Determining Optimal Features for Emotion Recognition from Speech by applying an Evolutionary Algorithm

David Hübner, Bogdan Vlasenko, Tobias Grosser, Andreas Wendemuth

Department of Electrical Engineering and Information Technology, Otto von Guericke University Magdeburg, Germany
{david.huebner, bogdan.vlasenko, tobias.grosser, andreas.wendemuth}@ovgu.de

Abstract

The automated recognition of emotions from speech is a challenging issue. In order to build an emotion recognizer well defined features and optimized parameter sets are essential. This paper will show how an optimal parameter set for HMM-based recognizers can be found by applying an evolutionary algorithm on standard features in automated speech recognition. For this, we compared different signal features, as well as several architectures of HMMs. The system was evaluated on a non-acted database and its performance was compared to a baseline system. We present an optimal feature set for the public part of the SmartKom database.

Index Terms: Emotion Recognition, Evolutionary Algorithms, Feature Optimization, Hidden-Markov Models

1. Introduction

The interaction between men and machines using language is nowadays becoming more and more self-evident, but machines still lack of many human abilities which would considerably simplify communication and would also help to increase the acceptance of such systems. For some time, research activities also focus stronger on the emotional aspect of speech. Exploiting information about the emotional state of a user, machines can be enabled to adapt their dialog strategy online, depending on the user’s emotions and hence react in a more appropriate and empathic manner. As emotion recognition in many applications goes hand in hand with automated speech recognition (ASR) it would be favorable to make use of the same features or even a subset thereof. Especially small devices like smart phones or PDAs, which do not provide huge computational power would benefit from such a sparse feature approach. Parallel research of other groups bases on pooling together (high level) features including the application of brute force methods in order to fully exploit the feature space (compare [1]). This paper however describes an evolutionary strategy (ES) of finding an optimal sparse feature set, given not more than the common acoustic features used in ASR. Applying an ES has the advantage of self adaptation of its parameters and is able to find optimal parameter constellations in high dimensional search spaces and further gives insights into the relevance of each parameter in terms of the model’s accuracy. Especially in case of many parameters with unknown relationships ES quickly avoids wasting time on generating and testing unsuitable parameter combinations as the evolutionary force minimizes the probability of the evolvement of such combinations effectively. In ASR Mel-Frequency-Cepstral-Coefficients (MFCCs) have established as a basic feature in order to train phoneme based recognizers. As we do not want additional parameters to be extracted from the speech signal we concentrate only on MFCCs, which have also proven to perform well in emotion recognition during the Emotion Challenge within Interspeech 2009 (compare [2]).

This paper is structured as follows: In Section 2 we describe the spontaneous database and the emotions we want to recognize. Section 3 describes, which parameters we investigated and in which range they were allowed to change during the evolution process. Section 4 introduces the evolutionary algorithm, shows how fitness is measured and the population evolves over time. The results are presented in Section 5 and compared to our baseline recognizer. Finally Section 6 summarizes our findings and gives an outlook.

2. The Data Base

Since we later want to be able to recognize emotions in realistic scenarios, we chose the public part of the SmartKom database [3] which provides spontaneous emotional dialogs recorded in noisy environment. This material sounds more natural and less exaggerated than acted material. We selected 1887 files (1124 female/ 763 male utterances), where each file contains up to 2 emotion changes. The original labeling consists of seven emotion classes: anger/irritation, helplessness, pondering, neutral, surprise, joy/gratitude and unidentifiable emotions. Further all classes but neutral are optionally weighted as weak and strong. Taking the amount of training material into account we decided to reduce the classes in order to gain superior classification results. According to [4] we merged the classes into a 4-class scenario without considerable loss of generality. The final classes are neutral, joy, helplessness and anger. Further we decided to preprocess the data since the material was too noisy to build a well performing recognizer. A huge part of numerous files consists of traffic noise, which causes misrecognitions especially in the pauses between the dialog turns. Unfortunately these parts are not annotated as such, but contribute to the true emotions on the level of annotation. In order to push the recognition performance we applied a self-written tool which analyzes the normalized energy values of each sample and creates a new file by omitting all samples which are below a threshold of 0.42. Applied to our baseline system, this approach resulted in a gain in performance of up to 15% for each of the four emotion classes. Further details are described in [5].

