A COMPUTATIONALLY SCALABLE SPEAKER RECOGNITION SYSTEM

W. M. Campbell and C. C. Broun

Motorola Human Interface Laboratory
Tempe, AZ 85284, USA

ABSTRACT

Computationally scalable speaker recognition systems are highly desirable in practice. To achieve this objective, we use a two-stage architecture for text-prompted speaker recognition. In this system, the input speech is first segmented on subword boundaries using a Viterbi alignment. The second stage applies a polynomial classifier to each subword for verification. Through a simple approximation, the scoring criterion for the polynomial classifier is made highly scalable. The resulting combination of speaker independent segmentation and a scalable recognition system results in a system which can perform speaker recognition on a large population with minimal computation.

1. INTRODUCTION

Speaker recognition (verification or identification) has many modes of interface–text-prompted, text-independent, or text-dependent. In this paper, we work with the text-prompted case. In this situation, the claimant is asked to repeat a prompted text. Utterance verification is then performed to verify the text has been correctly read. The system then identifies or verifies the identity of the speaker using the knowledge of the prompted text. In the case of (closed-set) identification, the speaker is identified from a list of speakers. In the case of verification, the speaker is either accepted or rejected as the claimed individual.

Popular methods for text-prompted recognition include Gaussian Mixture Models (GMM's), HMM's, Neural Tree Networks, and vector quantization. HMM's are the closest approach to ours. HMM models are constructed from knowledge of the subwords of enrollment data. Recognition is performed by concatenating HMM models for subwords of the prompted input [1, 2]. Typically, cohort normalization [3, 4] is applied to approximate the ideal Bayes decision rule.

We extend a novel approach presented in [5]. This approach uses a polynomial discriminant function. The advantage of this method is severalfold. First, the method is able to handle large amounts of enrollment data with ease. For speaker verification enrollment, the entire anti-speaker population is encapsulated into a single vector; this vector is fixed size (with respect to the amount of anti-speaker data). Second, the polynomial method is discriminative. It directly approximates the *a posteriori* probabilities and finds the global minimum. This eliminates the need for cohort normalization and selection. Finally, the training and recognition algorithms are simple multiply-add architectures which fit well with modern DSP implementations.

The outline of the paper is as follows. In Section 2, we introduce our scoring method. The scoring method is a novel combination of connectionist speech recognition scoring methods, a simple approximation, and simplification. Section 3 shows how to train a polynomial classifier to obtain the emission probabilities needed for scoring. Section 4 applies the method to the YOHO database. We show that this method outperforms recent methods in the literature in accuracy, parameter usage, and computation.

2. SCORING

2.1. Polynomial Classifiers

We use a polynomial classifier based upon a linear combination of monomials. The classifier output, $f(\mathbf{x})$ can be expressed as

$$f(\mathbf{x}) = \mathbf{w}^t \mathbf{p}(\mathbf{x}), \tag{1}$$

where $\mathbf{p}(\mathbf{x})$ is the vector of all monomials of degree K or less of the components of $\mathbf{x}$. Note that we use a bold $\mathbf{p}$ for polynomials to distinguish from probability functions such as $p(x)$. As an example of a polynomial function, let $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^t$ and $K = 2$, then

$$\mathbf{p}(\mathbf{x}) = \begin{bmatrix} 1 & x_1 & x_2 & x_1^2 & x_1 x_2 & x_2^2 \end{bmatrix}^t. \tag{2}$$

The vector $\mathbf{w}$ is a vector of coefficients representing the classifier model. We train the classifier to approximate *a posteriori* probabilities.

2.2. Sequence Scoring

For speaker recognition an input utterance is converted to a sequence of feature vectors, $\mathbf{x}_1, \ldots, \mathbf{x}_n$ by extraction of

spectral characteristics. We assume that speaker j is modeled by a concatenation of hybrid HMM/polynomial classifier models, ω_j, corresponding to the prompted phrase. That is, we use polynomials to model the *emission probabilities* of the HMM. We use an optimum Bayes approach to recognition. We first calculate $p(\mathbf{x}_1, \ldots, \mathbf{x}_n|\omega_j)$. We abbreviate this as $p(\mathbf{x}_1^n|\omega_j)$.

