A HYBRID SPEECH RECOGNIZER COMBINING HMMS AND POLYNOMIAL CLASSIFICATION

Franz Kummert, Gernot A. Fink, Gerhard Sagerer
Bielefeld University, Faculty of Technology,
33594 Bielefeld, Germany
e-Mail: {franz, gernot, sagerer}@techfak.uni-bielefeld.de

ABSTRACT
In this paper, we present a hybrid speech recognizer combining Hidden Markov Models (HMMs) and a polynomial classifier. In our approach the emission probabilities are not modeled as a mixture of Gaussians but are calculated by the polynomial classifier. However, we do not apply the classifier directly to the feature vector but we make use of the density values of Gaussians clustering the feature space. That means we model the emission probability as a polynomial of Gaussian distributions of n-th degree. As most of these density values are approximately zero for a single feature vector the calculation of a polynomial can be done very efficiently. The usefulness of this hybrid approach was successfully tested on a large conversational speech recognition task.

1. INTRODUCTION
Over the last years, many hybrid approaches for speech recognition have been developed. Most of the systems use Artificial Neural Networks (ANN) to generate emission probabilities of an explicitly or implicitly underlying Hidden Markov Model (see e.g. [1] for an overview). Although polynomial classifiers have been successfully used in pattern recognition for many years [9, 7] there are only a few approaches integrating this type of classifier into HMMs (see e.g. [5, 2]). One reason may be that the polynomial classifier is a Bayes-classifier which implicitly takes into account the prior probability of a class, whereas the Viterbi algorithm uses only the maximum likelihood of a class. In our approach we overcome this problem by weighting every feature vector by the inverse prior probability. In that way the polynomial classifier can directly be used to estimate the emission probabilities of a HMM.

The paper is organized as follows. In Section 2, we briefly introduce the polynomial classifier. Section 3 outlines our hybrid approach combining that classifier with HMMs. Results on the VERBMOBIL scheduling domain are then presented in Section 4, and finally, a conclusion is given.

2. THE POLYNOMIAL CLASSIFIER
Usually, the goal of a classification system is the mapping of a feature vector \( \vec{c} \) into one of \( K \) classes \( \omega_k \). The polynomial classifier approximates the perfect classification functions

\[
\delta_k(\vec{c}) = \begin{cases} 
1 & \text{if } \vec{c} \in \omega_k \\
0 & \text{otherwise}
\end{cases}, \quad k = 1, \ldots, K
\]

by a polynomial in the coefficients of the feature vector. Let

\[
\vec{x}(\vec{c}) = (1, c_1, \ldots, c_n, c_1^2, c_1 c_2, \ldots, c_n^2, c_n^3, \ldots)^T
\]

denote the expanded feature vector then the estimated classification function for class \( \omega_k \) can be written as:

\[
d_k(\vec{c}) = a_{0,0} + a_{k,1} c_1 + \ldots + a_{k,n} c_n + a_{k,n+1} c_1^2 + \ldots + a_{k,2n+n/2} c_n^2 + \ldots + a_{k,2n+n} c_n^3 + \ldots
\]

\[
= \bar{a}_k^T \vec{x}(\vec{c}), \quad k = 1, \ldots, K
\]

The combination of the functions \( d_k(\vec{c}) \) to a vector \( \vec{d}(\vec{c}) \) and of the vectors \( \bar{a}_k \) to a matrix \( \bar{A} \) leads to:

\[
\vec{d}(\vec{c}) = \bar{A}^T \vec{x}(\vec{c}) \quad (1)
\]
The optimal parameter matrix $\mathbf{A}^*$ is calculated on the basis of a classified training sample by minimizing the mean squared error between the perfect and the estimated classification functions:

$$\mathbf{A}^* = \arg \min_{\mathbf{A}} E\{(\hat{\delta}(\mathbf{c}) - \mathbf{A}^T \mathbf{x}(\mathbf{c}))^2\}$$

Assuming we have a classified training sample of $N$ elements, i.e. for every feature vector $j \mathbf{c}$, $j = 1, \ldots, N$ we know the perfect classification functions $\hat{\delta}(j \mathbf{c})$, the optimal solution results from:

$$\mathbf{A}^* = \left( \frac{1}{N} \sum_{j=1}^{N} \mathbf{x}(j \mathbf{c}) \mathbf{x}(j \mathbf{c})^T \right)^{-1} \left( \frac{1}{N} \sum_{j=1}^{N} \mathbf{x}(j \mathbf{c}) \hat{\delta}(j \mathbf{c})^T \right)$$  \hspace{1cm} (2)

If the inverse of the first matrix in equation (2) does not exist due to linear dependent components of $\mathbf{x}(\mathbf{c})$ the training algorithm of the polynomial classifier detects these components and reduces the expanded feature vectors accordingly. Thus in every case an optimal solution to the minimization problem can be calculated.

