ENROLLMENT IN LOW-RESOURCE SPEECH RECOGNITION SYSTEMS

Sabine Deligne and Satya Dharanipragada

IBM T. J. Watson Research Center
Yorktown Heights, NY (USA)
deligne@us.ibm.com,satya@us.ibm.com

ABSTRACT

In this paper we consider the problem of enrollment for low-resource speech recoginition systems designed for noisy environments. Noise robustness concerns, memory and computational constraints along with the use of compact acoustic models for fast Gaussian computation make adaptation especially challenging. We derive a Maximum A Posteriori (MAP) algorithm especially designed for the fast off-line adaptation of these compact acoustic models. It requires less computation and memory than standard Feature-space Maximum Likelihood Linear Regression (FMLLR) which is another technique well suited for compact acoustic models. In our experiments of speaker enrollment for speech recognition in the car, we present a computationally efficient procedure to simulate noisy conditions with the adaptation data. In these experiments, MAP compares favorably with FMLLR in terms of recognition accuracy. Besides, combining FMLLR and MAP significantly outperforms each technique individually, thus providing an efficient alternative for systems with larger resources.

1. INTRODUCTION

Adaptation to speaker and/or environment is a powerful way to improve speech recognition accuracy in real-world applications. In this paper, we consider off-line supervised adaptation assuming that enrollment data for a specific speaker or environment is available. We address issues that are inherent to the adaptation of noise-robust speech recognition systems with low resources such as the ones used in embedded devices. These issues mainly include maintaining low memory and computational cost at enrollment time, not increasing the memory and computational cost at decoding time, as well as ensuring the same level of noise robustness after adaptation. An additional and important issue arises in the context of systems using compact acoustic models specifically designed to speed up the Gaussian computation while maintaining low computational and memory requirements [4]. One commonly used approach which reduces the computational cost and results in a compact model consists in tying the Gaussian Mixture (GM) distributions across all the states in the acoustic models [3]. In the Subspace Distribution Clustering Hidden Markov Models (SDCHMM) [5], the acoustic space is split into low-dimensional *subspaces* before tying is applied. Standard model adaptation techniques [1], where the model parameters (means and possibly variances) are transformed to better match the speech feature vectors are not well suited for tied systems of this type as they would require re-slicing and re-clustering the distributions of the adapted model. Feature space adaptation techniques, such as Feature space Maximum Likelihood Linear Regression (FMLLR) [2], that transform the input speech features to better match the acoustic models, avoid that problem. Transformation of the input speech features, however, adds a computational overhead at recognition time and the estimation of the feature transform can be too expensive, from a computational and memory standpoint, for very low-resource devices. In this paper, we propose a Maximum A Posteriori (MAP) model adaptation technique that operates directly on the sliced and tied distributions. It does not add any computational overhead at decoding time, and, as will be shown, it requires less computation and memory than FMLLR while providing similar recognition gains. We also show that a combination of FMLLR and MAP yields superior performance and can be used in situations where resources are not at a premium.

The rest of the paper is organized as follows. In section 2, we describe the SDCHMM scheme and we show that it can be viewed as a particular case of a Compound Gaussian Mixture (CGM) model. In section 3, we derive a MAP adaptation procedure for the CGM model. In section 4, we describe the standard FMLLR technique and in section 5 we compare its memory and computational cost with those of our MAP algorithm. In section 6, we explain how we conducted our experiments of speaker enrollment for speech recognition in the car in order to maintain the noise robustness of the system through the adaptation process. The performances obtained with various combinations of MAP and FMLLR are presented in section 7. Conclusions are given in section 8.

