Approaches to Speech Recognition based on Speaker Recognition Techniques

Daniel Povey
Microsoft, One Microsoft Way,
Redmond, WA 98052
dpovey@microsoft.com
(work was performed at IBM)

Stephen M. Chu, Jason Pelecanos, Hagen Soltau
IBM T.J. Watson Research Center,
1101 Kitchawan Rd, Yorktown Heights, NY
{schu,jwpeleca,hsoltau}@us.ibm.com

Abstract

We have experimented with approaches to speech recognition that are inspired by work from the speaker recognition community, including an approach that was used in IBM's best speech recognition submissions in its January 2009 evaluation. Here we explain in general terms the techniques used, without the full technical details. We initially used an approach based on Maximum a Posteriori (MAP) adaptation of Gaussian Mixture Models; this approach gave improvements in a Maximum Likelihood system but did not seem promising to combine with discriminative training. More recently we have used a different approach based on a subspace adaptation of a Gaussian Mixture Model; this gave improvements over a standard system even when combined with discriminative training. It gave very substantial improvements when trained with limited data, which may relate to the smaller number of parameters needed with this approach.

1 Introduction

A basic approach long used in text-independent speaker recognition is to train a Gaussian Mixture Model (GMM) for each speaker and to verify a test utterance based on the likelihood of the utterance given that speaker's GMM compared with a cohort of other models or a single "universal background model". A drawback of this approach is that it is hard to robustly estimate the parameters of the speaker-specific GMMs given limited data. Therefore, in recent years, approaches have been devised to estimate the parameters more robustly based on Maximum A Posteriori (MAP) (GLAAd94; AFR00) and subspace methods such as nuisance attribute projection (SCB05), within-class covariance normalization (HKS06) and factor analysis (PPNV08). Thes factor analysis method in particular not only accounts for cross-speaker variability but also directly addresses the issue of within-speaker variability such as telephone type and background noise. The same problems that exist in speaker recognition, such as estimation given limited data and irrelevant sources of variation, also exist in speech recognition so it is natural to try to use those speaker recognition techniques in a speech recognition context. We have experimented with two different approaches. The rest of this chapter is split into two sections, with Section 2 summarizing our research on the MAP based approach and Section 3 summarizing the factor analysis based approach which is the one we currently favor.

2 MAP based adaptation

In (DMV08) we introduced a speech recognition technique based on MAP adaptation. Note that although we use the term "adaptation" it is not a speaker adaptation technique, the term refers to the adaptation of models to particular phonetic states. We first train a large mixture of Gaussians (e.g. 750 Gaussians) which we refer to as a "Universal Background Model" – the term is borrowed from speaker recognition. We then MAP adapt this large GMM to each clustered phonetic state. In order to improve the smoothing we actually used a modified form of MAP estimation which took into account

Adaptation	Baseline	MAP	MAP+STC
VTLN	17.9	17.5	16.6
+fMLLR	16.6		15.0
+MLLR	16.2		14.8

Table 1: Results from MAP-based adaptation: Mandarin, dev'07.

the phonetic decision tree, so that a phonetic state is adapted first to phonetic states that are close in the tree. The adapted models were diagonal covariance Gaussians, but we also introduced a separate Semi-tied Covariance (STC) transformation (Gal98) for each of the 750 Gaussians. This helped our experimental system, and in experiments performed later on we were unable to obtain any improvement from this type of technique in our baseline setup. In order to make the memory usage manageable, we introduced a parameter pruning scheme tied to the tree structure.

Table 1 shows the results; see (DMV08) for experimental details. The table shows about 1.5% absolute improvement for all conditions, but the situation is a little more complicated than that. The baselines with fMLLR and MLLR are "normal size" baselines with 500,000 Gaussians (about the size we would build for a normal evaluation system) but the VTLN-only baseline is twice the normal size at one million Gaussians, which helped by 0.8%. This masks the fact that the MAP based system gave much more improvement in the VTLN-only condition. We increased the size of this baseline to provide a fairer comparison to the MAP based system which has a very large number of (smoothed) parameters, several times more than even the largest baseline system. Due to time constraints we did not rerun the fMLLR and fMLLR+MLLR baselines with the larger size, which would have reduced our improvement. Regardless of the exact amount of improvement from the MAP based system, we did not pursue this approach because we believed that it would be too hard to combine this style of system with discriminative training. This is due to its very large number of parameters, which does not combine well with discriminative training (D.04).

