LATTICE-BASED UNSUPERVISED MLLR FOR SPEAKER ADAPTATION

Mukund Padmanabhan, George Saon and Geoffrey Zweig

IBM T. J. Watson Research Center P. O. Box 218,
Yorktown Heights, NY 10598

ABSTRACT

In this paper we explore the use of lattice-based information for unsupervised speaker adaptation. As initially formulated, maximum likelihood linear regression (MLLR) aims to linearly transform the means of the gaussian models in order to maximize the likelihood of the adaptation data given the correct hypothesis (supervised MLLR) or the decoded hypothesis (unsupervised MLLR). For the latter, if the first-pass decoded hypothesis is extremely erroneous (as it is the case for large vocabulary telephony applications) MLLR will often find a transform that increases the likelihood for the incorrect models, and may even lower the likelihood of the correct hypothesis. Since the oracle word error rate of a lattice is much lower than that of the 1-best or N-best hypotheses, by performing adaptation against a word lattice, the correct models are more likely to be used in estimating the transform. Furthermore, the particular MAP lattice that we propose enables the use of a natural confidence measure given by the posterior occupancy probability of a state, that is, the statistics of a particular state will be updated with the current frame only if the a posteriori probability of the state at that particular time is greater than a predefined threshold.

Experiments performed on a voicemail speech recognition task indicate a relative 2% improvement in the word error rate of lattice MLLR over 1-best MLLR.

1. INTRODUCTION

Acoustic adaptation is playing an increasingly important role in most speech recognition systems, to compensate for the acoustic mismatch between training and test data, and also to adapt speaker independent systems to individual speakers. Most speech recognition systems use acoustic models consisting of multi-dimensional gaussians that model the pdf of the feature vectors for different classes. A commonly used adaptation technique in this framework is MLLR [3], which assumes that the parameters of the gaussians are transformed by an affine transform into parameters that better match the test or adaptation data. This technique is also often used in unsupervised mode, where the correct transcription of the adaptation data is not known, and a first pass decoding using a speaker independent system is used to produce an initial transcription.

Although MLLR appears to work fairly well even when the unsupervised transcription is mildly erroneous (presumably because of strong parameter tying: often the same transformation is applied to all the gaussians of the acoustic model), it is possible to improve on this performance by taking into account the fact that the initial transcription contains errors. This may be done by considering not just the 1-best transcription produced during the first pass decoding, but the top N candidates. Alternatively, if the first pass decoding produces a word graph, this can be used as the reference word graph, instead of the 1-best or N-best reference transcriptions. We describe a formulation that affinely transforms the means of the gaussians to maximize the log likelihood of the adaptation data under the assumption that a word graph is available that represents all possible word sequences that correspond to the adaptation data. The word graph is produced during a first pass decoding with speaker independent models. It is also possible to consider only those regions of the word graph that represent a high confidence of being correct to further improve the performance. This use of confidence to guide training or adaptation is similar in spirit, but different in its use of MAP-lattice posteriors, from recent work by, e.g. [12, 10, 11].

In Section 2, we describe the theoretical aspect of the formulation, in Section 3, we describe the first pass decoding strategy that is used to produce the word graphs, in Section 4, we describe a confidence related pruning method that enables regions of low confidence to be discarded, and finally in Section 5, we describe the results of experiments on a Voicemail corpus.

2. THEORETICAL FRAMEWORK

Notation: y_t denotes the multi-dimensional observation at time t, y_1^T denotes the T observations corresponding to the adaptation data. The pdf of each context dependent phonetic state s is modeled by mixtures of gaussians, each with a mean and diagonal covariance μ_s, Λ_s. θ is used to indicate the current values of the gaussian parameters, and $\hat{\theta}$ is used to denote the future values (to be estimated). The probability density of the observation y_t given the pdf of state s is denoted $p_\theta(y_t/s)$. In this paper, we will assume that θ and $\hat{\theta}$ are related in the following way: $\hat{\mu}_s = A\mu_s$, $\hat{\Lambda}_s = \Lambda_s$, i.e. only the current means of the gaussians are linearly transformed, and all means are transformed by the same matrix A.

