HIERARCHICAL STOCHASTIC FEATURE MATCHING FOR ROBUST SPEECH RECOGNITION

Hui Jiang, Frank Soong and Chin-Hui Lee

Dialogue Systems Research, Multimedia Communication Research Lab,
Bell Labs, Lucent Technologies, Murray Hill, NJ 07974
Email: {hui,fks,chl}@research.bell-labs.com

ABSTRACT

In this paper we investigate how to improve the robustness of a speech recognizer in a noisy, mismatched environment when only a single or a few test utterances are available for compensating the mismatch. A new hierarchical tree-based transformation is proposed to enhance the conventional stochastic matching algorithm in the cepstral feature space. The tree-based hierarchical transformation is estimated in two criteria: i) maximum likelihood (ML) using the current test utterance; ii) Sequential maximum *a posterior* (MAP) using the current and previous utterances. Recognition results obtained using a hands-free database show the proposed feature compensation is robust. Significant performance improvement has been observed over the conventional stochastic matching.

1. INTRODUCTION

It is well known that any mismatch between training and testing conditions can significantly degrades the performance of a speech recognizer. Many compensation and adaptation algorithms [3], attempting to maintain or improve the ASR performance by minimizing the mismatch, have been proposed. When a massive amount of data is available, adaptation techniques[4, 6] are shown to be very effective in increasing the robustness of ASR under various conditions. However, the problem becomes more challenging when very few, sometimes, only one single test utterance is available for compensation. We study this difficult case in this paper and propose a method to deal with the data sparseness. Stochastic matching [5] has been shown to be effective in transforming the noisy test feature vector into the clean feature space. The approach, as discussed in [5], is not limited to the feature space transformation. Signal space or model space is also possible for this transformation. However, compensation in the feature space does indeed have certain advantages: (1) Transformation in the feature space is simpler and flexible than its counterpart in the model or signal space; (2) It is relatively easy to couple a feature compensation module in the overall recognizer since decoding and recognition algorithm need not to be modified.

In this paper, we propose a hierarchical transformation in the feature domain to compensate various mismatches between training and testing conditions in order to improve the robustness of a recognizer in various operating environments. The name "hierarchical" is from the layered structure of a tree where the transformation is estimated. The hierarchical transformation parameters are estimated from the test data (utterances) using two possible optimization criteria: i) maximum likelihood (ML); and ii) sequential maximum *a posterior* (MAP). Experimental results obtained using a hands-free database recorded in a moving car show that the proposed method yields significantly better performance than the conventional stochastic matching method.

2. HIERARCHICAL TRANSFORMATION FOR FEATURE COMPENSATION

Based on the signal distortion model used in [1], distortions in a noisy speech signal can be broken down into two components, additive and convolutional. Given clean speech signal, x, additive noise, n and convolutional noise, h, the noisy speech signal, y, can be expressed as the form of

$$y = x \otimes h + n \tag{1}$$

where $\otimes$ denotes convolution. In the cepstral domain upon which most modern speech recognitions are based, the above relation can be re-written in cepstral domain as

$$\mathbf{x} = \mathbf{y} - \mathbf{h} - \text{IDFT}\{\ln(1 + e^{\text{DFT}[\mathbf{n}-\mathbf{h}-\mathbf{x}]})\} \tag{2}$$

where bold letters represent the corresponding cepstral features, and DFT and IDFT denote discrete Fourier transform and its inverse. In the cepstral domain, the clean speech can be estimated from noisy speech via the following bias-based transformation:

$$\mathbf{x} = \mathbf{y} - \mathbf{b}(\mathbf{x}, \mathbf{h}, \mathbf{n}) \tag{3}$$

where the bias, $\mathbf{b}(\mathbf{x}, \mathbf{h}, \mathbf{n})$, depends on many factors. Usually the exact values of $\mathbf{x}$, $\mathbf{h}$ and $\mathbf{n}$ are unknown in practice, we assume the bias depends solely on noisy speech, $\mathbf{y}$, and we approximate the compensation function and rewrite as:

