Unsupervised Lattice-based Acoustic Model Adaptation for Speaker-Dependent Conversational Telephone Speech Transcription

K. Thambiratnam and F. Seide

Microsoft Research Asia, 5F Sigma Center, 49 Zhi Chun Road, Beijing, P.R. China 100080

{kit, fseide}@microsoft.com

Abstract

This paper examines the application of lattice adaptation techniques to speaker-dependent models for the purpose of conversational telephone speech transcription. Given sufficient training data per speaker, it is feasible to build adapted speaker-dependent models using lattice MLLR and lattice MAP. Experiments on iterative and cascaded adaptation are presented. Additionally, various strategies for thresholding frame posteriors are investigated, and it is shown that accumulating statistics from the local best-confidence path is sufficient to achieve optimal adaptation. Overall, an iterative cascaded lattice system was able to reduce WER by 7.0% abs., which was a 0.8% abs. gain over transcript-based adaptation. Lattice adaptation reduced the unsupervised/supervised adaptation gap from 2.5% to 1.7%.

Index Terms: Unsupervised Acoustic Model Adaptation, Conversational Speech Recognition

1. Introduction

As storage costs continue to decrease, so too does the feasibility of collecting hundreds and even thousands of hours of domain-dependent audio data for a multitude of tasks. How to best use this data for speech recognition without requiring expensive manual transcripts remains an open question. In particular, as telephony enabled smart devices become more prevalent, it is envisioned that recording conversations on these devices will provide a convenient means for collecting high-quality, speaker-labeled speech data for acoustic model training and adaptation. This paper investigates the application of lattice-based adaptation techniques to harness such data for conversational speech transcription.

The ability to provide stable adaptation even in the presence of recognition errors is important for unsupervised adaptation. The typical approach of using an unmodified automatically recognized transcript as the supervision for adaptation is problematic in this sense, though experimentally it has been shown that both Maximum Likelihood Linear Regression (MLLR) [1] and Maximum A Posteriori (MAP) [2] adaptation still perform well. A better approach is to use utterance-level confidence scores to discard poorly transcribed segments. This has met with some success [3], but results in entire utterances being wastefully thrown away due to a short span of poorly recognized words. This reduces the effective amount of adaptation data and thus the overall adaptation gains. A number of works have also demonstrated gains by discarding or confiding weighting data at the word level [4][5]. Such approaches still result in unnecessary discounting of data, since decisions are made on a whole word/phone basis.

The ideal solution would be to weight data at the frame-level. This is in fact already indirectly done by the forward-backward algorithm used in MLLR and MAP adaptation implementations. In forward-backward, the state occupancies (or posteriors) are computed at each frame. In areas of mis-recognition the occupancies would be smeared across a number of adjacent states, resulting in a smaller contribution of those mis-recognized frames to adaptation. This is however a serendipitous outcome of forward-backward for unsupervised adaptation - considering transcript errors is not part of the formulation for forward-backward.

A more correct approach is to consider all possible word hypotheses when computing frame posteriors. This is the approach in lattice adaptation, where the lattice path posterior weighted occupancy is accumulated at each frame. Lattice adaptation has been investigated in a number of past works, including [6, 7, 8] and has been shown to yield gains over transcript adaptation.

A benefit of lattice adaptation is frame-level weighting. Consider a case where the word STOCK (s.t.aa.k) is misrecognized as STOP (s.t.aa.p). In transcript adaptation, the p model and adjacent states would be incorrectly allocated frames of k. Word-level confidence thresholding methods would likely discard all frames for the word resulting in less adaptation data. In contrast for lattice adaptation it is likely that ‘STOCK’ appears in the lattice, and thus some probability would be assigned to the k model. In some cases the probability mass may even exceed that assigned to p if sufficient alternative paths exist in the lattice. In this way, lattice adaptation is able to better use adaptation data while gracefully handling local errors.

This paper focuses specifically on the application of lattice adaptation techniques to building speaker-dependent models for the purpose of conversational telephone speech transcription. Here, it is feasible to have larger amounts of speaker-labeled data that is collected via a telephony-enabled smart device (data is recorded prior to transmission to avoid compression/transmission artifacts). The larger amount of data allows both lattice versions of MAP and MLLR adaptation to be investigated, as well as iterative and cascaded adaptation configurations. Additionally, the paper reports on investigative experiments aimed at better understanding the contributions of the lattice to adaptation. In particular, it examines the use of lattice confidence thresholding and demonstrates that thresholding is unnecessary but that accumulating statistics from the local best-confidence path is sufficient to achieve optimal adaptation.

