INTEGRATED BIAS REMOVAL TECHNIQUES FOR ROBUST SPEECH RECOGNITION

Craig Lawrence and Mazin Rahim
University of Maryland, College Park, MD 20742
1AT&T Labs-Research, Murray Hill, NJ 07974

ABSTRACT
In this paper, we present a family of maximum likelihood (ML) techniques that aim at reducing an acoustic mismatch between the training and testing conditions of hidden Markov model (HMM)-based automatic speech recognition (ASR) systems. We propose a codebook-based stochastic matching (CBSM) approach for bias removal both at the feature level and at the model level. CBSM associates each bias with an ensemble of HMM mixture components that share similar acoustic characteristics. It is integrated with hierarchical signal bias removal (HSBR) and further extended to accommodate for N-best candidates. Experimental results on connected digits, recorded over a cellular network, shows that the proposed system reduces both the word and string error rates by about 36% and 31%, respectively, over a baseline system not incorporating bias removal.

1. INTRODUCTION
In this paper, we will focus on an acoustic mismatch between the training and the testing conditions of hidden Markov model (HMM)-based telephone speech recognition systems. This mismatch, which may be caused by variations in the telephone channel and transducer equipments, will be modeled as a time-varying additive bias, $b_t$, in the cepstral, or log-spectral, domain. Specifically, we assume that the cepstrum of the received signal at time $t$ is given by

$$y_t = x_t + b_t,$$

where $x_t$ is the cepstrum of the undistorted signal. A family of compensation techniques for estimating the bias $b_t$ is presented.

When an acoustic mismatch is known to exist, compensation may be carried out in one of two ways. One is to attempt to bring the testing data "closer" to the acoustic space of the training environment. Lines of research falling within this framework include spectral subtraction [9, 3], signal bias removal [11], cepstral mean subtraction [1], and stochastic matching [12]. Alternatively, we could transform the models to better characterize the test data. Typical techniques include Bayesian learning procedures [8], parallel model combination [5] and stochastic matching [12]. An excellent survey of techniques for robust speech recognition can be found in [7].

Both feature-based and model-based transformations are studied in this paper. A new method is proposed, referred to as codebook-based stochastic matching (CBSM) which applies the expectation-maximization (EM) algorithm [2] to compute a bias for each ensemble of HMM mixture components that share similar acoustic characteristics. CBSM enables integration of bias removal pre-processing methods and can accommodate for multiple search candidates. Our proposed system will be evaluated on a scenario in which a set of wireline-trained models are tested on data collected in a wireless environment. The time-varying nature of this problem provides a challenging testbed for the techniques considered in this paper.

2. ML STOCHASTIC MATCHING
Consider a sequence of feature vectors $Y = \{y_1, \ldots, y_T\}$, s.t. $y_t \in \mathbb{R}^D$, and a set of trained HMMs $\{\Lambda_x\}$. Let the observation density for state $n$ with $M$ components be defined as:

$$p_X(y|n) = \sum_{m=1}^{M} w_{n,m} N(y; \mu_{n,m}, C_{n,m}),$$

where $w_{n,m}$ is the mixture weight and $N(y; \mu_{n,m}, C_{n,m})$ is a Gaussian distribution with corresponding mean vector $\mu_{n,m}$ and covariance matrix $C_{n,m}$. Given an estimated bias sequence $B = \{b_1, \ldots, b_T\}$, we allow the action of bias removal to be either a feature-space transformation $x_t = y_t - b_t$, $t = 1, \ldots, T$, or, a model-space transformation (model adaptation) $\Lambda_Y = M_B(\Lambda_X)$, where $M_B(\cdot)$ is determined by the particular method being applied. In either case, the problem of robust speech recognition, in the framework we have chosen, is reduced to that of computing the bias estimate $B$.

