THE SYMMETRIC SUBSPACE GAUSSIAN MIXTURE MODEL
TECHNICAL REPORT MSR-TR-2010-138

Daniel Povey

Microsoft,
One Microsoft Way, Redmond, WA 98052
dpovey@microsoft.com

ABSTRACT

This document describes an extension of the Subspace Gaussian Mixture Model (SGMM). The extension is a symmetrization of the model, which makes the speaker and speech-state subspaces behave in the same way. The difference relates to the way the Gaussian weights within the substates are handled: now they depend on the speaker vector as well as the speech-state vector. This requires a little more per-speaker computation (to compute certain per-speech-state normalizing factors), but the main cost is in additional memory. The memory consumed by the model is almost doubled as we need to store in memory a new precomputed quantity. However, this method gives quite respectable WER improvements and it seems likely that it would give even greater WER improvements in situations where the number of Gaussians per speech-state is larger (i.e., with more data).

Index Terms— Speech Recognition, SGMM, Symmetric SGMM

1. INTRODUCTION

We assume at this point that the reader has already read the CSL paper [1] that describes the basic Subspace Gaussian Mixture Model (SGMM). The complete version of that model, with speaker adaptation and sub-states, can be written as follows (we omit the CMLLR adaptation since it is a feature-space transform that does not interact directly with what we are doing here):

$$p(\mathbf{x}|j,s) = \sum_{m=1}^{M_j} c_{jm} \sum_{i=1}^{I} w_{jmi} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{jmi}^{(s)}, \boldsymbol{\Sigma}_i) \quad (1)$$

$$\boldsymbol{\mu}_{jmi}^{(s)} = \mathbf{M}_i \mathbf{v}_{jm} + \mathbf{N}_i \mathbf{v}^{(s)} \quad (2)$$

$$w_{jmi} = \frac{\exp \mathbf{w}_i^T \mathbf{v}_{jm}}{\sum_{i'=1}^{I} \exp \mathbf{w}_{i'}^T \mathbf{v}_{jm}}. \quad (3)$$

The modification we introduce here is quite a simple one: to make the sub-state weights w_{jmi} a function of the speaker vector in addition to the speech-state vector, so:

$$w_{jmi}^{(s)} = \frac{\exp(\mathbf{w}_i^T \mathbf{v}_{jm} + \mathbf{u}_i^T \mathbf{v}^{(s)})}{\sum_{i'=1}^{I} \exp(\mathbf{w}_{i'}^T \mathbf{v}_{jm} + \mathbf{u}_{i'}^T \mathbf{v}^{(s)})}. \quad (4)$$

This just introduces a term that was natural but which we previously omitted because it makes things more complicated to estimate. The reasoning for making the effort to re-introduce it, is: we found in [1] that the sub-state weights were quite important in the phonetic part of the model, so they may also give a substantial improvement to the speaker vector adaptation. We can imagine models in which there are "male" and "female" indexes i, and these would be switched on and off by the speaker vectors. Of course this would all be automatically learned from data, without any explicit labels for gender or other such factors.

This technical report describes how we extend the fast likelihood computation and optimization methods described previously, to the extended model described in Equation (4).

In Section 2 we give an overview of the issues introduced by the change in the model and attempt to give the reader a sense of how we solve the resulting problems. In Section 3 we demonstrate the manipulations we use to obtain the auxiliary functions used in our update formulas. We do not provide the detailed derivation of all the auxiliary functions, but Section 3 does provide enough detail that the interested reader should be able to reconstruct those derivations. Section 4 describes the changes in the likelihood evaluation formulas that are necessary with this new model. Section 5 provides the new (and altered) accumulation and update formulae that we use with this new model.