2. COMPACT ACOUSTIC MODELS: SDCHMM, CGM

SDCHMM [5] takes advantage of the fact that acoustic models usually consist of Gaussian Mixtures (GM) with diagonal covariances so that the likelihood can be expressed as a product of likelihoods of lower dimension Gaussians. Assuming a GM model defined by the set of priors, means and diagonal covariances $\theta_{gm} = \{\rho_i, \mu_i, \Sigma_i\}_{i=1}^{I}$ the likelihood of an acoustic observation at time t is computed as:

$$p_{gm}(Y(t)) = \sum_{i=1}^{I} \rho_i \mathcal{N}(Y(t); \mu_i, \Sigma_i)$$

where $\mathcal{N}(Y(t); \mu_i, \Sigma_i)$ refers to the Normal distribution of mean μ_i and covariance Σ_i. In an SDCHMM, the Gaussian components are projected onto B orthogonal low dimensional *subspaces* or *streams* : $\{\mu_{i,b}, \Sigma_{i,b}\}_{i=1,b=1}^{i=I,b=B}$. Assuming diagonal covariances, the likelihood can be rewritten as:

$$p_{gm}(Y(t)) = \sum_{i=1}^{I} \rho_i \prod_{b=1}^{B} \mathcal{N}(Y_b(t); \mu_{i,b}, \Sigma_{i,b})$$

where $Y_b(t)$ is the projection of $Y(t)$ on the b^{th} subspace. The Gaussians within each stream b are tied to a small number of distributions called *subspace prototypes*. The subspace prototypes, $\{\overline{\mu}_{k,b}, \overline{\Sigma}_{k,b}\}_{k=1}^{k=K_B}$, can be obtained by clustering the original subspace distributions. The original likelihood is approximated with:

$$p_{sdchmm}(Y(t)) = \sum_{i=1}^{I} \rho_i \prod_{b=1}^{B} \mathcal{N}(Y_b(t); \overline{\mu}_{f(i,b),b}, \overline{\Sigma}_{f(i,b),b})$$

where the function $f : (i,b) \rightarrow k$ maps each original subspace Gaussian $(\mu_{i,b}, \Sigma_{i,b})$ to the closest[1] subspace prototype $(\overline{\mu}_{k,b}, \overline{\Sigma}_{k,b})$. All original distributions are thus closely approximated by some combinations of a small number of subspace prototypes, resulting in substantial memory and computational savings. In the following, we consider a more general model where each original subband distribution is approximated with a linear combination of all the subspace prototypes instead of being approximated with a single subspace prototype. It results in a Compound Gaussian Mixture (CGM) model [7] with a tied structure such that all compound Gaussians share a common pool of subspace prototypes:

$$p_{cgm}(Y(t)) = \sum_{i=1}^{I} \rho_i \prod_{b=1}^{B} \sum_{k=1}^{k=K_b} \pi_{i,b,k} \mathcal{N}(Y_b(t); \overline{\mu}_{k,b}, \overline{\Sigma}_{k,b})$$

where $\pi_{i,b,k}$ denotes the posterior probability of prototype k given the subspace Gaussian (i,b). SDCHMM can thus be seen as a special case of CGM where $\pi_{i,b,k} = 1$ if $f(i,b) = k$ and 0 otherwise. In the following, we propose a MAP algorithm to adapt directly the distributions of the subspace prototypes instead of the original distributions.

3. MAP ADAPTATION OF CGM

The set of MAP estimates are computed as a linear combination of an existing set of parameters and of a set of Maximum Likelihood (ML) estimates [6]:

$$\{\overline{\mu}_{k,b}, \overline{\Sigma}_{k,b}\}^{MAP} = r_{k,b}\{\overline{\mu}_{k,b}, \overline{\Sigma}_{k,b}\} + (1 - r_{k,b})(\overline{\mu}_{k,b}, \overline{\Sigma}_{k,b})^{ML} \quad (1)$$

The problem of computing ML estimates of a CGM model can be formulated as an ML estimation problem from incomplete data and as such it can be solved with an Expectation-Maximization (EM) procedure [8]. Using the same index notation as in section 2, the missing data are the Gaussian component i and the subspace prototype k, b drawn for each stream $Y_b(t)$. Following the EM framework yields the following equations for the ML estimates at iteration (n):

$$\pi_{i,b,k}^{(n+1)} = \frac{\sum_{t=1}^{T} \omega_{i,b,k}^{(n)}(t)}{\sum_{t=1}^{T} u_i^{(n)}(t)} \quad (2)$$

$$\overline{\mu}_{k,b}^{(n+1)} = \frac{\sum_{t=1}^{T} \sum_{i=1}^{I} \omega_{i,b,k}^{(n)}(t) Y_b(t)}{\sum_{t=1}^{T} \sum_{i=1}^{I} \omega_{i,b,k}^{(n)}(t)} \quad (3)$$