3 Subspace adaptation

The style of system we are currently working with is related to the factor analysis approach of (PPNV08). In this approach, the means of a large mixture of Gaussians are concatenated to form a "supervector", and we allow the parameters of our adapted system to vary in a subspace of that vector space, so each phone in context would be represented by a vector in our chosen subspace. For example, we have used a subspace dimension of 50 in a system with feature dimension 40 and 750 Gaussians in the GMM, so we are using 50 out of a possible 30000 dimensions. We also use a separate subspace to represent the nuisance effect of speaker variation, which is analogous to the use of a separate subspace to represent the nuisance effect of channel or session variation in (SCB05; PPNV08). We have added certain extensions to this basic model. One is to replace the use of a single vector to represent each phone in context, with a set of vectors each with their own weights. Each vector represents an adapted mixture of Gaussians, so our extended model represents a mixture of mixtures of Gaussians. We also make the Gaussian mixture weights vary as a function of our subspace vector. The log mixture weights vary linearly with the subspace vector, and we normalize them to sum to one. A further feature of our model is that each of the e.g. 750 basic Gaussians has a full covariance. The covariances are shared between the classes, and are not adapted as in the MAP based approach described above.

3.1 The model

For each acoustic state j and given the speaker s, the probability model $p(\mathbf{x}|j)$ is:

$$p(\mathbf{x}|j) = \sum_{i=1}^{I} w_{ji} \mathcal{N}(\mathbf{x}; \mu_{ji}^{(s)}, \mathbf{\Sigma}_i) \quad (1)$$

$$\mu_{ji}^{(s)} = \mathbf{M}_i \mathbf{v}_j^{+} + \mathbf{N}_i \mathbf{v}^{+(s)} \quad (2)$$

$$w_{ji} = \frac{\exp(\mathbf{w}_i^T \mathbf{v}_j^{+})}{\sum_{i'=1}^{I} \exp(\mathbf{w}_{i'}{}^T \mathbf{v}_j^{+})}, \quad (3)$$

so we describe the acoustic state by the vector $\mathbf{v}_j$ and the speaker by $\mathbf{v}^{(s)}$; the notation $\cdot^{+}$ means appending a 1 to the vector (to handle constant offsets). These vectors have a dimension of about 50. I is

the number of Gaussians in the model being adapted and will typically be about 750. The acoustic states $1 \leq j \leq J$ have the normal range, e.g. several thousand, although this style of system can give its best results with more states than the baseline. The parameters $\mathbf{M}_i$ and $\mathbf{N}_i$ describe the phonetic subspace and the speaker subspace. The weights w_{ji} are a log-linear function of the model weights, normalized to sum to one. The parameters $\mathbf{w}_i$ control the projection from the model subspace to the weights. The covariance matrices $\mathbf{\Sigma}_i$ are full covariance matrices that are not specific to the acoustic state but to the Gaussian index i. Experiments with a subspace representation of the inverse (diagonal) covariance matrix, combined with semi-tied covariance transform to handle more general rotations, failed to show any improvements. On top of the basic model described above, we use the notion of a substate so that a state can be represented by a weighted sum over the models given by different subspace vectors.

The speaker adaptation in this model involves training very few speaker-specific parameters (we use a subspace dimension of typically about 50), so our speaker adaptation is actually done on a per utterance basis. We combine this with conventional fMLLR/constrained MLLR, which is straightforward, and also with MLLR which is less straightforward because it does not combine well with the efficient model evaluation (we have to recompute the normalizing factors for each speaker). For the Arabic results that we report here, we also include an extension of the speaker-adaptation technique that uses more speaker-specific parameters, but we do not describe this here in detail

3.2 Likelihood evaluation

When we evaluate likelihoods given the model, we can arrange the computation so that computing the likelihood for a particular acoustic state j and Gaussian index i involves no more than a dot product in the model subspace. We can then prune the Gaussian indices i to compute only a small subset of the, say, 750 Gaussians. This computation requires us to store a normalizing term for each state j and Gaussian index i, and this normalizing term dominates the memory requirements of the model. Explicitly constructing the projected means in this model is impractical due to memory constraints. The model is about two to four times slower to decode with than a normal system but this gap could easily be eliminated without losing much performance.