According to the Weierstrass Theorem, arbitrary functions can be approximated in such a way where the accuracy only depends on the degree of the polynomial.

For details of the polynomial classifier and for hints for an efficient computation see e.g. [8].

### 3. THE HYBRID APPROACH

For our hybrid approach we use semi-continuous HMMs with Gaussian distributions [4]. But instead of modeling the emission probability $b_j(o_t)$ for state $S_j$ as a mixture of $L$ Gaussians, i.e.

$$b_j(o_t) = \sum_{l=1}^{L} c_{jl} \mathcal{N}(o_t; \mu_l, K_l)$$

with $\sum_{l=1}^{L} c_{jl} = 1$ and $0 \leq c_{jl} \leq 1$

we model $b_j(o_t)$ as a polynomial of Gaussians distributions of $n$-th degree. Due to space limitations we do not calculate the complete polynomial by omitting the mixed products of Gaussians, i.e. the emission probability is a constant factor plus a mixture of $L$ Gaussians plus a mixture of $L$ squared Gaussians and so on, i.e.

$$b_j(o_t) = c_{j0} + \sum_{l=1}^{L} c_{jl} \mathcal{N}(o_t; \mu_l, K_l) + \sum_{l=1}^{L} c_{jL+l} \mathcal{N}(o_t; \mu_l, K_l)^2 + \sum_{l=1}^{L} c_{j2L+l} \mathcal{N}(o_t; \mu_l, K_l)^3 + \ldots$$

with $c_{jl} \in \mathbb{R}$

The coefficients $c_{jl}$ are any real numbers which usually do not sum up to 1.

The training of our hybrid recognition system consists of the following two steps:

1. Training of a classical semi-continuous speech recognizer using the Baum-Welch procedure where both the codebook and the mixtures are optimized.

2. The trained system of step 1 is used to estimate the perfect classification functions $\hat{\delta}$ for the training sample allowing the calculation of the matrix $\mathbf{A}^*$ of equation (2).

This matrix $\mathbf{A}^*$ is then used by the hybrid recognition system to calculate the emission probabilities based on equation (1).

Typically, the polynomial classifier tries to minimize the mean-squared error between the estimation function of the classifier (here the vector of emission probabilities for every HMM-state) and the perfect decision function which has value 1 in the i-th component if the feature vector belongs to the i-th class and has value 0 for the other $K - 1$ components. In our approach we use a modified perfect decision function reflecting the state-occupation probabilities $\gamma(i) = P(s_t = S_i | \mathbf{Q}, \lambda)$ of an ordinary semi-continuous HMM with $K$ states. i.e.

$$\hat{\delta}(\mathbf{c}) = \left( \begin{array}{c} \gamma(1) \\ \vdots \\ \gamma(K) \end{array} \right) \text{ with } \mathbf{c'} = \left( \begin{array}{c} \mathcal{N}(o_t; \mu_1, K_1) \\ \vdots \\ \mathcal{N}(o_t; \mu_L, K_L) \end{array} \right)$$

The probabilities $\gamma(i)$ are calculated during the Baum-Welch training in step 2. Thus our hybrid ap-
proach can take advantage of the Baum-Welch training which is usually better than Viterbi training.

As mentioned in the introduction the polynomial classifier implicitly takes into account the prior probability of a class which is not compatible to the Viterbi-decoding of HMMs. In order to overcome this problem we weight the feature vectors by the prior probabilities. Doing this, equation (2) modifies to:

\[
\mathbf{\Delta}^* = \left( \frac{1}{\sum_{j=1}^{N} w(j) \mathbf{x}(j) \mathbf{x}(j)^T} \right)^{-1} \left( \frac{1}{\sum_{j=1}^{N} \mathbf{x}(j) \delta(j) \mathbf{x}(j)^T} \right)
\]

(3)

with \( \delta(j) = \left( \frac{\gamma(1)}{P(S_1)} \right) \ldots \left( \frac{\gamma(K)}{P(S_K)} \right) \)

and \( w(j) = \sum_{i=1}^{K} \gamma_i(i) / P(S_i) \)

In that way the polynomial classifier can directly be used to estimate the emission probabilities of a HMM using the soft decisions of the Baum-Welch training.