An introduction to lattice adaptation is presented in section 2, followed by implementation issues in section 3. Experiment results are reported in section 4 and conclusions in section 5.

2. Lattice Adaptation

Popular implementations of MLLR and MAP adaptation methods operate on an utterance transcript, which for unsupervised...
adaptation, is erroneous. Given a set of adaptation recordings, the transcript (more precisely the state sequence represented by the transcript) and the corresponding acoustic data is used to estimate occupancy statistics or frame posteriors, $\gamma_k(t)$, which is the probability of being in the $k$-th state at frame $t$. These frame posteriors are then used to compute parameter updates.

It is completely valid to substitute an alternative method for computing frame posteriors and still use the same adaptation formulae, as long as the physical meaning of $\gamma_k(t)$ is not changed. This is the case for lattice adaptation, where frame posteriors are estimated from a lattice (or rather the multitude of state sequences represented by the lattice). This results in a different estimate for $\gamma_k(t)$ but allows the same adaptation techniques to be used subsequently.

Frame posterior values can be estimated for an observation sequence $O = [o_1 \ldots o_T]$ and state sequence $S' = [s'_1 \ldots s'_L]$ using the well-known forward-backward algorithm. For convenience, $s_k$ is used to represent $s'_k$, $k$ being a 2-tuple index. Let $\alpha_k(t)$ be the probability of being in state $s_k$ at time $t$ and having observed the observation sequence $[o_1 \ldots o_t]$. Correspondingly, let $\beta_k(t)$ be the probability of being in state $s_k$ at time $t$ and then observing frame sequence $[o_{t+1} \ldots o_T]$, where $T$ is the number of frames in an adaptation utterance. Note both $\alpha_k(t)$ and $\beta_k(t)$ do not include language model probabilities, only state emission and transition probabilities. Then

$$\gamma_k(t) = \frac{\alpha_k(t)\beta_k(t)p(o_1|\lambda_{s_k})p_t(S')}{P(O)} \tag{1}$$

where $P(O)$ is the total probability of the observation sequence, $p(o_1|\lambda_{s_k})$ is the emission probability of $o_1$ from state $s_k$, and $p_t(S')$ is the language model probability of the word sequence of $S'$. Note that conditional dependence on $\lambda_{s_k}$ is omitted here for readability. $P(O)$ is given by

$$P(O) = \sum_e p(O|S') p_t(S') \tag{2}$$

where $S'$ is the set of all possible state sequences for $O$ contained in the lattice. The same form can be used for transcript-based forward-backward by simplifying equation 2 to a single path consisting of the transcript state sequence. In this case, $p_t(S')$ cancels out to give the familiar forward-backward equation $\alpha_k(t)\beta_k(t)p(o_1|\lambda_{s_k})$.

The individual $\gamma_k(t)$ values can then be folded across time, paths, and all instances of the same physical state, to give $\Gamma_q = \sum_t \Gamma_q(t)$ and $\Gamma_q(t) = \sum_{s_k \in V} \gamma_k(t)$, where $V_q$ is a physical state in the acoustic model, and $s_k$ is an individual occurrence of that state on a state sequence $S'$. Adaptation updates can then be computed using the standard MLLR and MAP formulae.

3. Lattice Adaptation Implementation

3.1. Fixed word boundaries

Lattice forward-backward requires a full forward-backward accumulation for each unique state sequence in a lattice. Most lattices have a large number of paths and so this is computationally infeasible. A seemingly reasonable approximation is to fix the word or phone boundaries in the lattice and then perform forward-backward only within each fixed arc. In the case where word boundaries are fixed, equation 1 then becomes

$$\gamma_k(t) = \frac{\alpha'_k(t)\beta'_k(t)p(o_1|\lambda_{s_k})A_kB_kp_t(S')}{P(O)} \tag{3}$$

$$\gamma_k(t) = \frac{\alpha'_k(t)\beta'_k(t)p(o_1|\lambda_{s_k})}{p_o(e)} \times p_s(e) \tag{4}$$

where $e$ is the lattice edge that contains the state $s_k$, $p_o(e)$ is the total acoustic probability in edge $e$, and $p_s(e) = A_kB_kp_o(e)p_t(S')/P(O)$ is the well-known lattice edge posterior. Additionally $\alpha'_k(t)$ and $\beta'_k(t)$ are the same as $\alpha_k(t)$ and $\beta_k(t)$ but computed on the reduced observation sequence for edge $e$ and its corresponding state sequence. This essentially becomes an isolated forward-backward on each lattice arc, $e$, weighted by the edge posterior $p_s(e)$, as shown in equation 4.