2. OVERVIEW

2.1. Likelihood evaluation

The term in the likelihood that is changed versus the baseline model is the term $\log w_{jmi}^{(s)}$. In order to more easily discuss this, and introduce terms we will use later, we define the un-normalized weights b_{jmi} and $b_i^{(s)}$ as follows:

$$b_{jmi} = \exp \mathbf{w}_i^T \mathbf{v}_{jm} \quad (5)$$

$$b_i^{(s)} = \exp \mathbf{u}_i^T \mathbf{v}^{(s)}. \quad (6)$$

Defining a normalizing factor as follows:

$$b_{jm}^{(s)} = \sum_i b_{jmi} b_i^{(s)}, \quad (7)$$

we can write the normalized weight as:

$$w_{jmi}^{(s)} = \frac{b_{jmi} b_i^{(s)}}{b_{jm}^{(s)}}. \quad (8)$$

Since $b_i^{(s)}$ is efficient to compute, and $b_{jm}^{(s)}$ can be computed fairly efficiently as a dot-product between a vector of b_{jmi} and $b_i^{(s)}$ (with i as the vector index), we can compute likelihoods fairly efficiently. We organize these computations in such a way that we don't have to compute $\log$ or $\exp$ functions on each frame, since these are fairly expensive functions to compute.

In the resulting computations, it would be most natural to define the per-Gaussian normalizer n_{jmi} that we precompute, to contain $\log b_{jmi}$ rather than the normalized weight $\log w_{jmi}$. This would lead to inconvenience when we compute likelihoods without speaker adaptation (i.e. with $\mathbf{v}^{(s)} = \mathbf{0}$), because we have to do the extra step of computing $b_{jm}^{(s)}$ (with unit $b_i^{(s)}$), which requires either extra per-speaker computation or a small amount of extra storage. To avoid this we actually do the normalizations above slightly differently: we define w_{jmi} as the weights without speaker adaptation, i.e. as in Equation (3), and write

$$d_{jm}^{(s)} = \sum_i w_{jmi} b_i^{(s)} \tag{9}$$

$$w_{jmi}^{(s)} = \frac{w_{jmi} b_i^{(s)}}{d_{jm}^{(s)}}. \tag{10}$$

This has the additional advantage of keeping the computations required to compute $d_{jm}^{(s)}$ within a better numeric range (i.e. we are less likely to encounter numerical overflow or underflow). We have introduced two different forms of the speaker-specific weight computation because, while Equation (10) represents the way we actually do the computations, Equation (8) is conceptually cleaner and is the way we can most easily derive some of the update formulae. We can demonstrate that when using our statistics accumulated using the second version of the computation, the resulting computations produce the same result as when using the first, more natural form.

2.2. Parameter estimation

Some changes are introduced into the parameter estimation formulae by the change in the formula for the weights. There are three parts of the estimation formulae that are changed: the estimation of the speaker vectors $\mathbf{v}^{(s)}$ and speech-state vectors $\mathbf{v}_{jm}$, and the speech-state weight projections $\mathbf{w}_i$. There is also a new estimation introduced, for the speaker-space weight projections $\mathbf{u}_i$. So there are four types of parameter estimation that we need to address, corresponding to the four parameter types appearing in Equation (4). We give an overview of the issues here; in Section 5 we will give the detailed accumulation and update formulae.

2.2.1. Change in speech-state vectors and weight projections estimation

The estimation of the speech-state vectors $\mathbf{v}^{(s)}$ and weight projections $\mathbf{w}_i$ are the ones that, in terms of formulae, change the least, but at the same time they introduce the most inconvenience. The estimation formulae for for $\mathbf{v}^{(s)}$ and $\mathbf{w}_i$ described in [1] both refer to the quantity w_{jmi}. Using the "symmetric" model, we have to replace this with a quantity we call $\tilde{w}_{jmi}$ which is an appropriately weighted average of the speaker-adapted weights $w_{jmi}^{(s)}$. The update formulae are not changed except for this replacement. The reason this is inconvenient is that we need to store additional statistics a_{jmi} in order to compute $\tilde{w}_{jmi}$, and these statistics occupy a lot of memory: they are the same size as the per-Gaussian counts γ_{jmi}, which are typically larger than any of the other statistics types in the model previously described. Therefore, the size of the statistics, as well as the size of the model, is nearly doubled by this change.

2.2.2. Change in speaker vectors estimation

The estimation of the speaker vectors changes qualitatively when we symmetrize the model. Our previous estimation just solves a linear system of equations. In the symmetric model it becomes more like the the speech-state vectors estimation, where the weight-related terms introduce difficult nonlinearities and force us to make approximations. The update process is very similar to the process for updating $\mathbf{v}_{jm}$, except we use an iterative solution. In the estimation for $\mathbf{v}_{jm}$ we just used a single iteration in the update phase, since it is part of a larger iterative process; in the estimation of the speaker vectors $\mathbf{v}^{(s)}$, since we start from zero each time we see a new speaker, and typically accumulate statistics just once or twice, it is more important to iterate in the update phase.