In the regular MLLR framework, the problem is defined as follows: find $\hat{\theta}$ (or equivalently A) so that the log likelihood of the adaptation data, y_1^T is maximized, assuming that

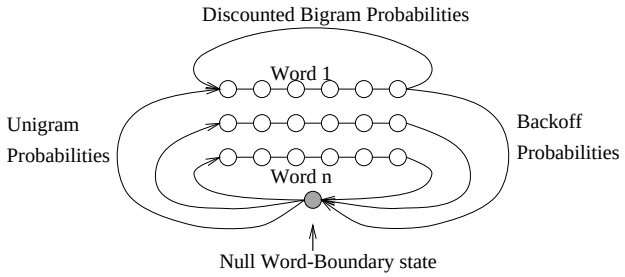

Figure 1: HMM structure used to generate MAP lattices. This HMM uses word internal acoustic context and the inter-word transition arcs encode a Kneser-Ney bigram language model.

w is the transcription corresponding to the adaptation data. The transcription w can be represented as a sequence of K states $s_{w,1} \cdots s_{w,K}$, and the T observation frames can be aligned with this sequence of states. However, the alignment of the T frames with the sequence of states is not known. Let s_t denote the state at time t. The objective can now be written as

$$\begin{aligned} \hat{\theta}^* &= \operatorname*{argmax}_{\hat{\theta}} \log[p_{\hat{\theta}}(y_1^T)] \\ &= \operatorname*{argmax}_{\hat{\theta}} E_{s_1^T/y_1^T,\theta} \log[p_{\hat{\theta}}(y_1^T)] \\ &\equiv \operatorname*{argmax}_{\hat{\theta}} \sum_{s_1^T} p_\theta(s_1^T/y_1^T) \log[p_{\hat{\theta}}(y_1^T, s_1^T)] \end{aligned} \quad (1)$$

In our proposed lattice-based MLLR, we assume that the word sequence corresponding to the adaptation data cannot be uniquely identified and incorporate this uncertainty in the form of a lattice or word graph. The word graph is produced by a first pass decoding with speaker independent models. The formulation of the maximum likelihood problem is identical to (1) with one big difference. In (1), the states s_t were assumed to belong to the alphabet of K states $s_{w,1} \cdots s_{w,K}$, with the only allowed transitions being $s_i \rightarrow s_i$ and $s_i \rightarrow s_{i+1}$. In the lattice-based MLLR formulation, the transitions between the states is dictated by the structure of the word graph. Additionally, it is possible to take into account the language model probabilities also (which are ignored in the MLLR formulation), by incorporating them into the transition probability corresponding to the transition from the final state of a word in the word graph to the initial state of the next connected word in the word graph.

3. FIRST PASS WORD GRAPH GENERATION

We tested lattice-based MLLR in the context of a multipass lattice decoder recently developed at IBM. In the first pass, we generate a Maximum A-Posteriori Probability word lattice (MAP lattice) [4, 13] using word internal acoustic models and a bigram language model. To construct a MAP lattice, we assume that the utterance is produced by an HMM with a structure as shown in Figure 1. Each pronunciation variant in the vocabulary appears as a linear sequence of phones in the HMM, and the structure of this model permits the use of word-internal context dependent

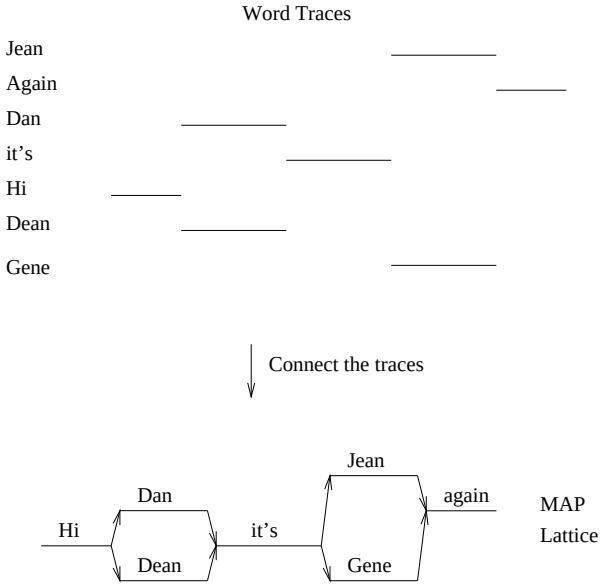

Figure 2: Word traces produced by the MAP lattice HMM, and their connection into a word lattice. In reality, since the N-best words at each frame are output, a vertical line should intersect a constant number of word traces; for visual simplicity, we have simplified the picture.

phones. We use a bigram language model with modified Kneser-Ney smoothing [5, 6], and this factors naturally as shown in Figure 1. There is an arc from the end of each word to a null word-boundary state, and this arc has a transition probability equal to the back-off probability for the word. From the word-boundary state, there is an arc to the beginning of each word, labeled with the unigram probability. For word pairs for which there is a direct bigram probability, we introduce an arc from the end of the first word to the beginning of the second, and this arc has a transition probability equal to the discounted bigram probability. We normalize the dynamic range of the acoustic and language-model probabilities by using an appropriate language model weight, typically 15.