Fixing edge boundaries means that estimating frame posteriors from a transcript is no longer the single-path equivalent of lattice adaptation. Performing forward-backward on the transcript means that all time alignments will be considered for all words, whereas doing so on a single-path lattice using fixed word boundaries will only consider local state alignment variations within a single arc. To better understand the ramifications of this, a small scale experiment was conducted where a fixed word boundary lattice-based implementation was used to perform lattice MAP on lattices containing only the transcript as a single path. It was found that the resulting Word Error Rate (WER) was 7% relative worse than using full-forward backward on the transcript. The word boundary limitation was relaxed by introducing a tolerance of $\pm \Delta$ frames on each edge. It was found that as much as 1 second (100 frames) of tolerance was required to achieve the same WER as full-forward backward.

However, once multiple paths are included in the lattice, multiple instances of the same word with different alignments typically occur in the lattice, reducing the above problem. The problem is not completely eliminated though because typical state of the art decoders discard lattice paths for the same word with the same context, and thus redundancy in alignments is only maintained for the same word with different contexts. In a context-dependent system, this means boundary phones will be different states and will thus be counted separately. In the above experiments, introducing the multi-path lattice resulted in WER reduction over transcript based adaptation.

This indicates fixing word boundary times does hurt the achievable gain, but is compensated for by word redundancies (with different contexts) in the lattice. Another small scale test was performed for lattice MAP where time boundary tolerance was allowed for a full lattice forward-backward. This was computationally and memory intensive, and thus could only be computed for a small set and only for tolerance up to $\pm 0.1$ seconds (10 frames). It was found that there was no significant gain from doing this, and particularly when compared to using the same tolerance for the single-path experiments described above.

3.2. Iterative adaptation

Unsupervised adaptation can be performed iteratively so that successive iterations benefit from model improvements. Iterative MLLR adaptation is straightforward - the adapted model from one iteration can be used as the input model for the following iteration. Iterative MAP requires more care. Here, the previous iteration’s model should be used for estimating occupancies in the following iteration but the initial unadapted model should be used as the prior model for parameter updating. This is the theoretically sound approach to iterative MAP adaptation, since the prior model is fixed in MAP. Experimentation found that failing to use the unadapted model as the prior resulted in very unstable iterative MAP adaptation.

Iterative adaptation is costly, as it requires a new decoder pass after each iteration to update the supervision transcript (or lattice in the case of lattice adaptation). For lattice adaptation though, it was found that redecoding after adaptation resulted
in very sparse lattices. This is because the adapted model is significantly better matched to the adaptation data. Using this resulting sparse lattice for adaptation resulted in the best path dominating frame posteriors. As discussed in section 3.1, best path only lattice adaptation is suboptimal. Broadening the decoding beam does not alleviate this since obviously the best path posterior will continue to dominate. Thus redecoding was not performed between each iteration.

In order to remain consistent with lattice adaptation experiments, transcript adaptation experiments were also run without redecoding at each iteration. Interestingly this did not affect accuracy at all. In all evaluated cases, the loss from not redecoding between iterations was less than 0.1% absolute i.e. negligible. This makes iterative adaptation (both transcript and lattice based) significantly less expensive computationally. However, it means that subsequent iterations only benefit from improved state alignments rather than corrected supervision state sequences.

