INTERSPEECH 2009 BRIGHTON

Brno University of Technology System for Interspeech 2009 Emotion Challenge

Marcel Kockmann[1] [2], *Lukáš Burget*[1] *and Jan "Honza" Černocký*[1]

[1]Speech@FIT, Brno University of Technology, Czech Republic
[2]SVOX Deutschland GmbH, Munich, Germany

{kockmann|burget|cernocky}@fit.vutbr.cz

Abstract

This paper describes Brno University of Technology (BUT) system for the Interspeech 2009 Emotion Challenge. Our submitted system for the Open Performance Sub-Challenge uses acoustic frame based features as a front-end and Gaussian Mixture Models as a back-end. Different feature types and modeling approaches successfully applied in speaker- and language recognition are investigated and we can achieve an 16% and 9% relative improvement over the best dynamic and static baseline system on the 5-class task, respectively.

Index Terms: Emotion recognition, GMM, MMI, JFA

1. Introduction

Gaussian Mixture Modeling has become the standard modeling approach in tasks like Speaker, Gender and Language Identification (SID/GID/LID). Universal Background Modeling (UBM) with adaptation to target model [1] is the standard in speaker verification. Discriminative training techniques such as training based on Maximum Mutual Information (MMI) have been applied very successfully to classification tasks like Gender- or Language Identification [2]. An eternal problem in all these tasks is the diversity in channel and acoustic condition between training and test data. Joint Factor Analysis [3] has become the standard to cope with this mismatch recently, even on small amounts of training and test data.

Our goal is to apply features and modeling techniques that are used in SID/LID to the related problem of emotion recognition. As it is mentioned in [4], Support Vector Machines (SVM) are mostly used to classify on a high dimensional chunk based feature. These techniques have also been applied in the field of SID and LID and combination with GMM based approaches gains huge improvements.

Our submission is for the Open Performance Sub-Challenge (recognizing 2 and 5 emotion classes with own contribution of features and classifier) [4]. Section 2 describes the system development and gives information on the acoustic and prosodic features we use as well as the modeling techniques for Gaussian mixtures we want to adapt to this task. We also provide experimental results on a development set and section 3 presents results for our final selected submission on the real test set. In section 4 we draw conclusions to our approach.

This work was partly supported by European project AMIDA (FP6-033812), by Grant Agency of Czech Republic project No. 102/08/0707, by Czech Ministry of Education project No. MSM0021630528 and by Czech Ministry of Trade and Commerce project No. FT-TA3/006.

2. System development

2.1. Features

2.1.1. MFCCs

The most widely used features in speech processing are Mel-Frequency Cepstral Coefficients (MFCC). They have been applied successfully for speech recognition as well as for speaker recognition and language identification. We will use them as our basic features for the emotion recognition task. MFCC vectors are generated every 10ms on a 20ms Hamming window. A Mel filter bank with 25 bands is used to create features with 13 coefficients including C0. Then, mean subtraction is applied on each coefficient per utterance. As audio files are provided at 16kHz sampling frequency, we create two sets of features. One with full resolution and one down-sampled to 8kHz and filtered from 300-3400 Hz (as it is common for telephone applications).

2.1.2. RASTA filter

The temporal trajectories of individual cepstrum feature vector coefficients are filtered using standard RelAtive SpecTrAl (RASTA) filter [5] to remove slow and very fast spectral changes which do not appear to be characteristic for natural speech.

2.1.3. VTLN

Like in language identification, we do not want to model the characteristics of the individual speaker and the position of the formants based on the length of the vocal tract. We use Vocal Tract Length Normalization (VTLN) [6] for simple speaker adaptation. Warping factors for training and test data are estimated using GMM trained on all unnormalized training data. Warped MFCCs are created for all files with warping factors in a range from 0.80-1.12 with a step-size of 0.02. The optimal warping factor per chunk is obtained by scoring all warped instances against the unnormalized GMM and selecting the maximum. We use a linear piecewise warping function with a warping cutoff of $0.875 \times N_f$, which is the Nyquist frequency.

2.1.4. Temporal context

Simple MFCCs do not model any temporal characteristics which might be a discriminative feature for this task. For this purpose, we generate delta, double and triple delta coefficients of the static features. This results in 26, 39 and 52 dimensional feature vectors containing information spanning a context of 5, 9 and 13 frames, respectively.