3.3 Training with the subspace model

The process of model training consists of training the global parameters $\mathbf{M}_i$, $\mathbf{N}_i$ and $\mathbf{w}_i$ and the state-specific parameters $\mathbf{v}_j$. We initialize the vectors $\mathbf{v}_j$ randomly and then iteratively optimize the projections $\mathbf{M}_i$ and $\mathbf{w}_i$ and the vectors $\mathbf{v}_j$. The speaker subspace is trained in a similar way; after training the main model we initialize the projections $\mathbf{N}_i$ and then iteratively re-estimate the speaker vectors $\mathbf{v}^{(s)}$ and the projections $\mathbf{N}_i$. In general, the statistics required to re-estimate these parameters are quite compact and the re-estimations are fairly straightforward; we get an auxiliary function which is quadratic in the parameters we are trying to optimize and it is easy to optimize. Training requires about the same number of iterations as a normal system (about 30), but the first 10 iterations are done using only one pass over the data because prior to introducing the substates, it is possible to store appropriately pruned statistics which allow us to do many iterations of update without re-accessing the data.

The only part of the optimization which is not absolutely straightforward is the part which relates to the weight projections $\mathbf{w}_i$ and the effect of the weights on the model vectors $\mathbf{v}_j$. This is handled with a suitable quadratic approximation and is not a problem in practice.

3.4 Discriminative training

We were able to combine discriminative training with this style of system. Model-space discriminative training is fairly straightforward, as we can use a version of the Extended Baum-Welch type of technique using weak sense auxiliary functions (D.04). We did introduce some details that are specific to this framework. We use a factor in the learning rate for the state-specific vectors that slows down learning when the discriminative data counts are very small. More importantly, we find that we tend to get an instability that is localized to particular Gaussian indices i so we introduced a method to detect this via differences in the (appropriately normalized) gradient directions on successive iterations; if we detect instability we slow down the learning rates

for the affected matrices. The discriminative training we implemented is based on Minimum Phone Error(MPE) (D.04) and Boosted MMI (DDB$^+$08) objective functions. The tuning is quite similar to a normal system, e.g. $E = 2$, four to six iterations required; with E defined by analogy with (D.04). Lattices were generated from the ML version of our factor-analyzed system, although this presented some challenges due to the slower decoding speed.

Feature space discriminative training, e.g. fMPE (D.05) is fairly straightforward as it only requires the calculation of the model likelihood gradients w.r.t. the features. We implemented a simpler version than the one described in (D.05) as we omitted the "indirect differential" and did not perform the ML model update in between each feature-space update. These two things are related as the "indirect differential" is intended to account for the effect of the model udpate. Our typical setup is to do ML training, then model space discriminative training, then feature space discriminative training, although for the Arabic system we used a more complicated sequence. Our experience is that model space discriminative training works about as well as for a normal type of system, but feature space discriminative training gives much less additional improvement. One plausible reason for this is that the structure of the transformation used in fMPE is quite similar to the model we are using (it is based on a similar sized mixture of Gaussians), so there may be less synergy between the feature and model space than for a normal system. Despite this, the improvement at the ML level tends to be large enough to outweigh the reduced benefit from discriminative training.

3.5 Probability scale

An interesting aspect of this kind of model (and it also applies to the MAP-adapted type of model), is that the optimal acoustic weight is very different from our normal system, around 1/12 as opposed to our normal value of 1/19. Because our features consist of nine frames spliced together and projected with LDA, and because LDA can be interpreted as an ML technique, we can argue that the "true" acoustic weight is (or should be bounded in some direction by) 1/9. This is because if we were to skip nine frames each time, we would be modeling each un-spliced frame exactly once. It is encouraging that this style of model takes us closer to the theoretically motivated acoustic weight.