In order to calculate $p(\mathbf{x}_1^n|\omega_j)$, we use several assumptions. A standard approach is to express this value as a sum over all possible state sequences for an HMM. We approximate this by using the most likely state sequence, $q_1, \ldots, q_n$. We assume that this state sequence can be derived independently of the speaker. That is, a Viterbi alignment using speaker independent speech recognition gives a sequence which has $p(\mathbf{x}_1^n, q_1^n|\omega_j)$ close to the optimal speaker dependent sequence value. Further by assuming independence and that the emission probability is dependent only on the current state, we obtain

$$\begin{aligned} p(\mathbf{x}_1^n|\omega_j) &\approx p(\mathbf{x}_1^n, q_1^n|\omega_j) \\ &= p(\mathbf{x}_1^n|q_1^n, \omega_j)p(q_1^n|\omega_j) \\ &= \left(\prod_{i=1}^n p(\mathbf{x}_i|q_1^n, \omega_j)\right) p(q_1^n|\omega_j) \qquad (3) \\ &= \left(\prod_{i=1}^n p(\mathbf{x}_i|q_i, \omega_j)\right) p(q_1^n|\omega_j) \end{aligned}$$

We discard the second term on the right hand side of (3) in our discriminant function, since the probability of the state sequence in our application is negligible compared to the probability of the observations.

We now use the relation

$$p(\mathbf{x}_i|q_i, \omega_j) = \frac{p(\omega_j|\mathbf{x}_i, q_i)p(\mathbf{x}_i, q_i)}{p(\omega_j|q_i)p(q_i)} \qquad (4)$$

and (3) to obtain the discriminant function

$$d'(\mathbf{x}_1^n, j) = \prod_{i=1}^n \frac{p(\omega_j|\mathbf{x}_i, q_i)}{p(\omega_j|q_i)}. \qquad (5)$$

We have discarded the numerator term $\prod_{i=1}^n p(\mathbf{x}_i, q_i)$ and the denominator term $\prod_{i=1}^n p(q_i)$, because they are independent of ω_j; i.e., they will cancel out in the likelihood ratio function.

We now perform two simplifications. First, we consider the logarithm of the discriminant function,

$$\log(d'(\mathbf{x}_1^n, j)) = \sum_{i=1}^n \log\left(\frac{p(\omega_j|\mathbf{x}_i, q_i)}{p(\omega_j|q_i)}\right). \qquad (6)$$

Using Taylor series, a linear approximation of $\log(x)$ around $x = 1$ is $x - 1$. Thus, we can approximate $\log(d'(\mathbf{x}_1^n, j))$ as

$$\log(d'(\mathbf{x}_1^n, j)) \approx \sum_{i=1}^n \left(\frac{p(\omega_j|\mathbf{x}_i, q_i)}{p(\omega_j|q_i)} - 1\right). \qquad (7)$$

The approximation (7) is especially good when the scaled *a posteriori* is near 1. Since $x - 1$ goes to -1 as x goes to 0 and $\log(x)$ goes to $-\infty$, using $x-1$ is approximately equivalent to replacing $\log(x)$ by $\max(\log(x), -1)$. The approximation $\max(\log(x), -1)$ is equivalent to ensuring that the probability is not allowed to go below a certain value. The discriminant function with all of the above approximations is

$$d(\mathbf{x}_1^n, j) = \sum_{i=1}^n \frac{p(\omega_j|\mathbf{x}_i, q_i)}{p(\omega_j|q_i)} \qquad (8)$$

where we have dropped the -1 since a constant offset will be eliminated in a log likelihood ratio function. The approximation (8) is the basis for our scalable scoring technique.