2.1.5. SDC

The importance of even a broader temporal information has been shown for LID [2]. The Shifted Delta Cepstra (SDC) features are created by stacking delta-cepstra computed across multiple speech frames. The SDC features are specified by the number of cepstral coefficients (7), the advance and delay for the delta-computation (1), the number of blocks whose delta-coefficients are concatenated to form the final feature vector (7), and the time shift between consecutive blocks (3). The features in our system are 7 MFCC coefficients (including coefficient C0) concatenated with delta cepstra which totals in 56 coefficients per frame, spanning a context of 21 frames.

2.1.6. Voice Activity Detection

For all our features, non-speech frames are discarded and only speech frames are considered in the following stages of training models and verification. Speech/non-speech segmentation is performed by our Hungarian phone recognizer [9], where all phoneme classes are linked to speech class.

2.1.7. Syllable Contours

Prosodic information based on a lexical context might be useful for this task and are complementary to the acoustic short time features. For this purpose, we use our detector of syllable based feature contours as presented in [7], based on classical prosodic features like duration, pitch and energy in a syllable-like temporal context. The trajectories of each feature are continuously modeled over the time span of a syllable and are represented by discrete cosine transformation (DCT) coefficients. Additionally, we also capture the contours of MFCCs and form a single feature vector out of duration, pitch, energy and the MFCC contours. Frame based pitch and energy is generated and are mean subtracted over the voiced part of the utterance before approximating the temporal trajectory. We use the syllable duration (number of frames) and 6 coefficients per feature contour which results in 13 dimensional vectors for the prosodic and 85 dimensional vectors for the combined prosodic and MFCC contours.

2.2. Classifier

2.2.1. GMM-UBM System

The baseline GMM system is based on standard Universal Background Model-Gaussian Mixture Modeling (UBM-GMM) paradigm [1]. Weights, means and variances of the UBM are trained iteratively prior to any class model on data from all classes by Expectation-Maximization (EM) and class models are derived via relevance Maximum-a-Posteriori (MAP) adaptation. Instead of frame-based full log-likelihood evaluation, we perform an approximate fast linear scoring based on utterance statistics [8].

2.2.2. Maximum Mutual Information

MMI is a discriminative training technique often applied to classification tasks similar to emotion recognition [2]. For this approach, an initial set of models is trained per class under conventional Maximum Likelihood (ML) framework, as for the UBM. These serve as a starting point for further discriminative re-estimations of means and variances using Maximum Mutual Information criterion.

Unlike in the case of ML training, which aims to maximize the overall likelihood of training data, MMI objective function to maximize is the posterior probability of correctly recognizing all training segments (chunks):

$$\mathcal{F}_{MMI}(\lambda) = \sum_{r=1}^{R} \log \frac{p_\lambda(\mathcal{O}_r|s_r)P(s_r)}{\sum_{\forall s} p_\lambda(\mathcal{O}_r|s)P(s)}, \quad (1)$$

where $p_\lambda(\mathcal{O}_r|s_r)$ is the likelihood of r-th training segment, $\mathcal{O}_r$, given the correct transcription (in our case the correct emotion class identity) of the segment, s_r, and model parameters, λ. R is the number of training segments. The denominator represents the overall probability density, $p_\lambda(\mathcal{O}_r)$ (likelihood given any emotion class). We consider the prior probabilities of all classes (emotions) equal and drop the prior terms $P(s_r)$ and $P(s)$.

In this approach, verification is done frame-by-frame with full log-likelihood computation.

2.2.3. Joint Factor Analysis

Joint factor analysis is a model recently introduced to cope with the problem of speaker and session variability in GMM-based speaker verification [3]. We explain the basic concept in terms of SID and show how to adapt it to the emotion classification problem. The basic assumption is that the model $\mathbf{M}$ can be decomposed into a speaker $\mathbf{s}$- and channel $\mathbf{c}$- dependent part

$$\mathbf{M} = \mathbf{s} + \mathbf{c}, \quad (2)$$

which can be represented in low dimensional spaces. The first term on the right hand side of (2) is modeled by assuming that if $\mathbf{s}$ is the speaker super-vector for a randomly chosen speaker, then

$$\mathbf{s} = \mathbf{m} + \mathbf{V}\mathbf{y} + \mathbf{D}\mathbf{z}, \quad (3)$$

where $\mathbf{m}$ is the speaker- and channel-independent super-vector (UBM), $\mathbf{V}$ is a rectangular matrix of low rank (eigenvoices), $\mathbf{D}$ is a diagonal matrix (which covers the residual speaker variability) and the components of $\mathbf{y}$ and $\mathbf{z}$ are respectively the speaker and common *factors*.