$$\overline{\Sigma}_{k,b}^{(n+1)} = \frac{\sum_{t=1}^{T} \sum_{i=1}^{I} \omega_{i,b,k}^{(n)}(t)(Y_b(t) - \overline{\mu}_{k,b}^{(n)})(Y_b(t) - \overline{\mu}_{k,b}^{(n)})^*}{\sum_{t=1}^{T} \sum_{i=1}^{I} \omega_{i,b,k}^{(n)}(t)} \quad (4)$$

[1] closest according to some chosen distance between distributions.

where the posterior probability $u_i^{(n)}(t)$ represents the fractional allocation of frame $Y(t)$ to the Gaussian i at iteration (n):

$$u_i^{(n)}(t) = \frac{\rho_i \prod_{b=1}^{B} \sum_{k=1}^{K_{i,b}} \pi_{i,b,k}^{(n)} \mathcal{N}(Y_b(t); \overline{\mu}_{k,b}^{(n)}, \overline{\Sigma}_{k,b}^{(n)})}{\sum_{i'=1}^{I} \rho_{i'} \prod_{b=1}^{B} \sum_{k=1}^{K_{i',b}} \pi_{i',b,k}^{(n)} \mathcal{N}(Y_b(t); \overline{\mu}_{k,b}^{(n)}, \overline{\Sigma}_{k,b}^{(n)})} \quad (5)$$

And where the posterior probability $\omega_{i,b,k}^{(n)}(t)$ represents the fractional allocation of frame $Y(t)$ to the Gaussian i and to the subspace prototype (k,b) at iteration (n):

$$\omega_{i,b,k}^{(n)}(t) = u_i^{(n)}(t) z_{i,b,k}^{(n)}(t) \quad (6)$$

with

$$z_{i,b,k}^{(n)}(t) = \frac{\pi_{i,b,k}^{(n)} \mathcal{N}(Y_b(t); \overline{\mu}_{k,b}^{(n)}, \overline{\Sigma}_{k,b}^{(n)})}{\sum_{k'=1}^{K_{i,b}} \pi_{i,b,k'}^{(n)} \mathcal{N}(Y_b(t); \overline{\mu}_{k',b}^{(n)}, \overline{\Sigma}_{k',b}^{(n)})}$$

In the particular case of SDCHMM, the EM equations simplify with $\omega_{i,b,k}^{(n)}(t) = u_i^{(n)}(t)$ if $f(i,b) = k$ and 0 otherwise. EM iterations are performed until the likelihood of the adaptation data stops significantly increasing.

The interpolation weight $r_{k,b}$ in equation 1 is computed so as to reflect the confidence put in each ML estimate. The reliability of the estimates $(\overline{\mu}_{k,b}, \overline{\Sigma}_{k,b})$ can be quantified by its occupancy probability over the adaptation data at the last ML iteration:

$$\overline{\omega}_{b,k} = \frac{\sum_{t=1}^{T} \sum_{i=1}^{I} \omega_{i,b,k}(t)}{M} \quad (7)$$

Therefore, we propose to compute $r_{k,b}$ as:

$$r_{k,b} = \begin{cases} \frac{\overline{\omega}_{b,k}^4}{(\overline{\omega}_{b,k})^4 + (1 - \overline{\omega}_{b,k})^4} & \text{if } \overline{\omega}_{b,k} \leq 1 \\ 1 & \text{otherwise} \end{cases} \quad (8)$$

where the threshold M is selected empirically.