$$\hat{\mathbf{x}} \approx \mathbf{y} - \mathbf{b}(\mathbf{y}) \tag{4}$$

Here the bias $\mathbf{b}(\mathbf{y})$ is a non-linear function without an explicit expression. In this paper, we adopt a piece-wise linear approximation for the above function, $\mathbf{b}(\mathbf{y})$. More specifically, we partition the feature space of noisy signal $\mathbf{y}$ into I disjoint regions, i.e., $\Omega_1, \Omega_2, \cdots, \Omega_I$. For every region Ω_i, we estimate the corresponding bias $\mathbf{b}_i$ to compensate noisy speech, $\mathbf{y}$. To use data more efficiently, we organize all biases into a hierarchical structure, e.g. a *tree*. By using a hierarchical structure, the bias function can be expressed in multi-scale, where biases at different level of tree represent different resolution. Every bias at the lowest level or a leaf node corresponds to one region, Ω_i, in the feature space and the node at a higher level represents a union of all its children. In

this paper, we denote the hierarchical bias-based transformation as $\mathcal{T} = \{\mathbf{b}_i^{(n)}\}$ where $\mathbf{b}_i^{(n)}$ denotes i-th bias at the n-th level of tree.

Given a transformation $\mathcal{T}$, every feature vector $\mathbf{y}$ in noisy speech is classified into an appropriate region Ω_i by a classifier $\mathcal{C}$. If the bias in the leaf node corresponding to Ω_i can be reliably estimated, it will be used to compensate $\mathbf{y}$ as in eq.(4). Otherwise, we use the bias in its parent node and the pop-up process can go up all the way to the root node. The classifier $\mathcal{C}$ can be implemented in various ways. If the tree is built directly from data by some clustering methods, such as hierarchical VQ, the classifier $\mathcal{C}$ is simply a VQ encoder. Alternatively, we also can construct the tree from an existing HMM rather than from the data itself. Some clustering methods are used to grow a tree from all Gaussian mixture components (kernels) in HMM. In the final tree each leaf node corresponds to a single Gaussian component in HMM. In this case, the classifier $\mathcal{C}$ is implemented in the recognition process: Each utterance is force-aligned (with transcription) or recognized (without transcription) against HMM models to get the optimal Viterbi path; Then each frame of data is distributed down the tree to the corresponding Gaussian component, i.e., leaf node of tree, based on the optimal path.

The hierarchical transformation $\mathcal{T}$ is estimated from data. Depending on the optimization criteria used in the estimation, we will have various strategies for robust speech recognition. In this paper, we study two different estimation methods, i.e., ML and sequential MAP.

3. HIERARCHICAL STOCHASTIC FEATURE MATCHING: ML ESTIMATION

At first, we extend Stochastic Matching in [5] to use the hierarchical transformation $\mathcal{T}$ proposed in section 2. In [5], one or two biases have been used to compensate telephone channel mismatch. By using a hierarchical transformation $\mathcal{T}$, we believe stochastic matching strategy will become more effective to compensate many general mismatches between test data and models since more structured parameters are used in hierarchical transformation. The method is named as *Hierarchical Stochastic Feature Matching* (HSFM) in this paper.

Stochastic matching in feature domain is usually performed in two passes. In the first pass, the transformation is estimated for current test data based on Maximum Likelihood (ML) criterion. Then the current test data is compensated by the estimated transformation. In the second pass, the compensated data is recognized with original model to get the final results. Given a test utterance $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_T\}$, our HSFM algorithm similarly runs as follows:

(1) First pass (Feature compensation)