The channel-dependent part $\mathbf{c}$, which represents the channel effect in an utterance, is assumed to be distributed according to

$$\mathbf{c} = \mathbf{U}\mathbf{x}, \quad (4)$$

where $\mathbf{U}$ is a rectangular matrix of low rank (known as eigenchannel matrix) and the components of $\mathbf{x}$ are the channel factors.

The underlying task in JFA is to train the hyperparameters $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{D}$ on a large training set which represents subspaces in which speaker and channel can be rapidly adapted.

In the task of emotion recognition, the eigenvoices can be seen as a low dimensional subspace modeling the properties of the different emotion classes and the "channel" covers unwanted attributes like the individual speaker and other session variability.

2.3. Experiments on Development Set

2.3.1. Setup

As there is no official development set we take a subset of the training data for system development. To get expressive results, we use a full jackknifing approach for the whole training set. 13 splits are created out of the training set, each excluding 1 male and 1 female, resulting in ca. 9000 chunks. Results are presented in terms of recognition rate (percentage of correctly recognized chunks, in total for all chunks and averaged over the classes). In this section, they will be presented for the 5-class task (**A**nger, **E**mphatic, **N**eutral, **P**ositive and **R**est) only.

Table 1: *Results for plain MFCCs, with RASTA and VTLN [%].*

feature	avg	tot	A	E	N	P	R
8kHz	35.6	37.1	48.9	28.7	40.2	52.7	7.3
RASTA	38.8	40.2	55.1	27.4	44.3	61.8	5.3
VTLN	36.3	38.4	48.4	28.7	42.5	56.8	4.9
16kHz	36.4	40.4	45.9	23.1	48.2	58.8	6.4
RASTA	37.4	40.9	55.2	30.6	45.7	50.5	5.0
VTLN	36.8	40.2	44.1	27.5	46.2	59.4	6.7

Table 2: *Results with longer temporal context [%].*

feature	avg	tot	A	E	N	P	R
8k Δ	40.6	41.1	56.8	30.4	43.9	66.2	5.7
Δ Δ	40.7	42.2	54.8	29.4	46.4	68.5	4.3
$\Delta\Delta\Delta$	41.0	40.6	56.3	30.4	42.7	70.4	4.9
SDC	42.2	41.3	54.8	44.6	39.3	68.7	3.6
16k Δ	41.8	41.3	59.3	33.3	42.7	69.6	4.1
$\Delta\Delta$	**43.5**	42.9	60.8	36.5	43.6	71.4	5.3
$\Delta\Delta\Delta$	42.6	40.7	58.1	38.9	39.2	72.2	4.8
SDC	41.9	41.0	52.6	46.7	38.4	68.2	3.7

2.3.2. Features

First experiments are carried out with the simple GMM-UBM system to find well performing features. Preliminary experiments indicated that a size of 64 Gaussians performs the best in average. Due to the unbalanced amount of class affiliation in the training data, it is important to define a balanced set for the UBM training. Results for different feature types are presented in table 1.

For the MFCC that are generated from the 16kHz data as well as from the downsampled audio, we achieve improvements through the RASTA filter. With VTLN, we gain less improvement. The plain 16kHz MFCCs perform better than the 8kHz ones, but better results are achieved with RASTA on the 8kHz data.

After filtering the MFCC features, we augment them with their derivatives up to third order to cover temporal context. Alternatively, we use SDC features with even a longer temporal context.

From table 2 it can be seen that the performance clearly benefits from adding the dynamic information. Here, the 16kHz data gains much more from the delta coefficients and the double deltas outperform the triple deltas as well as the SDC features.