4. FEATURE SPACE MLLR

In SDCHMM, standard model adaptation techniques where the model parameters (means and possibly variances) are transformed to better match the speech feature vectors are not well suited as they would require re-slicing and re-clustering the distributions of the adapted model. Feature space adaptation techniques, such as Feature space Maximum Likelihood Linear Regression (FMLLR) [2], that transform the input speech features to better match the acoustic models, avoid that problem. In FMLLR, adaptation is implemented through a feature space transform of the form

$$\hat{Y} = \mathbf{A}Y + \mathbf{b}$$

where Y are the speech frames, $\mathbf{A}$ is the transformation and $\mathbf{b}$, the bias. The transform and bias are computed iteratively such that the likelihood of the transformed adaptation data is maximized. With $u_i^{(n)}(t)$ denoting (as in section 3) the fractional allocation of frame $Y(t)$ to the Gaussian component i in the alignment of the adaptation data at iteration (n), the standard implementation first outlined in [2] requires two steps

1. Computation of the sufficient statistics[2] at iteration (n), the matrix $\mathbf{G}_d^{(n)}$ and the vector $\mathbf{k}_d^{(n)}$:

$$\begin{aligned}\mathbf{G}_d^{(n)} &= \sum_i \frac{1}{\overline{\Sigma}_{f(i,b),d}} \sum_t u_i^{(n)}(t)\mathbf{V}(t)\mathbf{V}(t)^T \\ \mathbf{k}_d^{(n)} &= \sum_i \frac{\overline{\mu}_{f(i,b),d}}{\overline{\Sigma}_{f(i,b),d}} \sum_t u_i^{(n)}(t)\mathbf{V}(t) \end{aligned} \quad (9)$$

where the index $d = 1 \ldots D$ (with D the dimension of the feature vector) refers to a particular dimension within the sub-band b and where, using the same convention as in section 2, $f(i,b)$ refers to the subspace prototype mapped to the original distribution i in sub-band b, and where $\mathbf{V}(t)$ is the extended feature vector

$$V_1(t) = 1 \quad V_d(t) = Y_{d-1}(t) \;\; d = 2 \ldots D+1$$

2. estimation of $\mathbf{A}^{(n)}$ and $\mathbf{b}^{(n)}$ using a row-by-row iterative update scheme derived from the EM framework:

$$\mathbf{w}_d^{(n)} = \left(\frac{\alpha^{(n)}}{\mathbf{c_d^{(n)}}} + \mathbf{k}_d^{(n)}\right) \mathbf{G}_d^{(n)-1} \quad (10)$$

where $\mathbf{w}_d^{(n)}$ is the dth row of $\mathbf{W}^{(n)} = [\mathbf{b}^{(n)} \; \mathbf{A}^{(n)}]$, where $\mathbf{c}_d^{(n)}$ is the extended cofactor row vector $[0 \;\; c_{d,1}^{(n)} \ldots c_{d,D}^{(n)}]$ with $c_{d,j}^{(n)}$ the cofactor of the matrix element $\mathbf{A}_{d,j}^{(n)}$ and where $\alpha^{(n)}$ is a scalar solution to a quadratic equation. Details of this scheme are given in [2].

5. COMPUTATIONAL/MEMORY REQUIREMENTS OF MAP AND FMLLR

For both MAP and MLLR the computations can be broken down into four different categories:

- *Alignment computation:* For both MAP and FMLLR the step of computing an alignment is common. The dominant cost of an alignment computation is Gaussian computation. If n_s is the average number of HMM states active at any given time in the Viterbi-search and n_g is the average number of Gaussians modeling an HMM state, the computational cost during the alignment phase is approximately equal to $n_s \times n_g \times 2D$ per frame of speech. In a typical low-resource system, such as the one we considered in our experiments, n_s and n_g are small and hence the alignment cost is approximately $O(D)$.
- *Statistics computation:* In the standard implementation of FMLLR, the computation of the D statistics matrices $\mathbf{G}_d$ in Step 2 requires $O(D^3)$ operations per frame of speech and each matrix requires storage for D^2 values. In comparison, the MAP procedure requires only $O(D)$ operations per frame to compute the sufficient statistics.
- *Transform computation:* The estimation of the transform and bias vector in FMLLR involves solving linear systems of equations at each iteration which requires $O(D^4)$ operations per iteration. In comparison, the MAP procedure requires only $O(D)$ operations to update the means and variances.
- *Runtime computations:* no additional cost is incurred for the MAP procedure at run-time, whereas FMLLR requires a matrix-vector multiplication at each frame.

The MAP adaptation of the subspace prototypes is thus computationally and memory-wise less demanding than the FMLLR algorithm. It is thus well-suited for the very low-resource systems under consideration. On the other hand, if resources are not very critical, a combination of FMLLR and MAP can be effectively employed.