The MAP lattice is constructed by computing the posterior state occupancy probabilities for each state at each time:

$$P(S_t = s|y_1^T) = \frac{\alpha_s^t \beta_s^t}{P(y_1^T)}$$

where $\alpha_s^t = P(y_1^t, S_t = s)$ and $\beta_s^t = P(y_{t+1}^T|S_t = s)$, and then computing posterior word occupancy probabilities by summing over all the states interior to each word. That is, if $\mathcal{W}_i$ is the set of states in word W_i, we compute

$$P_t(W_i) = \sum_{s \in \mathcal{W}_i} \frac{\alpha_s^t \beta_s^t}{P(y_1^T)}$$

at each time frame. We then keep track of the N likeliest words at each frame, and output these as a first step in the processing.

Note that a word will be on the list of likeliest words for a period of time, and then fall off that list. Thus the output of the first step is essentially a set of word traces, as illustrated

in Figure 2. The horizontal axis is time, and the vertical axis ranges over all the pronunciation variants.

The next step is to connect the word traces into a lattice. Many connection schemes are possible, but we have found the following simple strategy to be quite effective. It requires that one more quantity be computed as the word traces are generated: the temporal midpoint of each trace as computed from the first moment of its posterior probability:

$$\frac{\sum_{t=start}^{t=end} t P_t(W)}{\sum_{t=start}^{t=end} t}$$

To construct an actual lattice, we add a connection from the end of one word trace to the beginning of another if the two overlap, and the midpoint of the second is to the right of the midpoint of the first. This is illustrated at the bottom of Figure 2. (We have also found it convenient to discard traces that do not persist for a minimum period of time, or which do not reach an absolute threshold in posterior probability.)

To evaluate our lattices, we computed the oracle word-error rate, i.e. the error rate of the single path through the lattice that has the smallest edit distance from the reference script. This is the best word-error rate that can be achieved by any subsequent processing to extract a single path from the lattice. For voicemail transcription, the MAP lattices have an oracle word error rate of about 9%, and the ratio of the number of word occurrences in the lattices to the number of words in the reference scripts is about 64. Due to the rather lax requirements for adding links between words, the average indegree for a word is 74. The MAP lattice that is produced in this way is suitable for a bigram language model: the arcs between word-ends can be labeled with bigram transition probabilities, but is too large for a straightforward expansion to trigram context. In order to slim it down, we make a second pass, where we compute the posterior probability of transitioning along the arcs that connect word-traces. That is, if s_i is the last state in one word trace and s_j is the first state in a successor and a_{ij} is the weighted language model transition probability of seeing the two words in succession, we compute

$$P(S_t = s_i, S_{t+1} = s_j | y_1^T) = \frac{\alpha_{s_i}^t \beta_{s_j}^{t+1} a_{ij} b_j(y_{t+1})}{P(y_1^T)}$$

This is the posterior probability of being in state s_i at time t and in state s_j at time $t+1$, and transitioning between the words at an intermediate time. For each link between word traces, we sum this quantity over all time to get the total probability that the two words occurred sequentially; we then discard the links with the lowest posteriors. It should be noted that a separate quantity is computed for every link in the lattice. Thus, even if two links connect traces with the same word labels, the links will in general receive different posterior probabilities because the traces will lie in different parts of the lattice, and therefore tend to align to different segments of the acoustic data. As in [7], we have found that over 95% of the links can be removed without a major loss of accuracy. Our pruned lattices have an average indegree a little under 4, and an oracle error rate of about 11%. After pruning, we expand the lattices to trigram context, and compute the posterior state occupancy probabilities needed for MLLR with a modified Kneser-Ney trigram language model, and left-word context dependent acoustic models.

