AN EFFICIENT APPROXIMATION OF THE FORWARD-BACKWARD ALGORITHM TO DEAL WITH PACKET LOSS, WITH APPLICATIONS TO REMOTE SPEECH RECOGNITION

*Bengt J. Borgström and Abeer Alwan**

Department of Electrical Engineering,
University of California, Los Angeles
jonas, alwan@ee.ucla.edu

ABSTRACT

This paper proposes an efficient approximation of the forward-backward (FB) algorithm, for the purpose of estimating missing features, based on downsampling statistical models. The paper discusses the role of Hidden Markov Models (HMMs) in the estimation process, and presents an approximation to the FB method by developing HMMs based on lower resolution quantizers, which are obtained through a tree-structure mapping of quantizer centroids. To illustrate the effectiveness of the proposed method, we apply it to the problem of error concealment in remote speech recognition, using the Aurora-2 database. The FB approximation provides comparable word recognition accuracy results relative to the standard FB method, while reducing the computational load by a large factor (> 250 in this case).

Index Terms— Forward-Backward Algorithm, Missing Features, Error Concealment, Remote Speech Recognition.

1. INTRODUCTION

Statistical modeling of Markov sources has been extensively studied due to their elegant theoretical framework and applications in signal processing, communications, and other fields [1]. HMMs are used to implement efficient and accurate estimation and recognition systems, and can provide a framework for estimating missing features or corrupted data. The Viterbi and forward-backward (FB) algorithms have been applied to various tasks within automatic speech recognition [1].

Although it has been widely shown to be effective in the estimation of lost features based on an HMM framework, and to outperform the Viterbi algorithm [2], the FB can be restrictive due to its computational load. This constraint is especially true in speech recognition applications, where clients may be distributed and applications may be delay-sensitive.

We propose an efficient approximation of the FB algorithm for the task of missing feature estimation, by means of HMM downsampling. Utilizing a tree-structured quantizer in the estimation process, HMMs can be downsampled, and corresponding statistical parameters can be determined accordingly. Note that HMM downsampling does not refer to decimating the speech signal (i.e. $x\,(Mn)$), but instead operates on the parametric centroids of the HMM. We apply the proposed FB approximation to error concealment (EC) in packet-based Remote Speech Recognition (RSR). The proposed algorithm is shown to greatly reduce the required computational complexity, while providing system performance comparable to the original FB algorithm.

The application of HMMs to missing feature estimation is discussed in Section 2. The proposed approximation to the FB algorithm is presented in Section 3. Section 4 provides experimental results for error concealment in remote speech recognition. Conclusions are given in Section 5.

2. HMMS AND MISSING FEATURE ESTIMATION

Missing features occur when data become missing or corrupted. Examples include the loss of packets in the transmission of data, such as speech, due to channel noise or channel congestion [4], and unreliable spectral coefficients due to acoustic noise [5]. In general, lost features can be approximated either by interpolation or estimation. Interpolation involves calculating unreliable features as a function of past and future reliable data.

Estimation determines lost features based on a model of the underlying signal, as well as on correctly received features. A common method to model the features is with an HMM [3]. Let the quantizer of feature m be represented by Q_m, and let the set of corresponding centroids be referred to as $\left\{c_m^1, c_m^2, \ldots, c_m^N\right\}$, where N is the number of centroids in Q_m. In order to apply estimation methods, separate HMMs are constructed for each of the quantizers Q_m, for $1 \leq m \leq M$, to model the feature trajectories [3].

Let the HMM applied to the output signal from quantizer Q_m be referred to as Λ_m=($\mathbf{A}_m, \mathbf{B}_m, \vec{\pi}_m$), where $\mathbf{A}_m$ provides transitional statistics, $\mathbf{B}_m$ provides observation statistics, and $\vec{\pi}_m$ provides steady-state statistics [1]. The steady-state probabilities can be determined empirically from training data as:

*Supported in part by the NSF and a fellowship to Dr. Abeer Alwan from the Radcliffe Institute for Advanced Study.

$$\vec{\pi}_m(i) = \frac{\text{no. of samples quantized to centroid } c_m^i}{\text{total no. of samples}}, \quad (1)$$

and transitional probabilities of Λ_m are then:

$$\mathbf{A}_m(i,j) = \frac{\text{no. of samples transitioning from } c_m^i \text{ to } c_m^j}{\text{no. of samples quantized to } c_m^i}. \quad (2)$$

Formulation of the components of $\mathbf{B}_m$ is dependent on the specific application of the system. For example, in the case of transmission of data across a noisy channel, $b_i(\mathbf{o})$ may represent the probability that a transmitted binary codeword representing c_m^i is detected as a different codeword $\mathbf{o}$, and the probability distribution can be formulated from theory of a Binary Symmetric Channel (BSC) [2]. In the case of noisy spectrogram data, $b_i(o)$ may represent the probability that a clean feature element quantized to c_m^i is corrupted by acoustic noise to become o, [5]. Without loss of generality, we will refer to the observation probabilities as $b_i(\mathbf{o})$.