As a second simplification, we assume that the states in the discriminant function (8) are from a left-to-right HMM model with l states. Then the state sequence can be combined into similar groups. I.e., assume we have indices, i_k, j_k that partition $1, \ldots, n$ as $i_1 = 1 < j_1 < i_2 = j_1 + 1 < j_2 < \cdots < i_l = j_{l-1} + 1 < j_l = n$, so that $q_{i_k} = \cdots = q_{j_k}$. If we approximate each $p(\omega_j|\mathbf{x}_i, q_i)$ using a unique polynomial classifier with model $\mathbf{w}_{j,q_i}$ for every unique state and every speaker, then we obtain the following simplification

$$d(\mathbf{x}_1^n, j) = \sum_{k=1}^l c_{j,k}\mathbf{w}_{j,k}^t \left[\sum_{m=i_k}^{j_k} \mathbf{p}(\mathbf{x}_m)\right] \qquad (9)$$

where $c_{j,k} = 1/p(\omega_j|q_{i_k})$. We let

$$\bar{\mathbf{p}}_k = \sum_{m=i_k}^{j_k} \mathbf{p}(\mathbf{x}_m), \qquad (10)$$

then (9) becomes

$$d(\mathbf{x}_1^n, j) = \sum_{k=1}^l c_{j,k}\mathbf{w}_{j,k}^t \bar{\mathbf{p}}_k. \qquad (11)$$

The equation (11) gives a computationally scalable system. For example, if we segment the input utterance into six sections, we can calculate $\bar{\mathbf{p}}_k$ for each of these sections *independent of the speaker*. Then scoring for each speaker is accomplished by six inner products and a sum of six scores–a low-complexity operation.

For optimal Bayes identification with equal class priors, we choose j^* so that $j^* = \text{argmax}_j\, d(\mathbf{x}_1^n, j)$. For optimal Bayes verification, we compare the log likelihood ratio to a threshold. For our discriminant function, this corresponds to a threshold on $d(\mathbf{x}_1^n, 0) - d(\mathbf{x}_1^n, 1)$. Since $\mathbf{w}_{1,k}^t \bar{\mathbf{p}}_k = 1 - \mathbf{w}_{0,k}^t \bar{\mathbf{p}}_k$ by virtue of the training method, see [5, 6], the decision rule is based on a threshold for

$$\sum_{k=1}^l (c_{0,k} + c_{1,k})\mathbf{w}_{0,k}^t \bar{\mathbf{p}}_k - \sum_{k=1}^l c_{1,k}. \qquad (12)$$

That is, we can calculate a weighted sum of $\mathbf{w}_{0,k}^t \bar{\mathbf{p}}_k$ as in (11) and compare this to a threshold (i.e., we do not have to explicitly calculate a cohort score).

As a final point, we note that our scoring method extends to any linear approximation space. The key property that allows simplification of the scoring is that the quantity $\sum_{m=i_k}^{j_k} \mathbf{w}_{j,k}^t \mathbf{p}(\mathbf{x}_m)$ can be written as $\mathbf{w}_{j,k}^t (\sum_{m=i_k}^{j_k} \mathbf{p}(\mathbf{x}_m))$. This property is dependent only on the fact that we are using a linear basis, i.e., in our case, the components of $\mathbf{p}(\mathbf{x}_m)$.

3. TRAINING

To use the scoring method (8), we must find a polynomial function which approximates the quantity

$$\frac{p(\omega_j|q_i, \mathbf{x}_i)}{p(\omega_j|q_i)}. \tag{13}$$

We do this by training a polynomial classifier using a mean-squared error criterion. For simplicity, we consider the two-class problem (verification) for the remainder of this section. We assume that the speaker is ω_0 and the impostor set is ω_1.

We first consider the case when a one-state HMM/polynomial is used. For this case, the quantity (13) simplifies to

$$\frac{p(\omega_j|\mathbf{x}_i)}{p(\omega_j)}. \tag{14}$$

Since $p(\omega_1|x) = 1 - p(\omega_0|x)$, we consider only approximating $p(\omega_0|x)$. We can obtain a polynomial approximation of this quantity by solving the following optimization problem

$$\underset{\mathbf{w}}{\operatorname{argmin}} E\left\{ \left(\mathbf{w}^t \mathbf{p}(\mathbf{x}) - y(\omega)\right)^2 \right\}. \tag{15}$$

Here $y(\omega)$ is the ideal discrimination output; i.e., $y(\omega_0) = 1$ and $y(\omega_1) = 0$. A method which solves this optimization problem in a novel way is given in [5]. The result of (15) is a model such that the polynomial, $f(\mathbf{x}) = \mathbf{w}^t p(\mathbf{x})$, approximates $p(\omega_0|x)$. To obtain (14), we divide out the prior, $p(\omega_0)$, obtained from the *training* data set. This operation corresponds to scaling the model vector, $\mathbf{w}$, by $1/p(\omega_0)$.