3. The Parameters

In former investigations [6], we found mainly four parameters described below which have strong influence on the final recog-
nition rates. In order to apply an ES for the optimization of these parameters we created a population consisting of individuals which differ with respect to the parameters. Some of the parameters were allowed to change during the evolution, while some others had to be kept fixed. Table 1 summarizes all parameters of an individual, starting with the free evolvable ones.

### 3.1. Free Parameters

The 1st parameter addresses the number of MFCCs which are used for training. Using the Hidden Markov Toolkit (HTK), which is provided by the University of Cambridge, the tool HCopy allows to extract the first $2$ to $n$ MFCCs from the speech signal, where $n \in [2..12]$. The 2nd parameter describes different types of information which can be added to each MFCC. Here one has to decide whether the zero coefficients MFCC-0 or the energy terms MFCC-E shall be additionally computed. Further for each MFCC the corresponding delta (D) and acceleration (A) values given by the 1st and 2nd derivative can be added to the feature vectors. Hence the smallest number of MFCC features of an individual can originate by applying only the first two MFCCs plus MFCC-0/E resulting in 3 features/sample, while a maximum of 39 features per sample is represented by an individual including all 12 MFCCs plus MFCC-0/E plus the corresponding delta and acceleration values. The 3rd parameter reflects the internal structure of the HMMs, i.e. the number of hidden states each HMM provides. In our case each of the 4 emotions is represented by its own HMM. Within the HMM states the temporal flow within each emotion is modeled. In ASR commonly 3 states are used, modeling the initial, the middle and the end part of each phoneme in a wider sense. In [6] we were able to confirm for acted databases that initial, the middle and the end part of each phoneme in a wider range between 2-12 cycles. The 4th parameter represents the number of the model’s training cycles. In case this rate is too low the model remains untrained while a huge number of may lead to no further improvement or even to an overfitted model. We investigate iterations in the range between 2 and 12 cycles.

### 3.2. Fixed Parameters

Parameters listed in the lower part of Table 1 are fixed for all individuals and mainly become important during the feature extraction. These static parameters are necessary in order to draw conclusions about the influence of the four parameters we want to investigate. We also examined an approach with individual training and test sets, but the gain in performance during evolution was here mainly due to finding a perfect distribution of training and test material, which resulted in a poor generalization on the evaluation set. Especially for reasons of comparability of the individuals in terms of their performance, we decided for a static training and test set for all individuals. In order to compensate the variance in data we repeated the complete evolution 10 times on randomly chosen training, test and evaluation sets and averaged over the results.

### 4. The Evolutionary Algorithm

The aim of the algorithm is to iteratively eliminate unfavorable parameter combinations by simulating an evolutionary process. The first step consists of generating the fixed training, test and evaluation sets from the SmartKom data. The material of the test set is used to derive evolutionary parameters like an individual’s performance, reproduction probability (fitness) and mutation rate. The evaluation set however is independent from the adaptation procedure and it is merely used to prove the results with respect to generalization properties. The evaluation set consists of 100 items, while the remaining data is split up in 90% training and 10% test material which corresponds to 1608 respectively 179 items in absolute.

#### 4.1. Generating an initial population

In the next step an initial population is generated containing 100 individuals. For each individual the values for the free parameters are randomly chosen from a uniform distribution, hence each value is selected with equal probability. Before the training of the single individuals starts each HMM is initialized by applying the HCopy tool of HTK on data which represents the corresponding emotion. For instance the state mean and variance parameters for an HMM representing the emotion anger are initialized equal to the global speech mean and variance by analyzing data that contains the static emotion anger.