6. SPEAKER ADAPTATION AND NOISE ROBUSTNESS

We compare the performances of the MAP algorithm and of the standard FMLLR technique for speaker adaptation in a recognition system designed for car environments, i.e. noisy environments. For safety reasons, the adaptation data have to be collected in a car standing still, i.e. in a quiet environment, however using exclusively non-noisy adaptation data may degrade the performance of the system in the presence of noise. Therefore noisy conditions are simulated by artificially adding noise to the clean adaptation data. The segments of added noise are randomly selected from noise files that are pre-recorded in a car at 30mph and at 60mph. The noise feature vectors are pre-computed and combined with the speech feature vectors of the clean enrollment data when these become available. Adding the noise directly to the speech features rather than to the speech waveforms allows us to save memory since the noise features occupy less storage space than the digitized noise waveforms. It also allows us to save computational resources at adaptation time since feature vectors do not need to be extracted from the waveform of the adaptation data with added noise. A possible alternative to this scheme, that would allow further memory savings, would be to build, off-line, models on the noise data collected in the car. At enrollment time, these models would be used to generate the sequences of noise feature vectors to be combined with the enrollment data. In our system, the feature vectors are 13-dimensional mel-frequency cepstral (MFCC) vectors Y computed from the Mel-filtered spectrum Y^f as $Y(t) = C \log Y^f(t)$ where C refers to the Discrete Cosine Transform. The MFCC feature vectors X of the data with added noise are thus computed as $X(t) = Y(t) + C\log(1 + \exp(C^{-1}(N(t) - Y(t))))$ where Y and N refer respectively to the cepstral vectors of the clean adaptation data and noise data, and where C^{-1} refers to the inverse of the Discrete Cosine Transform[3].

7. SPEAKER ADAPTATION EXPERIMENTS

7.1. Experimental protocol

The test data on which the WER is measured consists for each speaker of: 75 sentences uttered in a quiet environment, 100 sentences uttered at 30mph and 100 sentences at 60mph. Each speaker is enrolled in a supervised mode with a total of 75 sentences: 25 sentences collected in a quiet environment and two noisy versions of these sentences with 30mph and 60mph car noise added. The sentences used for adaptation and for test are different but they all relate to a composite set of tasks: addresses, digit strings, dialing

[2] covariance matrix is assumed to be diagonal. All the equations can however be extended to handle the full covariance case.

[3] Since we use 13 dimensional MFCC vectors and 24 Mel filters, truncated Discrete Cosine Transforms are used with, based on our experiments, no loss in performance.

commands, control commands, navigation commands, point of interests, Vindigo style commands[4]. We conducted 3 experiments using different scenarios to design each speaker's adaptation set: (i) an **intra**-task scenario where both the adaptation and test sentences relate to a common single task, (ii) a **mixed**-task scenario where both the adaptation and test sentences relate to a common mix of up to 4 different tasks, (iii) an **inter**-task scenario where the adaptation sentences relate to tasks that are not present in test. The evaluation for the intra-task experiment was carried out on a test set comprising 193 distinct speakers. The evaluation for both the inter and mixed-task experiments were carried out on a common test set comprising 28 speakers. The front end of the speech recognition system used in our experiments computes standard 39-dimensional mel-frequency cepstral coefficients (including deltas and deltas-deltas) from 16-bit PCM sampled at 11.025 KHz. The acoustic models comprises 680 allophones covering all English sounds and modeled with just over 10,000 Gaussians. The 39 dimensional Gaussians are sliced into 19 streams of dimension 2 and one stream of dimension 1. Each of these 20 subspaces is quantized into 64 subspace prototypes.