2.2.3. Estimation of speaker-space weight projections

The estimation problem that is new for this model is that of the speaker-space weight projections $\mathbf{u}_i$. This problem is essentially the mirror-image of the problem of estimating the quantities $\mathbf{w}_i$. The difficulty is that in order to do it the same way, we need to store per-speaker statistics, i.e. the vectors $\mathbf{v}^{(s)}$ and certain count-like quantities (of size I per speaker). Previously we have avoided storing any quantities that scale with the number of speakers, because for very large corpora these could become large. The solution we have adopted is to describe two separate update methods: one which is "more exact" and is a precise mirror image of the estimation procedure for $\mathbf{w}_i$ (but which involves accumulating per-speaker statistics), and one which is "less exact" and which avoids accumulating any per-speaker statistics. The less exact method involves storing statistics sufficient to form a local quadratic approximation to the auxiliary function in each $\mathbf{w}_i$. It is equivalent to one iteration of the more exact method, except without certain convergence checks.

3. DERIVATIONS FOR OPTIMIZATION FORMULAE

In this section we present a partial derivation for some of the new optimization formulae. The intent is to introduce the new ideas used in the optimization, but not to provide a complete derivation. The main new idea described here is the way we use Jensen's inequality in the reverse sense to the way it is normally used, to move a $\log$ function out of, rather than into, a sum. The reason we can apply it in a reverse sense is that the term involved contains a negated logarithm ($-\log b_{jm}^{(s)}$).

Consider the formula for the weights, expressed in terms of b_{jmi} and other quantities as in Equation (8). The numerator of this formula does not present any problems as its log is linear in each of the quantities $\mathbf{w}_i$, $\mathbf{u}_i$, $\mathbf{v}_{jm}$ and $\mathbf{v}^{(s)}$. Any difficulties for optimization arise from the denominator. Let us write $\mathcal{Q}_1$ for the partial auxiliary function containing just this problematic term:

$$\mathcal{Q}_1 = -\sum_{j,m,s} \gamma_{jm}^{(s)} \log b_{jm}^{(s)} \tag{11}$$

$$= -\sum_{j,m,s} \gamma_{jm}^{(s)} \log \sum_i b_{jmi} b_i^{(s)}. \tag{12}$$

We are going to use the convexity of $-\log$, and Jensen's inequality, to push the $\log$ to the left past the outer summations. To do this, we need to renormalize so that, at the parameter values we accumulated with, we are taking the logarithm of 1. Let us use a bar (e.g. $\bar{b}_{jm}^{(s)}$) to represent a quantity considered as a constant, i.e. evaluated with all parameters the same as they were during accumulation. We rewrite Equation (12) as follows:

$$\mathcal{Q}_2 = -\sum_{j,m,s} \gamma_{jm}^{(s)} \log \frac{\sum_i b_{jmi} b_i^{(s)}}{\bar{b}_{jm}^{(s)}}. \tag{13}$$

This is the same as $\mathcal{Q}_1$, but with a constant offset. Our use of notation here is that by numbering them $\mathcal{Q}_1$ and $\mathcal{Q}_2$, we imply the following relationship. If we write the parameters as Λ, we will have

$$\mathcal{Q}_2(\Lambda) - \mathcal{Q}_2(\bar{\Lambda}) \leq \mathcal{Q}_1(\Lambda) - \mathcal{Q}_1(\bar{\Lambda}), \tag{14}$$

where $\bar{\Lambda}$ is the parameter values used in accumulation; i.e. the increase in $\mathcal{Q}_1$ will be at least as much as the increase in $\mathcal{Q}_2$. The same would apply for $\mathcal{Q}_3$ versus $\mathcal{Q}_2$, and so on. In order to apply Jensen's inequality we also need weighting factors that sum to one. Defining

$$\gamma = \sum_{j,m,s} \gamma_{jm}^{(s)}, \tag{15}$$

we can rewrite $\mathcal{Q}_2$ as:

$$\mathcal{Q}_2(\Lambda) = -\gamma \sum_{j,m,s} \frac{\gamma_{jm}^{(s)}}{\gamma} \log \frac{\sum_i b_{jmi} b_i^{(s)}}{\bar{b}_{jm}^{(s)}}. \tag{16}$$