- Recognize (decode) the current utterance $\mathbf{Y}$ to get the optimal Viterbi path.
- According to the optimal path, populate all frames of $\mathbf{Y}$ into corresponding nodes in the tree. Then all biases in the whole tree are estimated based on the maximum likelihood (ML) criterion. For instance, a bias $\mathbf{b}_i^{(n)}$ in a tree node $\mathcal{O}_i^{(n)}$ is estimated as:

$$\mathbf{b}_{id}^{(n)} = \frac{\sum_{m \in \mathcal{M}_i^{(n)}} \sum_{t=1}^{T} \gamma_t(m) \frac{\mathbf{y}_{td} - \mu_{md}}{\sigma_{md}^2}}{\sum_{m \in \mathcal{M}_i^{(n)}} \sum_{t=1}^{T} \gamma_t(m) \frac{1}{\sigma_{md}^2}} \qquad (5)$$

where $\mathcal{M}_i^{(n)}$ stands for the Gaussian set including all Gaussian mixands belonging to the node $\mathcal{O}_i^{(n)}$ or all nodes below it in tree, and $\gamma_t(m)$ denotes the probability of $\mathbf{y}_t$ residing in the mth Gaussian component which is calculated based on only the optimal Viterbi path, and μ_m and σ_m^2 denotes the mean and variance vectors of the m-th Gaussian component.

- Cut the whole tree according to some pre-set threshold, N, i.e., data allocated into a tree node has to be more than N frames.
- Compensate the data $\mathbf{Y}$ based on the biases estimated in the cut tree, as in eq.(4).

(2) Second pass (Recognition)

- The compensated data is sent to decoder to get the final recognition results.

4. HIERARCHICAL STOCHASTIC FEATURE MATCHING: INCREMENTAL MAP

In the above ML-based HSFM, the transformation $\mathcal{T}$ is estimated solely from current test utterance. For a short utterance, usually only a few biases at the top of $\mathcal{T}$ can be estimated reliably because of the data sparseness. This may limit performance improvement. A better strategy is that we estimate $\mathcal{T}$ based on sequential Bayesian learning, i.e., we keep a prior pdf, $\rho(\mathbf{b})$, for every node in the tree. Given any test utterance, $\mathbf{Y}$, all biases in the tree are estimated based on Maximum *a posteriori* (MAP) estimation. Meanwhile, $\mathbf{Y}$ is also used to update the prior $\rho(\mathbf{b})$ to derive a posterior pdf, which is treated as a new prior pdf for the next test utterance. In this way, all information seen before can be utilized for compensating the current test utterance.

4.1. Hierarchical Priors

As in [6], we introduce a prior pdf for every node in the tree. For any tree node $\mathcal{O}_i^{(n)}$, based on the concept of *natural conjugate prior*, a prior pdf for its bias $\mathbf{b}_i^{(n)}$ is chosen as

$$\rho_i^{(n)}(\mathbf{b}_i^{(n)}) = \prod_{d=1}^{D} \sqrt{\frac{\tau_{id}^{(n)}}{2\pi}} \exp[-\frac{\tau_{id}^{(n)}}{2}(\mathbf{b}_{id}^{(n)} - \theta_{id}^{(n)})^2] \qquad (6)$$

where $\{\theta_{id}^{(n)}, \tau_{id}^{(n)} \mid d = 1, 2, \cdots, D\}$ are hyper-parameters.

4.2. MAP Estimation of Biases

The MAP algorithm runs in the same way as that in ML-based HSFM except the biases are estimated based on MAP rather than ML in eq.(5). The new bias estimation formula is:

$$\mathbf{b}_i^{(n)} = \arg\max_{\mathbf{b}_i^{(n)}} [\rho_i^{(n)}(\mathbf{b}_i^{(n)})]^{\epsilon} \cdot l(\mathbf{Y} \mid \mathbf{b}_i^{(n)}) \qquad (7)$$

where $l(\mathbf{Y} \mid \mathbf{b}_i^{(n)})$ denotes likelihood function and $0 < \epsilon \leq 1$ is the forgetting factor. Based on the priors in eq.(6), the MAP estimate of $\mathbf{b}_i^{(n)}$ is:

$$\mathbf{b}_{id}^{(n)} = \frac{\epsilon \cdot \tau_{id}^{(n)} \theta_{id}^{(n)} + \sum_{m \in \mathcal{M}_i^{(n)}} \sum_{t=1}^{T} \gamma_t(m) \frac{\mathbf{y}_{td} - \mu_{md}}{\sigma_{md}^2}}{\epsilon \cdot \tau_{id}^{(n)} + \sum_{m \in \mathcal{M}_i^{(n)}} \sum_{t=1}^{T} \gamma_t(m) \frac{1}{\sigma_{md}^2}} \qquad (8)$$

