition is the number and type of inputs. Since this thesis was restricted to MFCCs, in order to make a smart phone application feasible, a next step is to try to gain other input features from the MFCCs. That way the transmission rate stays the same and the server can calculate more features. One example for this would be to train a MFCC to pitch classifier [37], use it to predict the pitch of new data samples and then compare the results with those without pitch information.
Another step in the future is to take these models and to test them on real life data, instead of modified benchmark datasets, in order to get a comparison of the impact of disturbances in real life.
Finally, a continuing work should try to use an emotion model that is derived from all four affect dimensions instead of only valence and arousal and then compare the results of a direct classification with those results of binary and translated regression.

List of Figures

List of Tables

Bibliography

[1] C.-N. Anagnostopoulos, T. Iliou, and I. Giannoukos. "Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011." In: *Artificial Intelligence Review* 43.2 (2015), pp. 155–177.

[2] K. R. Anne, S. Kuchibhotla, and H. D. Vankayalapati. *Acoustic Modeling for Emotion Recognition*. Springer, 2015.

[3] Y. Attabi and P. Dumouchel. "Emotion Recognition from Children's Speech Using Anchor Models." In: *WOCCI* (2012).

[4] J. K. Baghel. "Audio-based characterization of conversations." In: *Master Thesis, Technische Universität München* (2014).

[5] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. *Theano: new features and speed improvements*. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop. 2012.

[6] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. "Theano: a CPU and GPU Math Expression Compiler." In: *Proceedings of the Python for Scientific Computing Conference (SciPy)*. Oral Presentation. Austin, TX, June 2010.

[7] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss. "A database of German emotional speech." In: *Interspeech*. Vol. 5. 2005, pp. 1517–1520.

[8] N. Dehak, R. Dehak, J. R. Glass, D. A. Reynolds, and P. Kenny. "Cosine Similarity Scoring without Score Normalization Techniques." In: *Odyssey*. 2010, p. 15.

[9] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. "Front-end factor analysis for speaker verification." In: *Audio, Speech, and Language Processing, IEEE Transactions on* 19.4 (2011), pp. 788–798.

[10] F. Eyben, M. Wöllmer, and B. Schuller. "A multitask approach to continuous five-dimensional affect sensing in natural speech." In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 2.1 (2012), p. 6.

[11] F. Eyben, M. Wöllmer, and B. Schuller. "OpenEAR?introducing the Munich open-source emotion and affect recognition toolkit." In: *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*. IEEE. 2009, pp. 1–6.

[12] F. Eyben, M. Wöllmer, and B. Schuller. "Opensmile: the munich versatile and fast open-source audio feature extractor." In: *Proceedings of the international conference on Multimedia*. ACM. 2010, pp. 1459–1462.

[13] J. R. Fontaine, K. R. Scherer, E. B. Roesch, and P. C. Ellsworth. "The world of emotions is not two-dimensional." In: *Psychological science* 18.12 (2007), pp. 1050–1057.

[14] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue. *TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1*. Tech. rep. Web Download. Philadelphia: Linguistic Data Consortium, 1993.

[15] F. Gers. "Long short-term memory in recurrent neural networks." In: *Unpublished PhD dissertation, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland* (2001).

[16] A. Graves. "Generating sequences with recurrent neural networks." In: *arXiv preprint arXiv:1308.0850* (2013).

[17] A. Graves et al. *Supervised sequence labelling with recurrent neural networks*. Vol. 385. Springer, 2012.

[18] A. O. Hatch, S. S. Kajarekar, and A. Stolcke. "Within-class covariance normalization for SVM-based speaker recognition." In: *Interspeech*. 2006.

[19] H. He, E. Garcia, et al. "Learning from imbalanced data." In: *Knowledge and Data Engineering, IEEE Transactions on* 21.9 (2009), pp. 1263–1284.

[20] S. Hochreiter and J. Schmidhuber. "Long short-term memory." In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[21] T. Iliou and C.-N. Anagnostopoulos. "Comparison of different classifiers for emotion recognition." In: *Informatics, 2009. PCI'09. 13th Panhellenic Conference on*. IEEE. 2009, pp. 102–106.

[22] Y. Jiang, K. A. Lee, and L. Wang. "PLDA in the i-supervector space for text-independent speaker verification." In: *EURASIP Journal on Audio, Speech, and Music Processing* 2014.1 (2014), pp. 1–13.

[23] Z. N. Karam, W. M. Campbell, and N. Dehak. "Towards reduced false-alarms using cohorts." In: *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE. 2011, pp. 4512–4515.