The application of the EM algorithm for estimating the bias $B$ has been described in the stochastic matching (SM) framework proposed by Sankar and Lee [12]. Given a word (or phone) sequence $W$, the bias estimate

$$B = \arg \max_B p(Y|B, W, \Lambda_X)p(W)$$

can be computed by maximizing the following auxiliary function:

$$Q(\theta|B) = E\{\log p(Y|B, W, \Lambda_X)|Y, W, B, \Lambda_X\}. $$

Since $\tilde{W}$ is fixed at this stage, the term $p(\tilde{W})$ may be omitted.

*This work was performed at AT&T Labs-Research during the summer of 1996.
Upon conditioning the signal (or model), we can solve for the optimal $W$, i.e., $\tilde{W}$, and reiterate this procedure to further refine the bias estimate.

At this point we make no assumptions regarding the domain in which the bias is applied (i.e., whether it is frame-based, state-based, etc.), other than it is additive as proposed in Eqn. (1). In general, let $b(t, n, m)$ be the bias estimate associated with frame $t$, state $n$, and mixture component $m$. It can be shown (following the argument of [12]) that the auxiliary function reduces to

$$Q(\bar{B}|B) = \sum_{i=1}^{T} \sum_{n=1}^{N} \gamma_{i}(n, m) \cdot \frac{1}{2}[y_{t} - b(t, n, m) - \mu_{n,m},] C_{n,m}^{-1}[y_{t} - b(t, n, m) - \mu_{n,m}] , \tag{5}$$

where $\gamma_{i}(n, m) = p(Y, s_{t} = n, c_{t} = m|B, \Lambda_{X})$ and $N$ is the number of states. As $Q(\bar{B}|B)$ is a concave function of $\bar{B}$, the optimal bias may be obtained by differentiating the above expression and solving for the zeros.

3. CODEBOOK-BASED STOCHASTIC MATCHING (CBSM)

In [6], we demonstrated that the recognition performance can be enhanced when computing multiple biases for each utterance. On the other hand, we have experienced no improvement when relaxing the stochastic constraints by increasing the bias resolution, such that a separate bias is assigned to each mixture component. To provide a trade-off between increasing the bias resolution and relaxing the stochastic constraints, we propose to associate each bias to a group of mixture components that share similar acoustic characteristics. The so-called codebook-based SM (CBSM) approach uses the concept of “tying” among model parameters to determine the appropriate number of biases to be used.

Let $\{\mu_{n,m}\}$ be the set of mixture component mean vectors for $\Lambda_{X}$, where $n$ and $m$ represent the state and mixture component indices, respectively. Let $\Omega_{1}, \ldots, \Omega_{K}$ be $K$ classes (or codewords) that span the entire space of the models with associated set of centroids $z_{1}, \ldots, z_{K}$. In the current study, the codebook $\Omega$ is constructed by clustering $\{\mu_{n,m}\}$ using the Lloyd algorithm with a Euclidean distance. For notational simplicity, we will assume that the elements of the sets $\Omega_{k}$ are the tuples which index the corresponding mixture component means (e.g., if $\mu_{n,m}$ is clustered to the codeword $z_{k}$, then $(n, m) \in \Omega_{k}$). Since we are interested in computing a bias for each codeword, then

$$b(t, n, m) = \sum_{k=1}^{K} b_{k} I_{\Omega_{k}}(n, m), \tag{6}$$

where $I_{\Omega_{k}}(\cdot)$ is the indicator function for the set $\Omega_{k}$.

Using this notation, we may rewrite the auxiliary function (5) as

$$Q(\bar{B}|B) = \sum_{i=1}^{T} \sum_{k=1}^{K} \gamma_{i}(n, m) \cdot \left\{ -\frac{1}{2} [y_{t} - \tilde{b}_{k} - \mu_{n,m},] C_{n,m}^{-1}[y_{t} - \tilde{b}_{k} - \mu_{n,m}] \right\} \tag{7}$$

By differentiating this expression with respect to $\tilde{b}_{k}$, and setting to zero, we may solve for the class biases, $\{\tilde{b}_{k}^{(i)}\}_{i=1}^{T}, (i = 1, \ldots, K,$ yielding

$$\tilde{b}_{k}^{(i)} = \frac{\sum_{i=1}^{T} \gamma_{i}(n, m) \frac{y_{t,i} - \mu_{n,m,i}}{\sigma^{2}_{n,m,i}}}{\sum_{i=1}^{T} \gamma_{i}(n, m) \sigma^{2}_{n,m,i}}. \tag{8}$$