We can then apply Jensen's inequality and write:

$$\mathcal{Q}_3(\Lambda) = -\gamma \log \sum_{j,m,s} \frac{\gamma_{jm}^{(s)}}{\gamma} \frac{\sum_i b_{jmi} b_i^{(s)}}{\bar{b}_{jm}^{(s)}}. \tag{17}$$

At this point, there are two directions we can go in, depending which parameter we are optimizing. In some situations it is most convenient to get rid of the $\log$ entirely. In this case, we can use the inequality $-\log(x) \geq -x + 1$ (with equality at $x = 1$), to write (cancelling the γ):

$$\mathcal{Q}_4(\Lambda) = -\sum_{j,m,s} \gamma_{jm}^{(s)} \frac{\sum_i b_{jmi} b_i^{(s)}}{\bar{b}_{jm}^{(s)}} \tag{18}$$

Since the quantities b_{jmi} and $b_i^{(s)}$ are exponential in the parameters, the next step is generally to make a quadratic approximation to the $\exp$ function (i.e. quadratic in whatever quantity we are optimizing), and solve the resulting linear system to get a proposed step. This will generally be part of an iterative process in the udpate phase. The other direction we can go from $\mathcal{Q}_3$ is to forget the $1/\gamma$ inside the $\log$ (which is just a constant offset), and write:

$$\mathcal{Q}_{4'}(\Lambda) = -\gamma \log \sum_{j,m,s} \gamma_{jm}^{(s)} \frac{\sum_i b_{jmi} b_i^{(s)}}{\bar{b}_{jm}^{(s)}}. \tag{19}$$

This generally appears as part of a larger auxiliary function that is further optimized: the difference from Equation (18) is that in the auxiliary function that we are optimizing, we retain the $\log$, rather than getting rid of it. The same quadratic approximations would still be made one each iteration.

4. CHANGES IN LIKELIHOOD EVALUATION FORMULAE

In this section we describe how the likelihood evaluation formulae change with the symmetrized model.

4.1. Global and speaker-specific pre-computation

The normalization constant n_{jmi} which we compute per Gaussian is unchanged. We repeat the formula for easy reference:

$$\begin{aligned} n_{jmi} &= \log c_{jm} + \log w_{jmi} \\ &\quad -\tfrac{1}{2}\left(\log\det\boldsymbol{\Sigma}_i + D\log(2\pi) + \boldsymbol{\mu}_{jmi}^T \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\mu}_{jmi}\right) \end{aligned} \tag{20}$$

We now also need to store in memory the quantities w_{jmi}. These will be used in a per-speaker phase of the computation to compute the normalizing factors $d_{jm}^{(s)}$.

If we are doing speaker adaptation, then for each speaker we also need to compute the speaker-specific quantities. Note that these would typically be computed on the fly as we see each speaker. As before, we have the speaker offsets:

$$\mathbf{o}_i^{(s)} = \mathbf{N}_i \mathbf{v}^{(s)}. \tag{21}$$

To handle the speaker-specific weights we also need to compute the following quantities:

$$b_i^{(s)} = \exp \mathbf{u}_i^T \mathbf{v}^{(s)} \tag{22}$$

$$d_{jm}^{(s)} = \sum_i b_i^{(s)} w_{jmi}. \tag{23}$$

It would be most convenient to first compute and store $\log b_i^{(s)}$, and then compute $b_i^{(s)}$, which would then be used to compute $d_{jm}^{(s)}$ via dot products between vectors. The quantities $d_{jm}^{(s)}$ sould be stored in the form $\log d_{jm}^{(s)}$.

The process of Gaussian selection is the same as in the previously described SGMM.

4.2. Pre-computation per frame

With the symmetric model, we change the way we compute the quantity $n_i(t)$ (for pre-selected indices i). It now contains the quantity $\log b_i^{(s)}$:

$$\begin{aligned} n_i(t) &= \log|\det \mathbf{A}^{(s)}| - \tfrac{1}{2}\mathbf{x}_i(t)^T \boldsymbol{\Sigma}_i^{-1} \mathbf{x}_i(t) \\ &\quad + \log b_i^{(s)}. \end{aligned} \tag{24}$$

Quantities in Equation (24) that we have not separately introduced are as described in [1].