[24] T. Kinnunen and H. Li. "An overview of text-independent speaker recognition: From features to supervectors." In: *Speech communication* 52.1 (2010), pp. 12–40.

[25] A. Liaw and M. Wiener. "Classification and regression by randomForest." In: *R news* 2.3 (2002), pp. 18–22.

[26] Y. Liu. *Emotion Recognition from Speech*. URL: \url{https://community.apan.org/afosr/m/trust__influence_program_review/121315/download.aspx} (visited on 09/11/2015).

[27] G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. Schröder. "The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent." In: *Affective Computing, IEEE Transactions on* 3.1 (2012), pp. 5–17.

[28] G. McKeown, M. F. Valstar, R. Cowie, and M. Pantic. "The SEMAINE corpus of emotionally coloured character interactions." In: *Multimedia and Expo (ICME), 2010 IEEE International Conference on*. IEEE. 2010, pp. 1079–1084.

[29] M. Pitt, L. Dilley, K. Johnson, S. Kiesling, W. Raymond, E. Hume, and E. Fosler-Lussier. *Buckeye Corpus of Conversational Speech (2nd release) [www.buckeyecorpus.osu.edu]*. Tech. rep. Department of Psychology Ohio State University (Distributor) Columbus OH USA, 2007.

[30] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. "Speaker verification using adapted Gaussian mixture models." In: *Digital signal processing* 10.1 (2000), pp. 19–41.

[31] J. A. Russell. "A circumplex model of affect." In: *Journal of personality and social psychology* 39.6 (1980), p. 1161.

[32] S. N. Satyavolul. *Score Normalization techniques for text-independent speaker verification*. URL: \url{http://home.iitk.ac.in/~snitish/Stuff/Score_normalization_report.pdf} (visited on 09/11/2015).

[33] B. Schuller, M. Valstar, F. Eyben, G. McKeown, R. Cowie, and M. Pantic. "Avec 2011–the first international audio/visual emotion challenge." In: *Affective Computing and Intelligent Interaction*. Springer, 2011, pp. 415–424.

[34] B. Schuller, B. Vlasenko, F. Eyben, G. Rigoll, and A. Wendemuth. "Acoustic emotion recognition: A benchmark comparison of performances." In: *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE. 2009, pp. 552–557.

[35] F. Schulze and G. Groh. "Studying how character of conversation affects personal receptivity to mobile notifications." In: *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM. 2014, pp. 1729–1734.

[36] T. Seitle. "Audio-Based Social Interaction Detection for Mobile Interruption Management." In: *Master Thesis, Technische Universität München* (2014).

[37] X. Shao and B. P. Milner. "MAP prediction of pitch from MFCC vectors for speech reconstruction." In: *INTERSPEECH*. Citeseer. 2004.

[38] A. J. Smola and B. Schölkopf. "A tutorial on support vector regression." In: *Statistics and computing* 14.3 (2004), pp. 199–222.

[39] S. Steidl. *Automatic classification of emotion related user states in spontaneous children's speech*. University of Erlangen-Nuremberg Germany, 2009.

[40] S. Steidl. *Vocal Emotion Recognition*. URL: \url{http://web.stanford.edu/class/cs424p/materials/steidl.pdf} (visited on 09/11/2015).

[41] A. Stuhlsatz, C. Meyer, F. Eyben, T. ZieIke, G. Meier, and B. Schuller. "Deep neural networks for acoustic emotion recognition: raising the benchmarks." In: *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE. 2011, pp. 5688–5691.

[42] A. Taleb. "Audio-based Characterization of Conversations with Channel Compensation." In: *Master Thesis, Technische Universität München* (2015).

[43] R. Togneri and D. Pullella. "An overview of speaker identification: Accuracy and robustness issues." In: *Circuits and Systems Magazine, IEEE* 11.2 (2011), pp. 23–61.

[44] *Valence Arousal*. URL: \url{http://1.bp.blogspot.com/-YJuKVgRNav0/Uj_EXR4MH4I/AAAAAAAABAY/44JUYlasmZw/s1600/arval.png} (visited on 09/11/2015).

[45] *VoxForge*. URL: \url{http://www.voxforge.org/home/about} (visited on 09/11/2015).

[46] D. Wu, B. Li, and H. Jiang. *Normalization and Transformation Techniques for Robust Speaker recognition*. INTECH Open Access Publisher, 2008.

[47] R. Xia and Y. Liu. "Using i-vector space model for emotion recognition." In: *Thirteenth Annual Conference of the International Speech Communication Association*. 2012.