The means of mixture components are updated based on their codebook association, such that

$$\mu_{n,m} = \mu_{n,m} - \sum_{k=1}^{K} \tilde{b}_{k} \Omega_{k}(n, m), \quad \forall(n, m). \tag{9}$$

![Figure 1: A block diagram of an ASR system incorporating CBSM.](image)

A schematic characterization of CBSM is shown in Figure 1. Following feature analysis, the first-pass of recognition is performed using the original models $\Lambda_{X}$ along with the features $Y$. The resultant transcription $W$ and state segmentation are then used for model bias removal. CBSM is applied at this point, aided by the codebook $\Omega_{k}$ to generate a new set of HMMs, $\Lambda_{Y}$. A second-pass through the recognizer is then conducted using $\Lambda_{Y}$ and $Y$ to generate the recognized string.

3.1. INTEGRATED HSBR/CBSM

The basic strategy of hierarchical signal bias removal (HSBR) is described in two steps [11]. For a given codebook $\Omega = \{\mu_{1}, \ldots, \mu_{K}\}$ of size $K$, the first step includes estimating a bias for each codeword, i.e., $\{b_{k}\}_{k=1}^{K}$. In the second step, a frame-dependent bias $b_{t}$ is computed as a function of $\{b_{k}\}_{k=1}^{K}$, and subtracted from the input speech signal at each time frame $t$, yielding the conditioned signal $x_{t}$. The likelihood function in HSBR is based on a weighted acoustic distance with respect to the codebook $\Omega$ and neglects all the temporal constraints embedded in the
HMMs. The process may be iterated until some optimality criterion is satisfied. The hierarchical aspect of HSBR was inspired by [4] and operates by performing the signal conditioning as described above for multiple codebooks of increasing size in succession, beginning with \( K = 1 \) and ending with \( K = K_{\text{max}} \). The primary advantage of HSBR is that it is a one-pass scheme that generates a time-variably mismatch between the HMM and the test signal.

CBSM, as illustrated in Figure 1, is a model-based transformation in which bias removal is applied to the model parameters leaving the original signal \( Y \) unchanged. There is clearly no reason why the recognition performance cannot be further enhanced by applying bias removal both at the feature level as well as the model level. In fact, applying signal conditioning, as it is the case in HSBR, can further help in enhancing the signal prior to recognition, thus improving the efficiency of the recognizer and minimizing search errors due to incomplete decoding paths. This is particularly important in situations where frequent beam failures are occurring due to a severe acoustic mismatch between the testing features and the training model. In these circumstances, SM-based approaches have little hope in improving performance since no segmentation information would be available.

In this study, we have integrated CBSTM and HSBR for added robustness so that both feature and model space transformations are performed simultaneously. The procedure is similar to that shown in Figure 1 with the addition of the HSBR module prior to the first pass of recognition. Therefore, rather than applying \( g_i \) at each time frame in CBSTM, we would simply use the conditioned signal \( z_i \).

3.2. N-BEST CBSTM

Ideally, to improve recognition performance through SM, one would like to reduce the mismatch between the correct hypothesis and the model parameters. In practice, however, since SM deals with the recognized hypothesis, this level of performance can probably never be achieved. One possibility for reducing this problem is to provide bias removal with multiple candidates (i.e., alternative transcriptions) with the hope that one of the candidates is truly the correct one. In this study, we have incorporated multiple candidates during bias removal using an N-best search. From our experience with digit recognition, we have found that for those strings that are incorrectly recognized 35-60\% of the time the correct string appears in the top four candidates.