#### 4.2. Training and testing the population

After the creation of these flat-starts for the single emotions the actual training of the individuals starts. According to their individual parameters the HTK tool HERest is applied for the individual’s number of training cycles. In the following step the whole population is evaluated against both the test and the evaluation set. For this we applied HVite and HResults from the HTK toolbox. Further we evaluated the confusion matrix, which is part of the output of HResults. The matrix indicates how well the current parameter combination performs on the test and evaluation set. We measured the performance for the single emotion as the ratio of correct recognitions (confusions). From this we derived an overall performance measure being the average of the single emotion performances.

#### 4.3. Natural selection, reproduction and mutation

The next step represents the selection and mutation process, where each individual either dies or gets the chance to reproduce. As in nature the reproduction probability should correlate positively with the fitness of the individual. In our case we represent the fitness $f$ in form of a slightly modified version of the overall performance $p$:

$$ f(p) = \frac{1}{1 + \exp(-8(p - 0.6))} $$  \hspace{1cm} (1)

The reason why we do not use $p$ directly as a fitness measure bases on the observation that values for $p$ lie more or less in the

---

**Table 1:** Free and fixed parameters of each individual

<table>
<thead>
<tr>
<th>Parameter type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of MFCCs</td>
<td>2 - 12</td>
</tr>
<tr>
<td>MFCC type</td>
<td>0,0E,0DA,E,ED,EDA</td>
</tr>
<tr>
<td>Number of HMM states</td>
<td>1 - 8</td>
</tr>
<tr>
<td>Number of training iterations</td>
<td>2 - 12</td>
</tr>
<tr>
<td>Training-, Test-, Evaluation set</td>
<td>1608, 179, 100</td>
</tr>
<tr>
<td>Sampling rate / Hamming window</td>
<td>10ms / 25ms</td>
</tr>
<tr>
<td>Number of filter banks / Ceplifter</td>
<td>26 / 22</td>
</tr>
</tbody>
</table>

---

2359
We analyzed the performance of 5.1. Performance analysis how the test (tp) and the evaluation set’s performance (ep) de-

The function probabilities (fitness values) enlarge and can be used more efficiently. Another advantage is that the fitness of high performing individuals is bounded to appr. 0.95, which suppresses reproduction in 5% of the cases: like in nature a very high fitness should not automatically assure descendants.

Before reproduction starts there are two knock-out criteria, which directly lead to an individual’s death: In order to accelerate the evolution process and also to iteratively increase the performance of the whole population we do not apply a fixed lethal threshold but always remove the unfittest performance of the whole population we do not apply a fixed lethal threshold. The function now being joy independent of his fitness. This helps to maintain the diversity of the population and avoids that within the population identical clones can accumulate and wipe out competing individuals with comparable fitness.

Now, the population is increased to its original size by randomly selecting individuals, which are allowed to reproduce themselves with the probability $f(p)$ according to Equation 1. Such an individual produces a mutant of itself by randomly changing values of some of its free parameters. How many parameters are allowed to change is described by the mutation rate which correlates negatively with the overall performance $p$, i.e. the higher $p$ the lower the mutation rate and vice versa. In our case we applied fixed intervals to determine the mutation rate for a given $p$ (compare Table 2).

The new individuals are trained and tested, while after that the whole population is evaluated again. This procedure was repeated 25 cycles, which appeared to be enough as we found no obvious improvement after 25 cycles. For each cycle and individual all parameters and performances of both the test and the evaluation set were stored in a log-file for later analysis.

5. Results

In order to show that our approach is able to find optimal parameter sets we compared it to a manually calibrated baseline system. This uses all 12 MFCCs, the zero coefficient and all corresponding delta and acceleration values. The HMMs provide 3 states and are trained 5 iterations. Applying a 90-10 cross validation, we ended up with average performances for anger (52.8%), helplessness (71.6%), neutral (77.3%) and joy (58.4%). The overall average performance is 65.4%. Detailed results are presented in [5].