4. Results

All experiments were carried out in the ESMERALDA framework — a powerful development environment for statistical pattern recognition systems [3]. The emission probability modelling applying a polynomial classifier was implemented as an extension of the general methods for handling HMMs using mixtures of Gaussian densities.

The conversational speech recognition task was taken from the domain of appointment scheduling investigated in the German VERBMOBIL project [10]. In the configuration defined by the official 1996 evaluation guidelines over 33 hours of spontaneous human-human dialogues carried out by more than 650 speakers were available for training acoustic models. Testing is performed on approximately 41 minutes of speech from 29 different speakers using a 5335 word open vocabulary.

The baseline system set up for our experiments uses a 39-dimensional feature space. On 16 ms speech frames every 10 ms feature vectors consisting of 12 mel-frequency cepstral coefficients with adaptive mean normalisation and one energy coefficient are calculated. Additionally, the first and second order time derivatives are computed using regression polynomials within a 5 frame time window.

The single codebook shared by the semi-continuous HMMs consists of 1009 Gaussian distributions with full covariance matrices. It was initialized using an unsupervised vector quantisation approach and then jointly reestimated together with the remaining HMM parameters during Baum-Welch training.

The acoustic models for the words of the lexicon, some spontaneous speech effects, and some noise models mostly use triphone subword units. Only lexicon entries occurring more than 50 times in the training material are assigned separate whole-word models⁴. In order to be able to robustly train all required triphones without explicitly applying some generalisation technique an entropy based state clustering is carried out after the first parameter reestimation step.

During recognition the official bi-gram language model provided by Philips Research Labs, Aachen, Germany, was used which achieves a test-set perplexity of 64. Acoustic and language model restrictions are applied strictly synchronously during one-pass decoding. Without using any speaker adaptation techniques this baseline configuration achieves a word accuracy of 79.45% — a figure which is among the best results obtained on this task under comparable conditions.

<table>
<thead>
<tr>
<th>system config.</th>
<th>WA</th>
<th>Δ WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline system</td>
<td>79.45%</td>
<td>—</td>
</tr>
<tr>
<td>linear approach</td>
<td>79.91%</td>
<td>2%</td>
</tr>
<tr>
<td>quadratic approach</td>
<td>81.47%</td>
<td>10%</td>
</tr>
<tr>
<td>cubic approach</td>
<td>81.57%</td>
<td>10%</td>
</tr>
</tbody>
</table>

Table 1: Evaluation results of the hybrid approach

In principle, the linear approach is identical to the ordinary semi-continuous approach except the fact that

⁴ Though word models set up on top of a global pool of triphone models give slightly better results, the separate modelling makes estimating the parameters of the polynomial classifier easier.
the mixture weights $c_{ji}$ can be any real number. As is could be expected this additional degree of freedom only slightly increases the word accuracy to 79.91%. This linear approach is closely related to Radial Basis Functions (see e.g. [6]) where the distances to some prototypes are weighted by a kernel function and the estimated classification function $d_k(e)$ for class $\omega_k$ is a linear combination of these weighted distances. If the prototypes are the mean values and the kernel functions are the $L$ Gaussians of the codebook then both approaches are identical.

The quadratic approach reaches a word accuracy of about 81.5% an improvement which is statistically significant at a confidence level of 95%. A further expansion by the cubic Gaussians does not lead to a significant improvement of the word accuracy. Therefore, our hybrid approach achieves a relative improvement of the word error rate (WER) of about 10% which demonstrates impressively the usefulness of the polynomial classifier for speech recognition tasks.

5. CONCLUSION

In this paper, we presented a hybrid approach for speech recognizers combining Hidden Markov Models and a polynomial classifier. The emission probabilities are modeled as a polynomial of Gaussian distributions of 2nd degree. To compensate the prior probabilities used by the polynomial classification scheme we weighted every feature vector by the inverse prior probability. So a maximum likelihood classifier is achieved. Additionally, the perfect decision function was modified reflecting the state-occupation probabilities of a semi-continuous HMM during Baum-Welch training.

Our hybrid approach was successfully tested on the German VERBMOBIL appointment scheduling domain. Although our baseline system obtains a result which is among the best results published on this task (using the official bigram and no adaptation) the hybrid approach reaches a significantly improved word accuracy.

6. REFERENCES