4.3. On-line Update Priors

After all biases are derived in the MAP sense, all the prior pdf's in the tree are then updated for next test utterance according to sequential Bayesian learning.

$$\bar{\rho}_i^{(n)}(\mathbf{b}_i^{(n)}) \propto [\rho_i^{(n)}(\mathbf{b}_i^{(n)})]^{\epsilon} \cdot l(\mathbf{Y} \mid \mathbf{b}_i^{(n)}) \tag{9}$$

All hyper-parameters are updated based on Quasi-Bayes approximation as in [2]:

$$\bar{\theta}_{id}^{(n)} = \frac{\epsilon \cdot \tau_{id}^{(n)} \theta_{id}^{(n)} + \sum_{m \in \mathcal{M}_i^{(n)}} \sum_{t=1}^{T} \gamma_t(m) \frac{\mathbf{y}_{td} - \mu_{md}}{\sigma_{md}^2}}{\epsilon \cdot \tau_{id}^{(n)} + \sum_{m \in \mathcal{M}_i^{(n)}} \sum_{t=1}^{T} \gamma_t(m) \frac{1}{\sigma_{md}^2}} \tag{10}$$

$$\bar{\tau}_{id}^{(n)} = \epsilon \cdot \tau_{id}^{(n)} + \sum_{m \in \mathcal{M}_i^{(n)}} \sum_{t=1}^{T} \gamma_t(m) \frac{1}{\sigma_{md}^2} \tag{11}$$

For simplicity, at the beginning, the prior pdf's start from *non-informative* priors, i.e., for all nodes and $d = 1, 2, \cdots, D$, $\tau_{id}^{(n)} = 0$ and $\theta_{id}^{(n)} = 0$.

5. HANDS-FREE ASR EXPERIMENTS

5.1. Database and Experimental Setup

To evaluate the new algorithms, we use a hands-free car database, called *CARVUI*, recorded in a running car with typical car noise in the background. Speech data were simultaneously recorded through multi-microphone channels, including a head-mounted close-talking microphone and a 16-channel microphone array located on the visor. In our experiments, data from only two channels are used: the close-talking microphone, denoted as *CT*, as reference; and a microphone array channel, denoted as *HF*, as hands-free data. *CARVUI* was recorded from 56 speakers, some of whom are non-native English speakers. Each speaker recorded 3 sessions, including: i) phonetically-balanced TIMIT sentences, ii) digit strings with 1-7 digits, and iii) about 85 short commands for car application, such as "*Window Down*", "*Check Email*", etc. Data was originally recorded in 16-bit PCM format and sampled at 24 kHz sampling rate. All data is down-sampled to 8 kHz for the recognition experiments. In this paper, recognition of commands and digit-strings (unknown length) is performed. We choose command and digit-string data from 6 speakers as our testing data (993 utterances in total). Note some test utterances can be as short as 0.6 second. All data from the other 50 speakers (including TIMIT sentences) were used for adaptation (about 12,000 utterances altogether).

In our experiments, we used 38-dimension feature vector, consisting of 12 Mel LPCCEP, 12 delta CEP, 12 delta-delta CEP, delta log-energy and delta-delta energy. As for acoustic model, we use a state-tying, tri-phone HMM models, estimated mainly from telephone data, as our initial models. Then models are adapted based on the SMAP method[6]. The baseline performance is shown in Table 1. For example, the initial models before it was adapted yields a 15.3% string error rate when tested on the CT data but improved significantly to 5.8% after being adapted to the CT adaptation data. However, the adapted model (*CT-adapt*) does not perform well when tested against the HF test data. A string error rate of 28.6% was obtained, which is still better than the unadapted initial model performance of 46.2% string error rate, but is about two times worse than the 14.3% string error rate obtained by the HF adapted model when the matched adaptation data (HF) was used to adapt the initial model. We should note the all the model adaptation here was done in a supervised mode.

model	Initial	CT-adapt	HF-adapt
CT test data	15.3	**5.8**	13.8
HF test data	46.2	28.6	**14.3**

Table 1: The string error rate (in %) in the baseline system with different model and test data (993 utterances).