4.3. Gaussian likelihood computation

We compute the contribution to the likelihood from state j, mixture m and Gaussian index i as:

$$\log p(\mathbf{x}(t), m, i|j) = n_i(t) + n_{jmi} + \mathbf{z}_i(t) \cdot \mathbf{v}_{jm} - \log d_{jm}^{(s)}. \tag{25}$$

The new term here is $-\log d_{jm}^{(s)}$, which of course we store as a log quantity (so we don't have to evaluate the log function on each frame). Also $n_i(t)$ contains a new term which was absent in the original model.

5. NEW ACCUMULATION AND UPDATE FORMULAE

In this section we describe the new and modified accumulation and update formulae. Section 5.1 describes how the speaker vectors $\mathbf{v}^{(s)}$ are computed. Section 5.2 introduces the new statistics we now need to accumulate in order to update $\mathbf{v}_{jm}$ and $\mathbf{w}_i$. Sections 5.3 and 5.4 describe the changes in the update equations for $\mathbf{v}_{jm}$ and $\mathbf{w}_i$ respectively. Sections 5.5 and 5.6 describe the more exact, and the less exact (but more scalable) versions of the update equations for the speaker-space weight projections $\mathbf{u}_i$.

5.1. Speaker vector estimation

A new element is introduced into the speaker vector estimation through the effect of the speaker vectors on the weights. There are two new terms: an easy one and a hard one. The easy one is just the linear effect of the speaker vector on the log probabilities:

$$\mathcal{Q}(\mathbf{v}^{(s)}) = \ldots + \sum_i \gamma_i^{(s)} \mathbf{u}_i^T \mathbf{v}^{(s)}. \tag{26}$$

We don't need any any additional statistics to model this, since $\gamma_i^{(s)}$ is already one of the quantities we accumulate in order to update the speaker vectors.

The other new term, the hard one, is the "normalizer" term, and this is of the following form, after some manipulations as described above to take the log outside the sum. Note that we write these equations, corresponding to the actual implementation, in terms of w_{jmi} and $d_{jm}^{(s)}$ instead of b_{jmi} and $b_{jm}^{(s)}$. This harder part of the auxiliary function is:

$$\begin{aligned}
\mathcal{Q}_1(\mathbf{v}^{(s)}) &= \ldots - \gamma^{(s)} \log \sum_{j,m} \frac{\gamma_{jm}^{(s)}}{\gamma^{(s)}} \frac{\sum_i w_{jmi} b_i^{(s)}}{\bar{d}_{jm}^{(s)}} && (27)\\
\mathcal{Q}_2(\mathbf{v}^{(s)}) &= \ldots - \gamma^{(s)} \log \sum_i a_i^{(s)} b_i^{(s)} && (28)\\
a_i^{(s)} &= \sum_{j,m} \gamma_{jm}^{(s)} \frac{w_{jmi}}{d_{jm}^{(s)}} && (29)\\
&= \sum_{t \in \mathcal{T}(s)} \sum_{j,m} \frac{\gamma_{jmi}(t) w_{jmi}}{d_{jm}^{(s)}}, && (30)
\end{aligned}$$

and note that we remove the bar from $\bar{d}_{jm}^{(s)}$ in the equation for $a_i^{(s)}$ since it is obvious in accumulation equations that we are treating the parameters as fixed. It is the second form of $a_i^{(s)}$, as written in Equation (30), that we actually use for accumulation. This is less efficient than Equation (29), but it is more convenient and this part of the accumulation does not dominate the computation time. Also note that we would actually have to compute Equation (30) before we have estimated the speaker vector for speaker s, which means that we would have $\mathbf{v}^{(s)} = 0$ and hence $d_{jm}^{(s)} = 1$. Therefore the denominator of Equation (30) may seem pointless, but it would have an effect if we did more than one iteration of E-M to update the speaker vectors. We can write the complete auxiliary function as follows:

$$\begin{aligned}
\mathcal{Q}_2(\mathbf{v}^{(s)}) &= \mathbf{y}^{(s)\,T} \mathbf{v}^{(s)} + \sum_i \gamma_i^{(s)} \mathbf{u}_i^T \mathbf{v}^{(s)} \\
&\quad -\frac{1}{2} \sum_i \gamma_i^{(s)}\ \mathbf{v}^{(s)\,T} \mathbf{N}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{N}_i \mathbf{v}^{(s)} \\
&\quad -\gamma^{(s)} \log \sum_i a_i^{(s)} b_i^{(s)} && (31)
\end{aligned}$$