Once the underlying signals have been modeled by the set $\{\Lambda_1, \ldots, \Lambda_M\}$, lost features can be approximated by the means of the estimates. As was previously stated, the forward-backward algorithm provides an accurate algorithm for estimating missing or ambiguous data within an HMM framework. Let $\mathbf{f}(n) = [x_1(n), x_2(n), \ldots, x_M(n)]^T$ be the input feature vector at frame n. The estimate of component $x_m(n)$, using the FB algorithm, is determined as [1]:

$$\widehat{x}_m(n) = \sum_{i=1}^{N} c_m^i \gamma_m^i(n), \quad (3)$$

where:

$$\gamma_m^i(n) = \frac{\alpha_m^i(n)\,\beta_m^i(n)}{\sum_{j=1}^{N} \alpha_m^j(n)\,\beta_m^j(n)}. \quad (4)$$

The values $\alpha_m^i(n)$ and $\beta_m^i(n)$ can be determined as:

$$\alpha_m^i(n) = \left[\sum_{j=1}^{N} \mathbf{A}_m(j,i)\,\alpha_m^j(n-1)\right] b_i(\mathbf{o}_n), \quad (5)$$

$$\beta_m^i(n) = \sum_{j=1}^{N} \mathbf{A}_m(i,j)\,\beta_m^j(n+1)\,b_j(\mathbf{o}_{n+1}). \quad (6)$$

The first and last reliable features are known as α_m^i=β_m^i=1.

The computational load introduced by the FB algorithm is of order $O\left(N^2T\right)$, where T is the length of the unreliable region in terms of feature vectors [1]. This could create problems for constrained servers or delay-sensitive applications.

3. EFFICIENT APPROXIMATION OF THE FORWARD-BACKWARD ALGORITHM

We propose an efficient approximation of the forward-backward algorithm based on downsampling of the underlying statistical models. Instead of using the original quantizer centroids to model the signal during feature reconstruction, we use a quantizer with less resolution to build the statistical model. We implement a tree-structure mapping of centroids to allow downsampling of the discrete HMMs by factors of 2, with N=2^R. Note, however, that training of the statistical models is still carried out at the original R-bit resolution.

Let Q_m^R represent a R-bit quantizer for component m, with centroids $\left\{c_m^{R,1}, c_m^{R,2}, \ldots, c_m^{R,2^R}\right\}$. (Note that typically the original quantizer is allocated 8 bits, and thus Q_m^8=Q_m.) The signal model, now referred to as Λ_m^R, has statistical parameters $\left(\mathbf{A}_m^R, \mathbf{B}_m^R, \vec{\pi}_m^R\right)$. Using tree-structure quantization, centroids can be mapped according to:

$$c_m^{8,i} \Rightarrow c_m^{R,j}, \text{ for } 1 \le R < 8, \text{ where } j = \left\lfloor \frac{i}{2^{8-R}} \right\rfloor. \quad (7)$$

The steady-state and transitional statistics can be approximated according to:

$$\vec{\pi}_m^R(i) = \sum_{k=0}^{2^\tau - 1} \vec{\pi}_m^{R+\tau}\left(2^\tau i - k\right), \quad (8)$$

$$\mathbf{A}_m^R(i,j) = \frac{1}{2^\tau}\left[\sum_{k=0}^{2^\tau-1}\sum_{l=0}^{2^\tau-1} \mathbf{A}_m^{R+\tau}\left(2^\tau i - k, 2^\tau j - l\right)\right], \quad (9)$$

where τ is chosen as τ=8-R.

Regarding the observation statistics of Λ_m^R, denoted as $b_i^R(\mathbf{o})$, the following approximation needs to hold in order for the downsampling to be accurate:

$$b_i^R(\mathbf{o}) \approx b_i(\mathbf{o}). \quad (10)$$

This approximation holds naturally for many applications. For example, for the noisy spectrogram scenario described in [5], the approximation holds due to the continuous nature of the downsampled centroids. However, for the scenario involving the transmission of digital information across noisy channels, special attention must be paid to guarantee Equation 10. In this case, $b_i(\mathbf{o})$ corresponds to the probability that the binary codeword representing c_m^i is corrupted by noise and decoded as $\mathbf{o}$, and $b_i^R(\mathbf{o})$ corresponds to the probability that the binary codeword representing $c_m^{R,i}$ is detected as $\mathbf{o}$. Thus, the Hamming distance between the codeword representing c_m^i and that representing $c_m^{R,i}$ must be small.