5.1. Performance analysis

We analyzed the performance of 10 generated and fully evolved populations with respect to the baseline system. Figure 1 shows how the test (tp) and the evaluation set’s performance (ep) develop during 25 cycles of evolution. Each point represents the average of the overall performances of the complete population in the corresponding cycle. Further for both sets also the average overall performances of the 10 fittest individuals is plotted. One can see that ep and tp start little beneath the baseline (black dots) at appr. 64%. As expected the tp increases quite fast during the first cycles, while ep increases more slowly and crosses the baseline in the 5th cycle. The absolute increase in performance is about 4.5% in case of ep and 8.5% in case of tp. Due to a quite huge population with respect to the size of freely evolvable parameters, already the initial population provides quite well performing individuals. But still, their initial ep of appr. 71.0% could be increased by about 2.8% during the evolution process.

![Figure 1: Development of the average recognition rates on the evaluation and test set compared to the baseline](image)

Figure 2 visualizes the behavior of the single emotion’s performances over time. While helplessness and neutral start appr. 5% worse than their baselines and reach them more or less after the evolution stops, anger and joy both start above their baselines. Anger is still able to improve its performance by appr. 5% while the one of joy shows no improvement and oscillates weakly at appr. 65%. This can be explained since the emotion joy is quite underrepresented in the material.

![Figure 2: Development of the single emotion’s recognition rates during the evolution compared to the baselines](image)

5.2. Feature analysis

Since we not only intend to find a well performing emotion recognizer we are mainly interested in its parameters which cause this outstanding performance. For this reason we evaluated the distribution of the free parameters within the population during the evolution in order to present an optimal feature set.
Figure 3 shows the distribution of the different MFCC type characteristics (0, 0D, 0DA, E, ED, EDA) every 5 cycles. One sees that individuals equipped with the energy term E increase from approx. 55% to 71% while the zero coefficient decreases to 30%. Not separately illustrated is the distribution with respect to the single delta and acceleration values. Here ED and EDA outperformed the rest while 0DA (used in the baseline system) was on the third rank.

Figure 3: Percentage distribution of different MFCC type characteristics with respect to MFCC-0 and MFCC-E within the population over time

Figure 4 visualizes how the different numbers of HMM states are represented at the end of evolution. While we used 3 states in our baseline system, already 2 appear sufficient. In general 2-5 states can be recommended.

Figure 4: Percentage distribution of different HMM-state characteristics within the population during the last 5 cycles

The question how many MFCCs to use is answered by Figure 5. Well performing are at least the first 9 MFCCs while in some cases also only the first 6 were sufficient. The distribution of the different training iterations within the final population is qualitatively quite similar to the distribution of the MFCCs visualized in Figure 5. At least 6 iterations should be performed, while more do not consequently lead to an overfitted model. Most of the individuals provide 6-8 training iterations.

Figure 5: Percentage distribution of different MFCC coefficient characteristics within the population during the last 5 cycles

spontaneous material of SmartKom’s public setting we suggest a classifier using MFCC-ED/EDA for the first 6-12 coefficients. HMM models with 2-5 states trained at least 6 iterations result in a robust high performing classifier. We also showed that this model is sparse compared with the requirements in ASR: by limiting a recognizer to MFCC-ED and 2 HMM states one still is able to produce good results, but has to exploit fewer features than for a common ASR system. In future research we want to investigate single MFCCs and their influence on the recognition of the single emotions, i.e., examine if there exist individual optimal feature combinations for each emotion.

7. Acknowledgements

We acknowledge continued support within the Neuroscience Project “Neurobiologically Inspired, Multimodal Intention Recognition for Technical Communication Systems (UC4)” funded by the European Union via the Center for Behavioral Brain Sciences, Magdeburg. This work was further supported by the Transregional Collaborative Research Centre SFB/TRR 62 “Companion-Technology for Cognitive Technical Systems” funded by the German Research Foundation (DFG).

8. References