Conditions:	Baseline	Subspace
VTLN+fMLLR+MLLR	24.3%	19.6%
+fMMI+MMI	18.2%	17.3%

Table 2: Subspace-adapt vs normal system on 50 hours English BN (test: RT'04).

3.6 Details and tuning of training

We start training by initializing the un-adapted GMM, which typically has 750 full covariance Gaussians and is trained on all speech classes mixed together. We then accumulate count and first-order statistics using the product of these 750 Gaussians' posteriors and the (zero-one) phonetic class posteriors, based on a Viterbi alignment of the data using a baseline system. These statistics are compressed to fit in memory by discarding small counts. We perform the first 10 or so iterations of training at once without re-accessing the data, using these statistics. This is sufficient to get a good initial estimate of the model. From this point we can start increasing the number of substates and doing further passes over the data while computing Gaussian posteriors based on the adapted versions of the models. In these iterations of training, we accumulate statistics that are the same size as the parameter set (representing the linear term in the objective function for each parameter), plus Gaussian-specific counts which determine the quadratic parts of the objective function. These Gaussian-specific counts dominate the statistics in terms of memory.

If we are performing speaker adaptation as part of this approach, we compute on each training iteration the speaker-specific (actually, utterance-specific) vectors as we access each training file.

3.7 Results

3.7.1 English broadcast news, 50h training

We first show results on 50 hours of training data, on an English Broadcast News task. This 50 hours is a subset of the Hub4 data which we have previously published results on (DDB$^+$08). Testing is on the RT'04 English Broadcast News test set.

System:	Dev07	Dev08	Eval07 (unsequestered)
VxU.vfr	10.0%	11.5%	14.4%
SUBxU.vfr	9.5%	11.1%	14.2%

Table 3: Arabic evaluation system, January 2009.

System:	Dev07	Dev08	Eval07 (unsequestered)
TxN.vfr	9.7%	8.6%	9.2%
SUBxN.vfr	9.7%	8.3%	9.4%

Table 4: Mandarin evaluation system, January 2009.

The results in Table 2 are quite spectacular without discriminative training (top row); nearly 20% relative improvement from this approach. However the improvement after discriminative training (bottom row) is only 5% relative. This situation where we have relatively little data plays well to the strengths of this system which has relatively few parameters (about half the baseline). We do not have exactly comparable numbers on this testing setup for the MAP-based approach described above, but results on the same training and test sets using a slightly different configuration showed an improvement of 24.6% to 23.6% with only ML training. It seems that with this amount of training data the subspace approach does much better than the MAP-based approach[1]. However, the difference between the MAP and factor anlyzed approaches was less clear with the larger Mandarin setup which we used for MAP experiments Table 1. Again we do not have exactly comparable numbers, but we estimate that the two systems in their best configuration would give about the same results under ML training.

3.7.2 Arabic January 2009 Evaluation

Table 3 shows the word error rate numbers for our Arabic system prepared for the January 2009 GALE evaluation. The VxU refers to a vowelized (V) Arabic system cross-adapted on the output of an unvowelized (U) system. "vfr" refers to "variable frame rate," an experimental technique which we applied in test time to normalize speaking rate. The subspace (SUB) system is built on top of a vowelized system, so we cross adapt to the unvowelized output giving us SUBxU. The results shown are the final numbers, with all forms of adaptation plus discriminative training. The discriminative training regime applied to the subspace system was rather complicated and involved 6 iterations of model-space boosted MMI training, lattice regeneration, 4 further passes of boosted MMI, 5 iterations of fMPE and four iterations of MPE. Normally we would expect a system to stop improving after so much discriminative training, but this system proved quite robust to it. The discriminative training regime was applied *ad hoc*; we do not believe that there is anything special about this particular setup. The improvements versus our baseline system were about 1% absolute before discriminative training, i.e. larger than our final improvements with discriminative training included.