An alternative approach to dividing out the prior, $p(\omega_0)$, is to incorporate the desired prior into the training process. If we set $p(\omega_0) = p(\omega_1) = 1/2$, then we can ignore the prior in (8); that is, the prior will cancel out of the likelihood ratio. For this style of training, we can write the expectation in (15) as

$$\begin{aligned} E\left\{ \left(\mathbf{w}^t \mathbf{p}(\mathbf{x}) - y(\omega)\right)^2 \right\} = \\ p(\omega_0) E\left\{ \left(\mathbf{w}^t \mathbf{p}(\mathbf{x}) - 1\right)^2 | \omega = \omega_0 \right\} + \\ p(\omega_1) E\left\{ \left(\mathbf{w}^t \mathbf{p}(\mathbf{x})\right)^2 | \omega = \omega_1 \right\}. \end{aligned} \tag{16}$$

The criterion for training then becomes

$$\begin{aligned} \underset{\mathbf{w}}{\operatorname{argmin}} \Bigg[\frac{p(\omega_0)}{N_0} \sum_{j=1}^{N_0} |\mathbf{w}^t \mathbf{p}(\mathbf{x}_{0,j}) - 1|^2 + \\ \frac{p(\omega_1)}{N_1} \sum_{j=1}^{N_1} |\mathbf{w}^t \mathbf{p}(\mathbf{x}_{1,j})|^2 \Bigg] \end{aligned} \tag{17}$$

where $\mathbf{x}_{i,j}$ is the jth training sample from class ω_i, and N_i is the number of training samples in class ω_i. Training with equal priors typically produces better results than scaling the model by training set priors [7].

The extension to the multistate case is straightforward. We illustrate with an example. Suppose we have two utterances "23-45" and "45-23" prompted during training for a speaker and an anti-speaker population. We first segment all utterances into the vectors corresponding to the word "23" and the vectors corresponding to the word "45" (using words as our fundamental states). We then train a polynomial model, $\mathbf{w}_{23}$, to distinguish between the speaker and anti-speaker set only on the vectors corresponding to word "23" using the two class training criterion (17). We train a similar model, $\mathbf{w}_{45}$, for "45." These models approximate $p(\omega_0|\mathbf{x}, q = 23)$ and $p(\omega_0|\mathbf{x}, q = 45)$, respectively, where $q = 23$ means the state in the HMM corresponding to the word "23" (and similarly for $q = 45$). We can then use the resulting models in the scoring equation (8).

4. RESULTS

We applied our method to the YOHO database. The YOHO database [8] is a large (138 speaker) multisession database designed for testing speaker recognition systems. Enrollment and verification consists of combination lock phrases; e.g., "26-81-57." Enrollment consists of 4 sessions of 24 phrases. Recognition consists of 10 sessions of 4 phrases per session per speaker. We perform recognition both on the 40 phrases (a 1-phrase test) and on the combination of the 4 phrases in each session (a 4-phrase test).

For Viterbi segmentation, we designed models based on 12 MFCC's and 12 Δ-MFCC's. We segmented utterances based on subwords for each of the decades $20, 30, \ldots, 90$, and the digits $1, \ldots, 9$. The choice of these subwords, rather than the more common monophones, was motivated by two factors. First, there are 20 monophone models needed to model the words in YOHO. Our selection reduces the number of models to 16. Note that we cannot use models for individual numbers, e.g. 23, because some numbers in recognition are not represented in enrollment. Second, segmenting with decades and digits gives fewer states per utterance. For the 1-phrase test, there are only 6 states. With fewer states, the number of frames per segment increases. This property increases the scalability of the system.

Table 1: Verification performance on the YOHO database.

MFCC	Δ	$\Delta\Delta$	order	Avg. EER 1-phrase %	Avg. EER 4-phrase %
12	-	-	3	0.35	0.02
12	12	-	2	0.29	0.08
12	12	-	3	0.07	0.01
12	12	12	2	0.24	0.09

Table 2: Identification performance on the YOHO database.