Starting from these models, in the following experiments, we are trying to use the proposed HFSM methods to compensate some unknown mismatches when supervision information is unavailable.

5.2. ML-HSFM Experimental Results

For more interesting results, in the rest of the paper, only the HF data was used to test the proposed HFSM algorithm in various conditions. In our experiments, except explicitly stated, HFSM is used to compensate static feature, then delta and delta-delta features are derived from compensated static features. Three different models: Initial, CT-adapt and HF adapted, were used for evaluations. The first set of experiments were performed using the ML criteria in HFSM compensation. For each test utterance, a hierarchical transformation was obtained with a threshold $N = 10$ to cut the hierarchical tree. In other words, the biases are estimated at a node of the tree when at least 10 frames test data frames have been assigned to that node. The results are shown in Table 2 under *ML-HFSM*. Next to the ML-HFSM are the conventional ML-based stochastic matching (ML-SM) results, where only a single bias was used. For comparison, baseline performance (without any compensation) is also repeated in the table. For either the matched (i.e., test data and model adaptation data the same) or mismatched conditions, the ML-HFSM yields better results than the baseline performance and the performance of the conventional SM.

5.3. Sequential MAP-HSFM Experimental Results

Since the test utterance can be very short, a tree-cut threshold $N = 10$ only allows a few biases estimated at the top of the tree. To overcome this insufficient data problem, we use the sequential MAP-HSFM proposed in the section 4. All initial priors in the tree are assigned non-informative. Given a test utterance, HFSM transformation is then estimated based upon the MAP criterion and the priors are updated based on sequential Quasi-Bayes approximation in an un-supervised mode. In this case, all test utterances seen before can be utilized to improve the feature compensation. We use a forgetting factor of $\epsilon = 1.0$ in our sequential Bayesian estimates. In Table 3, the results of sequential MAP-based HFSM are compared with those of the conventional MAP-based stochastic matching (MAP-SM) using a single bias. For the mismatched test condition, the sequential MAP-based HFSM yields better performance than the corresponding entries in Table 2. For example, the intial model performance improves from 43.7% string error rate for the ML-HFSM to 39.0% for sequential MAP-HFSM. When HF data was tested with the matched models *HF-adapt*, we found only a relatively small improvement at $N = 10$ in this case.

model	baseline	ML-SM	ML-HSFM
Initial	46.2	47.3	43.7
CT-adapt	28.6	28.6	26.7
HF-adapt	14.3	14.4	13.8

Table 2: The string error rate (in %) comparison of ML-based HSFM (ML-HSFM) with the conventional ML-based stochastic matching (ML-SM) when HF data is tested with various models. In HSFM, the tree is cut with threshold $N = 10$.

model	baseline	MAP-SM	MAP-HSFM (N)
Initial	46.2	44.9	39.0 (300)
CT-adapt	28.6	28.6	25.6 (300)
HF-adapt	14.3	14.4	14.1 (10)

Table 3: The string error rate (in %) comparison of online MAP-based HSFM (MAP-HSFM) with the conventional MAP-based stochastic matching (MAP-SM) when HF data is tested with various models. Numbers in parentheses indicate thresholds to cut the tree.

5.4. Different feature compensation scheme Δ_0, Δ_1 and Δ_2

Actually we have three different feature compensation schemes for delta and delta-delta features: 1)Δ_0: compensate the static, delta and delta-delta features together; 2)Δ_1: compensate the static feature only, and leave the delta and delta-delta features unchanged; 3)Δ_2: compensate the static feature only, and derive the delta and delta-delta features from compensated static parameters. Three feature compensation schemes yields not too great a performance difference. The Δ_2 scheme is probably the most sensible approach. The recognition performance of Δ_2 seems to confirm that and the result is shown in Fig. 1 for different tree-cutting threshold values.