3.7.3 Mandarin January 2009 Evaluation

We also trained this style of system for IBM's Mandarin submission to the January 2009 evaluation, trained on 1737 hours of data; see Table 4 for the results. Here, "T" means a tonal system (i.e. with tone feaures) and "N" means a non-tonal system, so the TxN baseline is a tonal system adapted (with fMLLR and MLLR) on the text output of a non-tonal system. The subspace (SUB) system is built on tonal features, and adapted on the non-tonal system. The baseline discriminative training regime was with five iterations of feature-space boosted MMI (fBMMI) followed by tree rebuilding and four iterations of model-space boosted MMI. The subspace system had five iterations of model-space BMMI and five of feature-space BMMI. The results with the subspace are not clearly better than the baseline; however, we need to consider the improvement we get from tree-based MLLR for cross-adaptation, which we implemented for the baseline but not the subspace system. E.g. using our default, non-optimized min-count of 3000 (vs 1250 in Table 4), our baseline numbers on Dev07 and Eval07(unsequestered) were 9.9% and 9.6% respectively which shows some of the effect of tree-based MLLR. So if we had implemented tree-based MLLR for the subspace system we believe we would have been able to show a clearer improvement over the

[1]Note that these numbers were obtained without MAP-adapting the variances, since the variance adaptation hurts on this smaller training set.

baseline.

4 Conclusion

We have briefly described our work with MAP-based adaptation and subspace adaptation (factor analysis) for speech recognition. In these techniques we apply robust model adaptation techniques originally devised for adapting to speakers, but instead apply them to adapting to phonetic states. In both cases we get improvements under ML training compared with a conventional system, when trained with both small and large amounts of data. The first, MAP-based technique has the drawback that it has a very large number of parameters and therefore presents problems for discriminative training. The second, factor analysis based technique is more promising. We have demonstrated word error rate improvements on a large task in an evaluation setting; however, the improvements are quite modest. In addition to the word error rate improvements, the framework is quite useful in terms of allowing new kinds of techniques to be introduced. It may also offer the opportunity to improve speaker recognition techniques as it represents a kind of "unified model" of speaker and phonetic-state variation. The main challenge is the complexity of the approach which may limit its uptake. Technical details are provided in (D.09).

Acknowledgments

This work was funded by DARPA contract HR0011-06-2-0001. The authors thank the other team members Upendra Chaudhari, Brian Kingsbury, Hong-Kwang Kuo, Lidia Mangu, Mohamed Omar and George Saon.

References

Reynolds D. A, Quatieri T. F, and Dunn R. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41, 2000.

Povey D. *Discriminative Training for Large Voculabulary Speech Recognition*. PhD thesis, Cambridge University, 2004.

Povey D. Improvements to fMPE for discriminative training of features. In *Interspeech*, 2005.

Povey D. Subspace Gaussian Mixture Models for Speech Recognition. Technical Report MSR-TR-2009-64, Microsoft Research, 2009.

Povey D., Kanevsky D., Kingsbury B., Ramabhadran B., Saon G., and Visweswariah K. Boosted MMI for Feature and Model Space Discriminative Training. In *ICASSP*, 2008.

Povey D., Chu S. M., and B. Varadarajan. Universal Background Model Based Speech Recognition. In *ICASSP*, 2008.

M.J.F. Gales. Maximum likelihood linear transformations for hmm-based speech recognition. *Computer Speech and Language*, 12:75–98, 1998.

J.L. Gauvain, L. F. Lamel, G. Adda, and M. Addadecker. Speaker-independent continuous speech dictation. *Speech Communication*, 15:21–37, 1994.

Andrew O. Hatch, Sachin Kajarekar, and Andreas Stolcke. Within-class covariance normalization for svm-based speaker recognition. In *Proc. of ICSLP*, pages 1471–1474, 2006.

Kenny P., Ouellet P., Dehak N., and Gupta V. A study of Interspeaker Variability in Speaker Verification. *IEEE Trans. on Audio, Speech and Language Processing*, 16(5):980–987, 2008.

A. Solomonoff, W.M. Campbell, and I. Boardman. Advances in channel compensation for svm speaker recognition. In *ICASSP*, volume 1, pages 629–632, 2005.