Our approach is to incorporate the statistics of the N-best candidates into the process of estimating the bias. During the first-pass through the recognizer, N-best candidates are generated with corresponding state segmentations \( \{ s_i^l \}_{l=1}^7 \), \( l = 1, \ldots, N \). The computed bias for the \( k^{th} \) codeword is

\[
B_k = \frac{\sum_{n=1}^N \sum_{m=1}^T \sum_{(n,m) \in \mathcal{B}_k} \gamma_i(n,m) \frac{y_i(n,m) - \mu_{n,m,i}}{\sigma_{n,m,i}^2}}{\sum_{n=1}^N \sum_{m=1}^T \sum_{(n,m) \in \mathcal{B}_k} \gamma_i(n,m) \frac{1}{\sigma_{n,m,i}^2}},
\]

\( i = 1, \ldots, D, k = 1, \ldots, K \), where \( \gamma_i(n,m) \) is computed for the \( i^{th} \) candidate. The computed biases are then used to update all model parameters according to Eqn. 9. This approach of using N-best candidates with CBSTM will be referred to as NB-CBSTM.

4. EXPERIMENTAL RESULTS

A speaker-independent telephone-based connected digits database was used in this study. Digit strings ranging from one to sixteen digits in length were extracted from different field-trial collections with varied environmental conditions and transducer equipment. The training set consisted of 16089 digit strings and were used for designing the recognition models. The testing set consisted of 402 nine-digit strings collected in a wireless telephone environment from speakers using mostly analog handsets. The recording conditions varied from call to call, ranging from 5dB to 35dB SNR. During feature extraction, each 30 msec frame was represented by 12 linear predictive coding (LPC) liftered cepstral coefficients, along with a normalized logarithmic energy. This feature vector was augmented with its first and second order time derivatives, the so-called delta cepstrum/delta energy and delta-delta cepstrum/delta-delta energy, resulting in a vector of 39 features per frame. For recognition, each digit was modeled by a set of left-to-right continuous density HMMs which captured all possible inter-digit coarticulation [10]. A total of 274 context-dependent subword models were used, trained by ML estimation. Subword models consisted of 3 to 4 states, each having a mixture of 8 Gaussian distributions. The background noise model (i.e., silence) included a single state with 32 Gaussian distributions. All experiments were performed using the same training model without any type of conditioning or bias removal and with a known length grammar.

Figure 2 illustrates the effect of varying the CBSTM codebook size (i.e., number of biases utilized) on the recognition performance. The experiment which was conducted with 0, 1, 2, 4, 8, 16 and 32 biases seem to indicate that a CBSTM codebook size of 8 and beyond produces the lowest error rate. Therefore, the CBSTM codebook was fixed at 8 in all remaining experiments. Results for integrating HSBR and CBSTM are shown in Table 1 (column 3). These results were obtained when first processing the signal with HSBR and

---

2Each digit is divided into three segments, a head, a body, and a tail. A digit has one body and multiple heads and tails depending on the preceding and following context.
to increase the processing time by a factor of two to four folds. This is a major problem when trying to operate a real-time service. A bias removal framework which can accommodate for sequential processing is clearly needed and remains a challenging problem in speech recognition research.

6. REFERENCES


5. CONCLUSIONS

A codebook-based stochastic matching method was proposed in this paper. CBM associated a separate bias with an ensemble of mixture components that shared similar acoustic characteristics. This method was found to produce improvement in recognition performance when using 8 or more codewords (or biases). Further, we integrated hierarchical signal bias removal with an extension of CBM that accommodated for four best candidates. Without having to retrain the original HMM system, we experienced a reduction in the word and string error rates by 36% and 31%, respectively, when testing on data collected from a cellular environment. The major disadvantage of stochastic matching based techniques is that they require multiple passes through the recognizer. In Table 1, we found HSBR/CSBM and HSBR/NB-CSBM to be better than other methods.

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>HSBR/CSBM</th>
<th>HSBR/NB-CSBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word Error (%)</td>
<td>3.3</td>
<td>2.4</td>
<td>2.1</td>
</tr>
<tr>
<td>String Error (%)</td>
<td>14.1</td>
<td>10.8</td>
<td>9.7</td>
</tr>
<tr>
<td>Rejection (%)</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>CPU (sec)</td>
<td>1831</td>
<td>3419</td>
<td>7063</td>
</tr>
</tbody>
</table>