The update for $\mathbf{v}^{(s)}$ is now mostly analogous to the update for $\mathbf{v}_{jm}$, except that we use the following definition:

$$\tilde{w}_i^{(s)} \equiv \frac{a_i^{(s)} b_i^{(s)}}{\sum_i a_i^{(s)} b_i^{(s)}}. \tag{32}$$

Thus, $\tilde{w}_i^{(s)}$ is the "normalized" version of the speaker weights (i.e. normalized to sum to one), but normalized with respect to the statistics $a_i^{(s)}$. This quantity will appear in the update equations in the same places that w_{jmi} appears in the update equations for $\mathbf{v}_{jm}$. Note that when $\tilde{w}_i^{(s)}$ appears in equations we will treat it as a shorthand for the right hand of (32). It should be recomputed each time from the current values of $b_i^{(s)}$.

We now describe the speaker vector update. It is an iterative process with iterations $p = 1 \ldots P$. We write the p'th iteration of the speaker vector as $\mathbf{v}^{(s,p)}$, and if we are on the first iteration of the E-M process we would be starting from $\mathbf{v}^{(s,0)} = 0$. On each iteration we form a quadratic approximation to the auxiliary function. Defining $\mathbf{v}^{(s,p)} = \mathbf{d} + \mathbf{v}^{(s,p-1)}$, we approximate (28) as a quadratic, with:

$$\mathcal{Q}_3^{(p)}(\mathbf{d}) \simeq \mathbf{g}^{(p)}\ \mathbf{d} - \frac{1}{2} \mathbf{d}^T \mathbf{F}^{(p)} \mathbf{d}, \tag{33}$$

where $\mathbf{g}^{(p)}$ and $\mathbf{F}^{(p)}$ are defined as follows. First, we write $\mathbf{H}^{(s)}$ as the quadratic term from the old equations:

$$\mathbf{H}^{(s)} = \sum_{i=1}^{I} \gamma_i^{(s)} \mathbf{N}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{N}_i \tag{34}$$

Then we define

$$\begin{aligned}
\mathbf{g}^{(p)} &= \mathbf{y}^{(s)} + \sum_{i=1}^{I} (\gamma_i^{(s)} - \gamma^{(s)} \tilde{w}_i^{(s,p-1)}) \mathbf{u}_i \\
&\quad -\mathbf{v}^{(s,p-1)} \mathbf{H}^{(s)}, && (35)
\end{aligned}$$

where the last term is needed due to the change from an "absolute" to "offset-based" representation of the auxiliary function, and $\tilde{w}_i^{(s,p-1)}$ is defined as in Equation (32) but with $b_i^{(s)}$ written instead as $b_i^{(s,p-1)} = \exp(\mathbf{u}_i \cdot \mathbf{v}^{(s,p-1)})$. We define $\mathbf{F}^{(p)}$ as:

$$\mathbf{F}^{(p)} = \mathbf{H}^{(s)} + \sum_{i=1}^{I} \gamma^{(s)} \tilde{w}_i^{(s,p-1)} \mathbf{u}_i \mathbf{u}_i^T. \tag{36}$$

The solution is then:

$$\mathbf{v}^{(s,p)} = \mathbf{v}^{(s,p-1)} + \text{solve\_vec}(\mathbf{F}^{(p)}, \mathbf{g}^{(p)}, 0, K^{\max}). \tag{37}$$

There is the potential for non-convergence here, but I consider it so remote that I don't recommend to check for it, at least for initial experiments. Note that the same issue exists for the quantities $\mathbf{v}_{jmi}$, and in that case also we do not check for convergence. We do, however measure and report the changes in Equation (31) on each iteration p as a diagnostic.