### 3.3. Confidence Thresholding

Confidence thresholding is commonly applied (e.g. [7][8]) to frame posteriors to discard low confidence states and is motivated as a means to prune mis-recognized spans of speech. From a practical sense, this appears reasonable, but from a theoretical maximum likelihood sense, does not have a sound foundation. Typically thresholding is applied on the state level, however, it is well known that posteriors estimated from longer context are more stable. A multitude of thresholding approaches were explored here. At each time, $t$, $\gamma(t)$ is computed for all active states and then zeroed using one of:

1. state: $\gamma(t) < \tau$
2. triphone: $\sum_{k' \in T(k)} \gamma(t) < \tau$, where $T(k)$ is the set of all states in the triphone that state $k$ belongs to. This pools the decision across all instances of the same triphone at the current frame.
3. word: $\sum_{k' \in W(k)} \gamma(t) < \tau$, where $W(k)$ is the set of all states in the word of state $k$. This pools the decision across all instances of a word, including alternate pronunciations. This is not a super-set of item 2 since pooling will not occur for the same state in different words.

### 4. Experiments and Results

Experiments were performed on a 62-speaker subset of the Switchboard-1 (SWB1) conversational telephone speech corpus. Sixty minutes of adaptation data (about 13 conversation sides), and ten minutes of evaluation data were selected per speaker. A speaker-dependent acoustic model was then created per speaker using the associated adaptation data and evaluated on each speakers’ evaluation data.

Baseline acoustic (72-mixture ML-trained HMM set) and trigram language models were trained on 1700h of data comprising The Fisher English Corpus Part 1 and 2 (Fisher). All experiments used a 22.6k vocabulary. The baseline WER on the SWB Eval2000 evaluation data set was 28.1%. The adaptation and evaluation data used in experiments were more difficult, with a WERs of 34.9% and 34.0% respectively. Lattice adaptation experiments used lattices decoded with a beam-width of 160 and then pruned to 120.

Three adaptation approaches were examined: global MLLR (glob), 256-class regression tree MLLR (rtree), and MAP adaptation. Only mean parameters were updated. Global adaptation used a 2-class speech/non-speech regression tree. Lattice systems used an acoustic model scale equal to the inverse of the decoding language model scale. The results clearly show that lattice adaptation equals or out-performs transcript adaptation. There are no gains for global adaptation, which is to be expected since global adaptation pools all the data and thus is more robust to local errors. However, there are notable gains of 0.7% absolute for both rtree and MAP adaptation. More importantly, the gap between supervised and unsupervised is closed from 2.2%/1.6% to 1.5%/0.9% for rtree/MAP respectively, which means the loss of using unsupervised over supervised adaptation is reduced from 31.8%/44.4% to 21.7%/25.0% for rtree/MAP respectively.

It is also interesting to note the behavior of the various systems during iterations. All systems achieved significant gains with multiple iterations, except for MAP. This is without any update to the transcript or lattice, and thus is solely from improved state alignments. The gains for the global MLLR systems are particularly interesting, as in these systems, adaptation statistics are pooled and thus any gains must be from improving the alignments at speech/silence boundaries only. To date, no proper explanation for this has been determined but the matter continues to be investigated.

### 4.1. Confidence Thresholding

To better understand the role of thresholding, further experiments on unsupervised MAP adaptation were performed. Experiments were done on MAP instead of MLLR as it has more free parameters and therefore more susceptible to errors. Experiments for lattice MAP used word, phone or state thresholding, while transcript MAP used only state thresholding. Figure 1 shows the results of these experiments. The results clearly show that word thresholding is the most stable and consistently converges to the best WER across all thresholds. State and phone thresholding systems are worse and in some cases get worse with more iterations, though it should be noted that the deltas here are small ($\pm 0.1\%$ abs.). Using lower thresholds of 0.2 and 0.5 is slightly better than 0.8, and 0.5 systems tend to converge quicker. Also of note is that the non-thresholded system also converged to an almost optimal system. From a theoretical sense, this demonstrates that there is no need for the empiri-