Our last feature experiment is carried out on features that are generated for each syllable in the utterance. The duration and contours of pitch, energy (DPE) and optionally MFCCs (DPEC) are modeled by these features. Table 3 indicates that they perform quite worse compared to the frame based acoustic features, while they still might contain complementary information. A problem for statistical modeling is that we get very few feature frames for the short test utterances, often none at all (as there might be no detected pitch).

2.3.3. Classifier

As a second step, we use our best performing features on more sophisticated modeling approaches as presented in section 2.2.

For the discriminative training, we use ML-trained models as initial models for each class, and we retrain them using MMI training in 10 iterations.

Table 3: *Results for syllable based feature contours [%].*

feature	avg	tot	A	E	N	P	R
DPE	32.3	39.6	43.3	23.8	49.2	35.6	9.7
DPEC	36.0	38.3	47.7	30.9	41.7	48.7	10.9

Table 4: *Results for ML and MMI systems [%].*

feature	avg	tot	A	E	N	P	R
ML	44.0	49.2	51.5	45.0	54.1	46.3	23.1
MMI	43.7	49.5	49.5	45.1	55.0	44.0	24.8

Although the ML trained models perform slightly better than the UBM-GMM system, the performance even slightly degrades for the MMI trained system, as shown in table 4. This might be due to overtraining on the relatively small amount of data. Anyway, this system might give complementary information to the GMM-UBM system as can be seen e.g. from the much higher recognition rate for class R.

Starting point for our JFA experiments is the GMM-UBM system as JFA is an expansion to this system. Initially, the low dimensional sub-spaces that model the attributes of the class and the "channel" have to be estimated. The estimation of eigenvoices does not promise much gain in performance in this case, as only 2 or 5 classes are available for training the subspace and the space is estimated on the same data as used for the model training itself.

More interesting is the question how to estimate the "channel" matrix. One can assume to model the variability over all chunks (which would cover the speaker as well as general chunk variability) or to sum up all statistics belonging to one speaker. After preliminary tests, we use the second case where the "channel" represents the dimensions of unwanted variability caused by the individual speaker.

We initialize $\mathbf{V}$ and $\mathbf{U}$ by PCA [10] and iteratively retrain first $\mathbf{V}$, then $\mathbf{U}$, and then $\mathbf{D}$. For small amounts of test data, the integrative scoring over the channel distribution [8] has proved to be beneficial. On this task, we have an average of 80 frames of speech per test utterance (0.8s) which is extremely little data for statistical adaptation.

Table 5 shows some experiments to find suitable numbers of eigenvoices ($\mathbf{V}$) and eigenchannels ($\mathbf{U}$). As we use small models and have very little adaptation data, we get only some improvement with one eigenchannel. Increasing the number even degrades the performance. Also, the use of more eigenvoices decreases the recognition rate.

Although we get some improvement, this is negligible compared to the improvements of over 50% relative gained in SID [10]. This is mainly due to the small amount of adaptation data for the channel estimate.

2.4. System Calibration/Fusion

Finally, we use multiclass linear regression tool [11] to perform calibrated fusion of our systems.

Fusion parameters are trained directly on the development set. Improvement is gained through all performed fusions of 2 systems, see Table 6. Combination of two JFA systems with different feature types results in the best performance.

Table 5: *Results for different JFA system configurations [%].*

V	U	avg	tot	A	E	N	P	R
0	0	43.5	42.9	60.8	36.5	43.6	71.4	5.3
1	1	**44.4**	47.1	52.8	42.3	50.3	57.3	19.6
1	2	43.0	41.4	53.5	43.2	39.2	58.1	21.3
1	3	42.9	42.4	55.5	43.0	41.4	53.7	20.9
5	1	43.0	46.5	52.1	42.9	49.9	53.4	16.8
0	1	44.2	47.2	52.8	42.1	50.6	56.7	19.0

Table 6: *Results for fusion of 2 systems [%].*

System 1	System 2	avg
GMM-UBM	JFA	45.3
MMI	JFA	46.65
ML	JFA	46.45
MFCC RASTA Δ Δ	SDC	**47.18**

3. Submission

This section shows the results for the systems we have selected to submit for the official Open Performance Challenge [4]. Results are presented with the official metric on the 2- and 5-class task.