4. CONFIDENCE PRUNING

Word lattices have been used in a variety of confidence estimation schemes [8, 9], and in our work, we used the simplest possible measure - posterior phone probability - to discard interpretations in which we had low confidence. Recall that as a first step in MLLR, we compute the posterior gaussian probabilities for all the gaussians in the system. We compute this on a phone-by-phone basis, first computing the posterior phone probability, and then multiplying by the relative activations for the gaussians associated with the phone. For phone s_i with gaussian mixture $\mathcal{G}_i$, and for a specific time frame y_t,

$$P(G_t = g_j | y_1^T) = P(S_t = s_i | y_1^T) \frac{g_j(y_t)}{\sum_{g \in \mathcal{G}_i} g(y_t)}$$

Since the gaussian posteriors are used to define a set of linear equations that are solved for the MLLR transform, it is reasonable to assume that noisy or uncertain estimates of the posteriors will lead to a poor estimate of the MLLR transform. To examine the truth of this hypothesis, we estimated the MLLR transform from subsets of the data, using only those estimates of $P(S_t = s_i | y_1^T)$ that were above a threshold, typically 0.7 to 0.9.

5. EXPERIMENTS AND RESULTS

The experiments were performed on a voicemail transcription task [1]. The speaker independent system has 2313 context dependent states called *leaves* (of the context decision tree) and 134K diagonal mixture components and was trained on approximatively 70 hours of data. The feature vectors are obtained in the following way: 24 dimensional cepstral vectors are computed every 10ms (with a window size of 25ms). Every 9 consecutive cepstral vectors are spliced together forming a 216 dimensional vector which is then projected down to 39 dimensions using heteroscedastic discriminant analysis and maximum likelihood linear transforms [2].

The test set contains 86 randomly selected voicemail messages (approximately 7000 words). For every test message, a first-pass speaker independent decoding produced a MAP word lattice described in section 3. For the MLLR statistics we used gaussian posteriors as described in section 4. The regression classes for MLLR were defined in the following way: first all the mixture components within a state were bottom-up clustered using a minimum likelihood distance and next, the representatives for all the states were clustered again until reaching one root node. The number of MLLR transforms that will be computed depends on the number of counts that particular nodes in the regression tree get. In

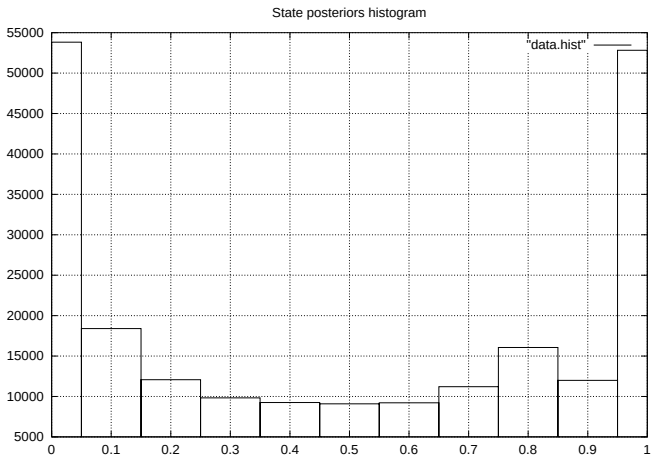

Figure 3: Histogram of the state posterior probabilities.

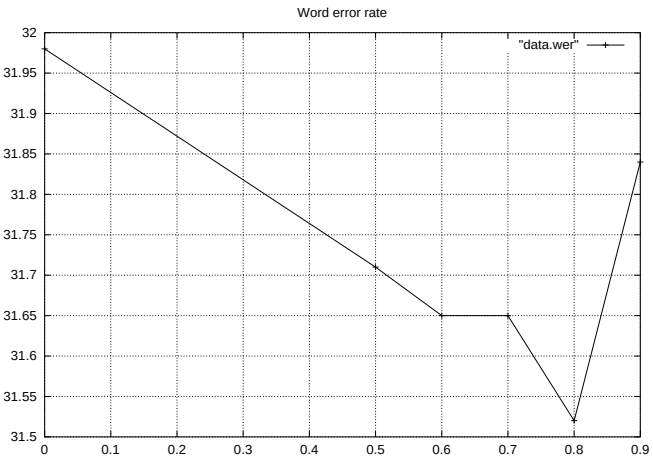

Figure 4: Word error rate versus confidence threshold.

practice, a minimum threshold of 1500 was found to be useful. For voicemail messages which are typically 10 to 50 seconds long this results in computing 1-3 transforms per message.