5.5. The effects of tree-cutting threshold

The choice of threshold N used to cut the tree is dependent upon the HFSM estimation criterion, ML or sequential MAP. The ML-HFSM due to its memoryless bias estimation, i.e., the transformation is estimated based entirely upon the current test utterance, not too large an N can be chosen. On the other hand, for sequential MAP-based HFSM, due to its exploitation of the prior information derived from the past utterances we can relax its choice of N to a rather large value and further performance improvement can be obtained. The ML-HFSM performance is shown in Fig. 2, where Δ_2 was used. When comparing Fig. 1 with Fig. 2, we found that MAP-HSFM consistently outperforms the baseline system. However, ML-HSFM becomes worse when N is larger then 100 because too few of biases are left due to insufficient data.

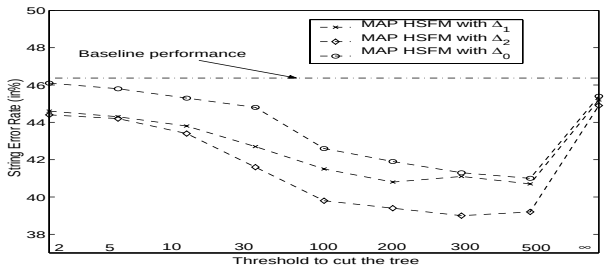

Figure 1: Performance comparison of sequential MAP-HSFM as a function of threshold to cut the tree. (HF data is tested with *No-Adapt* models)

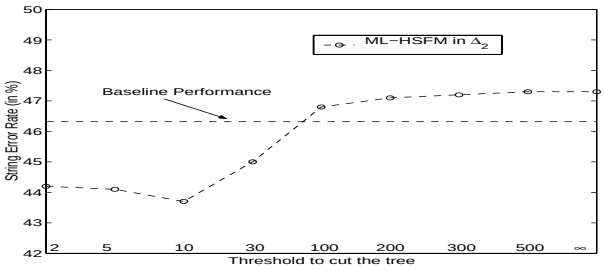

Figure 2: Performance comparison of ML-based HSFM as a function of threshold to cut the tree. (HF data is tested with *No-Adapt* models)

6. CONCLUSIONS

In this work, we have proposed a new hierarchical transformation for stochastic matching in speech feature space in order to compensate mismatches between recognition models and testing data. An advantage to use hierarchical transformation is that we can compensate some complex (even non-linear) mismatches and distortions in a simple and tractable way. In this paper, we have investigated two different optimization criteria to estimate hierarchical transformation from data, namely ML and on-line MAP estimations. Both of them gives significant performance improvements over the conventional stochastic matching method on a hands-free car ASR database. Currently, we are estimating such a transformation using simultaneously recorded stereo data to achieve a more precise mapping between two channels. Moreover, structural constraints can also be imposed as [6] to improve the sequential MAP method.

7. REFERENCES

[1] A. Acero, *Acoustical and Environmental Robustness in Automatic Speech Recognition*, Kluwer Academic Publishers, 1993.

[2] Q. Huo and C.-H. Lee, "On-line Adaptive Learning of the Continuous Density Hidden Markov Model based on Approximate Recursive Bayes Estimate," *IEEE Trans. on Speech and Audio Processing*, Vol.5, No.2, pp.161-172, 1997.

[3] C.-H. Lee, "On stochastic feature and model compensation approaches to robust speech recognition," *Speech Communications*, pp.29-47, Vol. 25, 1998.

[4] C. J. Leggetter and P. C. Woodland,"Maximum likelihood linear regression for speaker adaptation of the parameters of continuous density hidden Markov models," *Computer Speech and Language*, Vol. 9, pp.171-185, 1995.

[5] A. Sankar and C.-H. Lee, "A Maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. on Speech and Audio Processing*, pp.190-202, Vol. 4, No.3, May 1996.

[6] K. Shinoda and C.-H. Lee, "Structural MAP speaker adaptation using hierarchical priors," *Proc. of IEEE Workshop on Speech Recognition and Understanding*, pp.381–388, Santa Barbara, Dec. 1997.