<table>
<thead>
<tr>
<th>Method</th>
<th>H1</th>
<th>H2</th>
<th>H4</th>
<th>H13</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>34.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>glob</td>
<td>31.2</td>
<td>30.7</td>
<td>30.5</td>
<td>30.3</td>
</tr>
<tr>
<td></td>
<td>rtree</td>
<td>28.6</td>
<td>27.6</td>
<td>27.2</td>
<td>27.1</td>
</tr>
<tr>
<td></td>
<td>MAP</td>
<td>30.5</td>
<td>30.4</td>
<td>30.4</td>
<td>30.4</td>
</tr>
<tr>
<td></td>
<td>lat-glob</td>
<td>31.8</td>
<td>31.3</td>
<td>31.1</td>
<td>30.9</td>
</tr>
<tr>
<td></td>
<td>rtree</td>
<td>31.7</td>
<td>31.4</td>
<td>31.1</td>
<td>30.9</td>
</tr>
<tr>
<td></td>
<td>MAP</td>
<td>30.0</td>
<td>29.5</td>
<td>29.2</td>
<td>29.3</td>
</tr>
<tr>
<td></td>
<td>lat-rtree</td>
<td>29.7</td>
<td>29.0</td>
<td>28.6</td>
<td>28.6</td>
</tr>
<tr>
<td></td>
<td>MAP</td>
<td>32.0</td>
<td>32.0</td>
<td>32.0</td>
<td>32.0</td>
</tr>
<tr>
<td></td>
<td>lat-MAP</td>
<td>31.7</td>
<td>31.6</td>
<td>31.5</td>
<td>31.3</td>
</tr>
</tbody>
</table>

Table 1: Single-stage WER results, 60mins adapt 13 iterations, 10mins eval data. Loss is WER gain for unsupervised relative to equivalent supervised system. S/U indicates supervised/unsupervised.
Also of note is the fact that thresholded transcript MAP does not yield any significant gains. Thus, simply pruning bad frames is not sufficient for realizing the gains in lattice MAP. The alternative state sequences clearly add value for lattice adaptation. Also, the 0.5 and 0.8 thresholded systems performed similarly to the 0.2 systems. However, in those systems there would be at most one dominating state/phone/word at each frame, since occupancies must sum up to 1. This suggests that using the local (not global/transcript) best path only, and not considering alternatives at each frame is sufficient to get optimal adaptation. To validate this a series of experiments were conducted where the lattice was used for estimating occupancies, but then only states from the global best path (transcript) were used for parameter updates, subject to thresholding. It was found that these hybrid systems had a similar behavior to the transcript-only MAP systems in 1 but with marginally lower WERs of 31.7%. One can conclude then that the gains in lattice MAP stem from considering locally optimal paths on and off the global best path and from discarding/weighting the contribution from low-confidence frames. Furthermore, they show that errors in occupancy estimates are more random rather than systematic, since accumulating low occupancies did not affect the final model accuracy.

### 4.2. Cascaded adaptation

A final set of experiments were performed to evaluate lattice adaptation in a cascaded setup. The common cascade setup of global MLLR followed by regression-tree MLLR and then MAP adaptation was evaluated. Thirteen iterations were performed at each stage, and the final iteration model was used as the input for the next stage. The results of these experiments are shown in table 2. Once again, it can be seen that lattice adaptation provided gains, achieving a 0.8% abs. gain over the best unsupervised transcript system. The best lattice system had a WER of 28.0%, compared to 26.3%/28.8% for the best supervised/unsupervised transcript systems respectively. Interestingly the lattice needed to appear only once in the cascade to yield its benefits, as shown by the fact that the glob+rtree+latMAP, glob+latrtree+latMAP and glob+lattree+latMAP systems all had approximately the same result. It is not clear why this is the case, since the glob+lattree and glob+latMAP systems both had gains of 0.8% and 0.6% abs. Overall, the best cascaded lattice system reduced supervised/unsupervised loss by 9.2%.

### 5. Conclusion

The reported experiments on conversational speech transcription have demonstrated the modest value of the lattice for unsupervised acoustic adaptation. A 3-stage cascaded lattice MLLR+MAP system was able to reduce the word error rate by 7.0% absolute over an unadapted baseline of 34.0% for a Switchboard-1 conversational transcription task. This was a 0.8% abs. gain over the corresponding transcript-only adaptation setup and reduced the loss of using unsupervised compared to supervised adaptation from 32.2% to 22.2%.

Experiments to better understand the contribution of the lattice to adaptation were also reported on. It was shown that the common practice of lattice confidence thresholding was not necessary to achieve optimal accuracy, given sufficient iterations were performed. Furthermore it was shown that using the local (not global/transcript) best path only, and not considering alternatives at each frame is sufficient to get optimal adaptation. However, some minor gains in adaptation stability can be achieved if word-based thresholding is used. Finally, it was shown that lattice adaptation gains preserve themselves across adaptation approaches as well as cascaded adaptation stages.

### 6. References