7.2. Results

Speech recognition Word Error Rates (WER) averaged over all speakers are shown in Table 1 for the intra-task, mixed-task and inter-task experiments (as explained in section 7.1). The baseline WER is obtained without enrolling the speakers. We compare the baseline WER with the WER obtained after adapting the system with either FMLLR, MAP adaptation of the means, MAP adaptation of both the means and variances, and FMLLR followed by either MAP adaptation of the means or MAP adaptation of both the means and variances. For the MAP algorithm the count-threshold in (7) was chosen to be 100 for all experiments. FMLLR reduces the baseline WER by 20% relative in the intra and mixed-task experiments and by 10% relative in the inter-task experiment. The MAP adaptation of the means does not perform as well as FMLLR with relative WER reductions of respectively 16%, 13% for the intra and mixed-task cases, and no significant improvement for inter. The MAP adaptation of both means and covariances on the other hand compares favorably with FMLLR in the intra-task and mixed-task experiments with respectively 24% and 20% relative reductions of the baseline WER. In the inter-task experiment, its performance is slightly inferior than FMLLR with a 7% relative WER reduction. In both the FMLLR and MAP schemes, the performance of the adaptation degrades as the tasks represented in the adaptation set differ more and more from the tasks present in test. In our experiments FMLLR is less task-sensitive however than MAP adaptation. While not being exactly additive, the gains provided by FMLLR and MAP sum up to a certain extent when combining the two techniques: for example FMLLR followed by MAP adaptation of the means reduces the baseline WER by 30%, 27% and 12% in each adaptation scenario. FMLLR followed by MAP adaptation of both means and variances does not give better results than FMLLR followed by MAP adaptation of the means only.

[4] i.e. up to 3-word sentences like "Greenwich Village", "Restaurant", "Show walking directions", "Italian"...etc

	intra	mixed	inter
no adaptation	2.82	3.54	
FMLLR	2.27	2.83	3.19
MAP mean	2.35	3.07	3.52
MAP mean+cov	2.15	2.84	3.28
FMLLR+MAP mean	1.97	2.61	3.12
FMLLR+MAP mean+cov	1.99	2.60	3.22

Table 1. Average WER over all speakers on the intra, mixed and inter-tasks adaptation scenarios

8. CONCLUSION

Low-resource speech recognition systems rely on compact acoustic models to reduce memory and computational costs, hence the need for specific and low-cost model adaptation schemes. We have presented a MAP adaptation algorithm that is well suited in this particular context. MAP is shown to be computationally more attractive than the alternative solution, FMLLR. We reported on speaker adaptation experiments for speech recognition in the car, following a protocol aimed at preserving noise robustness of the adapted system with limited computational/memory costs. In our experiments, MAP and FMLLR yielded similar improvements in terms of recognition accuracy. Furthermore, a combination of FMLLR and MAP yielded a significant improvement over each technique individually.

9. REFERENCES

[1] C. J. Legetter and P. C. Woodland, "Maximum Likelihood linear regression for speaker adaptation of continuous density HMM's," in Comp. Speech Lang., vol9, pp. 171-186, 1996.

[2] M.J.F. Gales, "Maximum linear likelihood transformations for HMM-based speech recognition", Computer Speech and Language, 12:75-98, 1998.

[3] S.J. Young and P.C. Woodland, "The use of state-tying in continuous speech recognition", Proceedings of EUROSPEECH93, 1993.

[4] S. Astrov, "Memory space reduction for Hidden Markov Models in low-resource speech recognition systems", Proceedings of ICSLP02, Denver, USA, 2002.

[5] E. Bocchieri and B.K-W. Mak, "Subspace Distribution Clustering Hidden Markov Models" IEEE Transactions on Speech and Audio Processing, 9:03, 2001.

[6] J. -L Gauvain and C.H. Lee, "Maximum a posteriori estimation of multivariate gaussian mixtures of Markov chains," IEEE transactions on Speech and Audio Processing, 2(2):291-298, April 1994.

[7] S. Chen and R. A. Gopinath, "Gaussianization", Proceedings of NIPS 2000, Denver, USA, 2000.

[8] A. P. Dempster and N. M. Laird and D. B. Rubin, "Maximum-Likelihood from Incomplete Data via the EM algorithm", Journal of the Royal Statistics Society, pp 1-38, vol. 39(1), 1977.

[9] S. Deligne, S. Dharanipragada, R. Gopinath, B. Maison, P. Olsen, H. Printz, "A robust high-accuracy speech recognition system for mobile applications", IEEE Transactions on Speech and Audio Processing, 10:08, 2002.