3.1. Systems

We have selected the four different modeling approaches (ML, MMI, GMM-UBM, JFA) we used in the system development for the best performing features on the test set. They are based on MFCCs generated from 16kHz data with RASTA filter and double deltas.

Table 7 shows results for the 2-class task. Consistent to our development set, we get the best results for the JFA system, while the others perform approximately the same. On the primary measure, the unweighted average recall (UA), we achieve only a minor 3%/1% relative improvement to the dynamic and static modeling, respectively, which was provided as a baseline [4].

Table 8 shows results for the 5-class task. Like for the 2-class, we achieve the best results for the JFA system. Surprisingly, the ML and the MMI system perform worse, unlike than on the development set. This might indicate that even the ML trained model is already over-adapted to the training data. On the UA, we achieve a 15%/8% relative improvement to the dynamic and static modeling, respectively, which was provided as a baseline [4].

Our final submission is a fusion of the two JFA systems fused in table 6, based on the averaged parameters trained on the splits of the development set. This sub-optimal fusion results in further improvement to 41.7% for 5-class task.

4. Conclusions

For our submitted systems, we could achieve relative improvement of 16% over the low-level descriptor baseline system and 9% relative improvement to the static modeling baseline using supra-segmental information for the 5-class task.

Although we could not achieve great benefit from applying MMI or JFA, we could show that a GMM-based approach with relatively "simple" features containing only acoustic information on a frame level can yield to a comparable or even better performance than using much higher dimensional chunk-based features.

Table 7: *Submitted systems for 2-class task [%].*

System	UA	WA
GMM-UBM	67.8	64.2
JFA	**68.3**	65.8
ML	67.5	63.8
MMI	67.65	64.1

Table 8: *Submitted systems for 5-class task [%].*

feature	UA	WA
GMM-UBM	40.8	41.0
JFA	**41.3**	43.9
ML	38.5	45.4
MMI	38.7	46.0

As we have observed in other areas, the small benefit from MMI and JFA is often due to small amount of training and test data. The benefit from JFA that can be in the range of 50% relatively for several minutes of speech (typical SID task) reduces to less than 10% for a few seconds of speech. This becomes even more dramatic for the syllable based contour features.

Also, appropriate feature type still has to be found. Standard features like MFCCs do work, but it is obvious that for detecting emotions in speech a simple "acoustic fingerprint" may not be sufficient, especially if the emotion covers only a few words.

5. References

[1] Reynolds, D. A. et al., "Speaker Verification Using Adapted Gaussian Mixture Models", Dig. Sig. Proc. 10, p. 19-41 (2000).

[2] Matejka, P. et al., "Brno University of Technology System for NIST 2005 Language Recognition Evaluation", in proc. of A Speaker Odyssey, 2006, p. 57-64.

[3] Kenny, P. et al., "A Study of Inter-Speaker Variability in Speaker Verification", in IEEE Trans. Audio, Jul 2008, Vol. 16, p. 980-988.

[4] Schuller, B.; Steidl, S.; Batliner, A.: The Interspeech 2009 Emotion Challenge, Interspeech (2009), ISCA, Brighton, UK, 2009.

[5] Hermansky, H. Morgan, N., "RASTA processing of speech", in IEEE Trans. Audio, Oct 1994, Volume 2, p. 578-589.

[6] Cohen, J. et al., " Voacl tract normalization in speech recognition: Compensating for systematic speaker variability", J. Acoust. Soc. Am., May 1995, Volume 97, p. 3246-3247.

[7] Kockmann, M. and Burget, L. "Contour Modeling of Prosodic and Acoustic Features for Speaker Recognition", in proc. of Spoken Language Technology, 2008, p. 45-48.

[8] Glembek, O. et al. "Comparison of Scoring Methods used in Speaker Recognition With Joint Factor analysis", accepted to ICASSP, 2009, p. 4057-4060.

[9] Schwarz, P. et al., "Hierarchical structures of neural networks for phoneme recognition", in proc. of ICASSP, Toulouse, 2006, p. 325-328.

[10] Burget, L. et al.,"Analysis of feature extraction and channel compensation in GMM speaker recognition system," in IEEE Trans. Audio, Sep 2007, Volume 15, p. 1979-1986.

[11] Brümmer, N., "FoCal Multi-class", Online on: http://niko.brummer.googlepages.com/focalmulticlass.