The estimate of component $x_m(n)$ using the approximation of the FB algorithm with R bits of resolution becomes:

Table 1. *Orders of Complexity Required as a Function of R: T refers to the length of the unreliable data in samples.*

R	Complexity
1	$O(4 \cdot T)$
2	$O(16 \cdot T)$
3	$O(64 \cdot T)$
4	$O(256 \cdot T)$
5	$O(1,024 \cdot T)$
6	$O(4,096 \cdot T)$
7	$O(16,384 \cdot T)$
8	$O(65,536 \cdot T)$

$$\widehat{x}_m^R(n) = \sum_{i=1}^{2^R} c_m^{R,i} \gamma_m^{R,i}(n), \tag{11}$$

where:

$$\gamma_m^{R,i}(n) = \frac{\alpha_m^{R,i}(n)\,\beta_m^{R,i}(n)}{\sum_{j=1}^{2^R} \alpha_m^{R,j}(n)\,\beta_m^{R,j}(n)}. \tag{12}$$

The values $\alpha_m^{R,i}(n)$ and $\beta_m^{R,i}(n)$ can be determined as:

$$\alpha_m^{R,i}(n) = \left[\sum_{j=1}^{2^R} \mathbf{A}_m^R(j,i)\,\alpha_m^{R,j}(n-1)\right] b_i^R(\mathbf{o}_n), \tag{13}$$

$$\beta_m^{R,i}(n) = \sum_{j=1}^{2^R} \mathbf{A}_m^R(i,j)\,\beta_m^{R,j}(n+1)\,b_j^R(\mathbf{o}_{n+1}). \tag{14}$$

Thus, the computational complexity required for the estimation of a series of unreliable features of length T using the proposed FB approximation method is of order $O\left(2^{2R}T\right)$. Table 1 gives the specific orders of complexity involved with the proposed approximated FB algorithm. For example, if a series of 10 unreliable features is estimated with the standard FB method, it would require $587,775$ additions and $592,641$ multiplications. However, if a series of unreliable features of the same length is estimated with the proposed method with a resolution of R=4 bits, it would only require $2,175$ additions and $2,481$ multiplications.

4. RESULTS FOR REMOTE SPEECH RECOGNITION

Remote speech recognition (RSR) provides distributed clients with the ability to transmit speech to a central server for the purpose of automatic speech recognition (ASR), thus forwarding the computational load involved with the recognition process. This, however, requires transmission of speech over error-prone channels, introducing the need for error concealment (EC). Two types of RSR systems exist: Distributed Speech Recognition (DSR) and Network Speech Recognition (NSR). The former involves transmission of speech features strictly for the purpose of recognition, thus allowing for low-bitrate transmission. The latter involves the transmission of coded speech, requiring larger bitrates than DSR, but allowing for compatibility with existing speech communication systems.

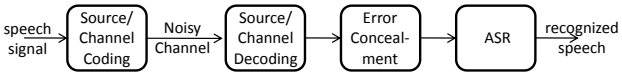

Fig. 1. *Overview of the Remote Speech Recognition System*

Table 2. *Testing Channel Conditions: $\overline{d}$ refers to the average duration in state 1, and "ave. loss' refers to the percentage of frames in the lossy state (state 1).*

Cond.	P_e	p	q	$\overline{d}$	ave. loss
1	0.25	0.1071	0.2500	4.0	30.0 %
2	0.25	0.0833	0.1250	8.0	40.0 %
3	0.25	0.0625	0.0625	16.0	50.0 %

We apply the proposed FB approximation algorithm to error concealment within a RSR system to examine whether or not there is any degradation in performance. The RSR system, shown in Figure 1, includes a client-based feature extraction scheme that processes the first 13 Linear Predictive Cepstral Coefficients (LPCCs), along with the log-spectral energy. The choice of using LPCCs over MFCCs is for compatibility with speech coders, which typically transmit Line Spectral Frequencies (LSFs). Note that the source coding can stand alone, as in Distributed Speech Recognition, or be embedded within a speech coder, as in Network Speech Recognition. For the packet-based channel, the observation probabilities of Λ_m^R can be determined as follows. Each packet is transmitted as a bit pattern $\mathbf{d}_m^R$, and the received bit pattern, $\widehat{\mathbf{d}}_m^R$, is either known to be correctly received, so that:

$$P\left(\widehat{\mathbf{d}}_m^R \mid \mathbf{d}_m^R\right) = \begin{cases} 1, \text{ if } \widehat{\mathbf{d}}_m^R = \mathbf{d}_m^R \\ 0, \text{ if } \widehat{\mathbf{d}}_m^R \neq \mathbf{d}_m^R \end{cases}, \tag{15}$$

or corrupted, so that $\widehat{\mathbf{d}}_m^R(i)$=$\mathbf{d}_m^R(i)$ with probability 1-P_e. Detection of corrupt packets is assumed, but can be otherwise implemented by means of error-detecting codes [2].