MFCC	Δ	$\Delta\Delta$	order	Error 1-phrase %	Error 4-phrase %
12	-	-	3	0.74	0.14
12	12	-	2	0.51	0.07
12	12	-	3	0.14	0.07
12	12	12	2	0.38	0.07

We trained a system to approximate the *a posteriori* probabilities, $p(\omega_j|x_i, q_i)$, with a polynomial classifier. We preprocessed the data using preemphasis and a Hamming window. We extracted 12 MFCC's, 12 Δ-MFCC's, and 12 $\Delta\Delta$-MFCC's for every 30 ms frame with an overlap of 20 ms. The results of verification are shown in Table 1. The results for identification are shown in Table 2.

Our results compare very favorably with those in the literature. Our best 1-phrase average EER is 0.07% and our best 1-phrase identification error rate is 0.14%. For comparison, in the literature, 1-phrase average EER rates of 0.62% [2] and 1.07% [9] have been reported. Identification error rates of 0.56% [2] and 1.74% [9] have also been produced. We note that the comparison is not entirely equivalent, since different methods are used in each cited reference.

If we compare the parameter-usage performance of our system, then the systems reported in Table 2 use 7280, 5200, 46800, and 11248 parameters, respectively. We estimate the parameter usage of [2] (for example) for the 3-mixture case with 20 monophone models to be about 14220 parameters. Thus, if we consider the second system in Table 2, we have a lower error rate and use 63% less parameters.

For a computationally scalable identification system, we want the computation growth with the number of speakers, $cN_{\rm spk}$, to have as small of c as possible. If we scored each speaker's utterance directly using a polynomial model, then the computation required is approximately

$$N_{\rm frames}\left(3N_{\rm coeff}+1\right)N_{\rm spk} \tag{18}$$

where $N_{\rm coeff}$ is the number of coefficients in the model and $N_{\rm frames}$ is the number of frames in the utterance. Using our new approach, the computation is approximately

$$2N_{\rm coeff}N_{\rm frames}+(2N_{\rm coeff}+1)N_{\rm states}N_{\rm spk} \tag{19}$$

where $N_{\rm states}$ is the number of unique states in the model. Asymptotically as the number of speakers increases, our computation savings is about $1.5N_{\rm frames}/N_{\rm states}$. Our implementation computation speedup is $240/6 = 40$ for a 1-phrase test (approximately 2.4 seconds of speech per test).

5. CONCLUSIONS

We derived a novel scoring method for a HMM with polynomial emission probabilities that produces a computationally scalable speaker recognition system. Results showed high accuracy and significant computation reduction.

6. REFERENCES

[1] T. Matsui and S. Furui, "Concatenated phoneme models for text-variable speaker recognition," in *International Conference on Acoustics Speech and Signal Processing*, vol. II, pp. 391–394, 1993.

[2] C. Che and Q. Lin, "Speaker recognition using HMM with experiments on the YOHO database," in *Proc. Eurospeech*, pp. 625–628, 1995.

[3] A. Higgins, L. Bahler, and J. Porter, "Speaker verification using randomized phrase prompting," *Digital Signal Processing*, vol. 1, pp. 89–106, 1991.

[4] A. E. Rosenberg, J. DeLong, C.-H. Lee, B.-H. Juang, and F. K. Soong, "The use of cohort normalized scores for speaker verification," in *Proceedings of the International Conference on Spoken Language Processing*, pp. 599–602, 1992.

[5] W. M. Campbell and K. T. Assaleh, "Polynomial classifier techniques for speaker verification," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 321–324, 1999.

[6] J. Schürmann, *Pattern Classification.* John Wiley and Sons, Inc., 1996.

[7] K. T. Assaleh and W. M. Campbell, "Speaker identification using a polynomial-based classifier," in *International Symposium on Signal Processing and its Applications*, pp. 115–118, 1999.

[8] J. P. Campbell, Jr., "Testing with the YOHO CD-ROM voice verification corpus," in *Proceedings of the Internation Conference on Acoustics, Speech, and Signal Processing*, pp. 341–344, 1995.

[9] J. Colombi, D. Ruck, S. Rogers, M. Oxley, and T. Anderson, "Cohort selection and word grammar effects for speaker recognition," in *International Conference on Acoustics Speech and Signal Processing*, pp. 85–88, 1996.