Figure 3 shows the histogram of the non zero phone posteriors computed over all the test sentences. There are two things to note. First, there are a significant number of entries with moderate (0.1-0.9) probabilities. Secondly, although there are a significant number of entries at the left-end of the histogram, they have such low probabilities that they account for an insignificant amount of probability mass. This suggests that we can use high values for the confidence thresholds on the posteriors without loosing too much adaptation data.

Figure 4 shows the word error rate as a function of the confidence threshold. The optimal results were obtained for a threshold of 0.8. Increasing the threshold above this value results in discarding too much adaptation data which counters the effect of using only alignments that we are very confident in.

Finally, Table 1 compares the word error rates of the speaker independent system, 1-best MLLR, lattice MLLR and confidence-based lattice MLLR. The overall improvement of the confidence-based lattice MLLR over the 1-best MLLR is only about 1.8% relative but has been found to be consistent across different test sets, with the same 80% confidence threshold. We expect the application of iterative MLLR, i.e. repeated data-alignment and transform estimation, to increase the differential. This is because the lattice has more correct words to align to than the 1-best transcription. For comparison, [11] cites a gain on the Wall Street Journal task of 3-4% relative over standard mllr by combining confidence measures with mllr.

System	WER
Baseline (SI)	33.72%
1-best MLLR	32.14%
Lattice MLLR	31.98%
Lattice MLLR + thresh.	31.56%

Table 1: Word error rates for the different systems.

6. CONCLUSION

In this paper we explored the use of a word lattice in conjunction with MLLR. Rather than adjusting the gaussian means to maximize the likelihood of the data given a single decoded script, we generated a transform that maximized the likelihood of the data given a set of word hypotheses concisely represented in a word lattice. We found that the use of a lattice alone produces a very small improvement, but that we can gain a more significant improvement by discarding statistics in which we have low confidence.

REFERENCES

[1] M. Padmanabhan, G. Saon, S. Basu, J. Huang and G. Zweig. Recent improvements in voicemail transcription. *Proceedings of EUROSPEECH'99*, Budapest, Hungary, 1999.

[2] G. Saon, M. Padmanabhan, R. Gopinath and S. Chen. Maximum likelihood discriminant feature spaces. to appear in *Proceedings of ICASSP'2000*, Istanbul, 2000.

[3] C.J. Leggetter and P. Woodland. Speaker Adaptation of Continuous Density HMMs Using Multivariate Linear Regression. *Proceedings of ICSLP'94*, Yokohama, Japan, 1994.

[4] F. Jelinek. *Statistical Methods for Speech Recognition*. The MIT Press, 1997.

[5] R. Kneser and H. Ney. Improved Backing-off for n-gram Language Modeling. *Proceedings of ICASSP'95*. 1995.

[6] S.F. Chen and J. Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. Center for Research in Computing Technology, Harvard University, 1998.

[7] L. Mangu and E. Brill. Lattice Compression in the Consensual Post-Processing Framework. *Proceedings of SCI/ISAS*, Orlando, Florida, 1999.

[8] T. Kemp and T. Schaff. Estimating Confidence using Word Lattices. *Proceedings of ICASSP'97* 1997.

[9] G. Evermann and P.C. Woodland. Large Vocabulary Decoding and Confidence Estimation using Word Posterior Probabilities. *Proceedings of ICASSP'00* 2000.

[10] M. Finke, J. Fritsch, P. Geutner, K. Ries, M. Westphal, T. Zeppenfeld, and A. Waibel. 1997 Hub-5e Eval System Evaluation. http://isl.ira.uka.de/ISL.speech.lvcsr.html

[11] F. Wallhoff, D. Willett and G. Rigoll. Frame-Discriminative and Confidence-Driven Adaptation for LVCSR. *Proceedings of ICASSP'00 2000.*

[12] T. Kemp and A. Waibel. Unsupervised Training of a Speech Recognizer: Recent Experiments *Proceedings of EUROSPEECH'99*, Budapest, Hungary, 1999.

[13] G. Zweig and M. Padmanabhan. Exact Alpha-Beta Computation in Logarithmic Space with Application to MAP Word Graph Construction *Proceedings of ICSLP'00* Beijing, China, 2000.