For the current application, the approximation given by Equation 10 was guaranteed in the following way: First, the centroids of the original quantizer Q_m^8 were ordered such that: $c_m^{8,i} < c_m^{8,i+1}$. Secondly, prior to transmission, centroid in-

Table 3. *Root-Mean-Square Distortion Measures of Estimated Features. Channel conditions are defined in Table 2.*

	Forward-Backward Approximation with Resolution R							
Condition	R=1	R=2	R=3	R=4	R=5	R=6	R=7	R=8
1	0.310	0.244	0.213	0.204	0.202	0.196	0.191	0.179
2	0.399	0.345	0.315	0.305	0.303	0.298	0.292	0.280
3	0.478	0.436	0.412	0.404	0.405	0.398	0.390	0.378

Table 4. *Recognition Results for Remote Speech Recognition, Using Clean Speech from the Aurora-2 Database*

	Forward-Backward Approximation with Resolution R							
Condition	R=1	R=2	R=3	R=4	R=5	R=6	R=7	R=8
1	77.86 %	92.05 %	94.87 %	95.64 %	95.73 %	95.89 %	96.13 %	96.62 %
2	64.63 %	81.21 %	85.08 %	86.12 %	86.34 %	86.83 %	87.17 %	88.39 %
3	57.11 %	70.62 %	73.38 %	74.27 %	74.24 %	74.70 %	74.61 %	76.24 %

dices were mapped to binary codewords according to traditional binary representation.

The channel was modeled using a 2-state Gilbert-Elliot model, in which state 1 induced a bit error probability of P_e. The transitional probability from state 0 to state 1 is given by p, and the transitional probability from state 1 to state 0 is given by q. The proposed algorithm was applied to estimate unreliable features, prior to recognition.

The system was tested using clean speech from the Aurora-2 database [6]. The recognition engine used 16-state, 3-mixture word models. The channel conditions used for testing are summarized in Table 2. Table 3 provides the root-mean-square (RMS) distortion measures of the estimated features, averaged across feature elements. Word recognition accuracy results are provided for a range of resolutions in Table 4.

As can be concluded from Table 4, the complexity of the FB algorithm can be greatly reduced via the proposed approximation method with little performance degradation, when applied to the problem of error concealment in RSR. The word recognition accuracy results obtained across a range of channel conditions show a small drop in performance for resolution settings as low as R=4 and R=5, as compared to the original algorithm with R=8. These settings correspond to complexity reductions of 256 and 64 times, respectively.

5. CONCLUSIONS

This paper proposes an efficient approximation to the general forward-backward algorithm based on HMM downsampling, for the purpose of estimating lost features. We present the FB approximation method by developing HMMs based on quantizers of lower resolutions, which are obtained through a tree-structure mapping of centroids. Statistical parameters of lower-resolution HMMs are adapted as functions of the original parameters. To illustrate the effectiveness of the proposed method, we apply it to the problem of error concealment in remote speech recognition. The FB approximation is shown to provide comparable recognition accuracy results for packet-based RSR relative to the standard FB algorithm, while reducing the required computational load by a factor of over 250.

6. REFERENCES

[1] L. R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of the IEEE, vol. 77, No. 2, pp. 257-286, 1989.

[2] A. M. Peinado, V. Sanchez, J. L. Perez-Cordoba, and A.J. Rubio, *Efficient MMSE-Based Channel Error Mitigation Techniques. Application to Distributed Speech Recognition Over Wireless Channels*, IEEE Trans. Wireless Channels, Vol. 4, No. 1, pp. 14-19, Jan. 2005.

[3] A. M. Peinado, V. Sanchez, J. L. Perez-Cordoba, and A. Torre, *HMM-Based Channel Error Mitigation and its Application to Distributed Speech Recognition*, Speech Communication, vol. 41, pp. 549-561, Nov. 2003.

[4] A. M. Kunduz, *Digital Speech: Coding for Low Bitrate Communication Systems*, Wiley, 2004.

[5] B. Raj, M. L. Seltzer, and R. M. Stern, *Reconstruction of Missing Features for Robust Speech Recognition*, Speech Communication, vol. 43, pp. 275-296, 2004.

[6] D. Pearce, *Enabling New Speech Driven Services For Mobile Devices: An Overview of the ETSI Standards Activities for Distributed Speech Recognition Front-Ends*, AVIOS 2000: Speech Appl. Conf., Vol. 5, May 2000.