As regards the derivation of this update rule: $\mathbf{g}^{(p)}$ is the derivative of (31) with parameters $\mathbf{v}^{(s)} = \mathbf{v}^{(s,p-1)}$. The $\mathbf{F}^{(p)}$ is not exactly the negated second derivative of (31), but a slight overestimate of the negated second derivative, that differs only by a rank-one correction factor. The way we derive the second term of (36) from the last term of (31) is: first we use $-\log x \geq -\log \bar{x} + 1 - \frac{x}{\bar{x}}$ with equality at $x = \bar{x}$, and here $\bar{x}$ corresponds to $\sum_i a_i^{(s)} b_i^{(s,p-1)}$ with $b_i^{(s,p-1)} = \exp \mathbf{u}_i^T \mathbf{v}^{(s,p-1)}$. Ignoring constant factors, that term becomes $-\gamma^{(s)} \frac{\sum_i a_i^{(s)} \exp \mathbf{u}_i^T \mathbf{v}}{\sum_i a_i^{(s)} \exp \mathbf{u}_i^T \mathbf{v}^{(s,p-1)}}$ (corresponding to $-\frac{x}{\bar{x}}$, with $\gamma^{(s)}$ as a scaling factor), and taking the second derivative of this w.r.t. $\mathbf{v}$ at $\mathbf{v} = \mathbf{v}^{(s,p-1)}$, we get

$$\begin{aligned}
&-\gamma^{(s)} \sum_i \frac{a_i^{(s)} \exp \mathbf{u}_i^T \mathbf{v}^{(s,p-1)}}{\sum_i a_i^{(s)} \exp \mathbf{u}_i^T \mathbf{v}^{(s,p-1)}} \mathbf{u}_i \mathbf{u}_i^T && (38)\\
&= -\gamma^{(s)} \sum_i \tilde{w}_i^{(s,p-1)} \mathbf{u}_i \mathbf{u}_i^T. && (39)
\end{aligned}$$

5.2. Additional statistics for speech-state vectors and speech-state weight projections

We require some additional statistics in order to update the quantitites $\mathbf{v}_{jm}$ and and $\mathbf{w}_i$. These are required to compute the $\tilde{w}_{jmi}$ quantities that appear in the update equations. The statistics can be defined as a sum over speakers:

$$a_{jmi} = \sum_s \frac{\gamma_{jm}^{(s)}}{d_{jm}^{(s)}} b_i^{(s)}. \tag{40}$$

In fact, we compute them as a sum over time, as follows:

$$a_{jmi} = \sum_{t,j,m,i} \frac{\gamma_{jmi}(t)}{d_{jm}^{(s[t])}} b_i^{(s[t])}, \tag{41}$$

where $s[t]$ is the speaker active on frame t. This much less efficient than it could be but it is more convenient, and it does not dominate the computation time of the overall accumulation process.

5.3. Speech-state vector estimation

In computing the speech-state vectors $\mathbf{v}_{jm}$, we use the quantity:

$$\tilde{w}_{jmi} = \frac{w_{jmi} a_{jmi}}{\sum_i w_{jmi} a_{jmi}} \tag{42}$$

$$= \frac{b_{jmi} a_{jmi}}{\sum_i b_{jmi} a_{jmi}}. \tag{43}$$

with $b_{jmi} = \exp(\mathbf{v}_{jm}^T \mathbf{w}_i)$, and where we use the most "updated" forms of $\mathbf{v}_{jm}$ and $\mathbf{w}_i$ available to compute this, i.e. we use $\hat{w}_i$ if available (in the experiments we ran, $\mathbf{v}_{jm}$ was updated before $\mathbf{w}_i$ so the actual value of $\mathbf{w}_i$ used was the un-updated value. This quantity $\tilde{w}_{jmi}$ replaces $\hat{w}_{jmi}$ in Equations (58) and (59) of [1]. The derivation follows the general outline given in Section (3).

5.4. Speech-state weight projection estimation

In estimating the speech-state weight projections $\mathbf{w}_i$, the same change is made as above. In the auxiliary function, Equation (68) of [1], $\tilde{w}_{jmi}$ replaces w_{jmi}, and in the update equations (71) and (72), $w_{jmi}^{(p)}$ is replaced with:

$$\tilde{w}_{jmi}^{(p)} = \frac{a_{jmi} \exp \mathbf{w}_i^T \hat{\mathbf{v}}_{jm}}{a_{jmi} \exp \mathbf{w}_i^T \mathbf{v}_{jm}} \tag{44}$$

Again, the most "updated" values of $\mathbf{v}_{jm}$ and $\mathbf{w}_i$ available should be used in (44); this will generally correspond to the updated values $\hat{\mathbf{v}}_{jm}$ and whatever value of $\mathbf{w}_i$ we have on the current iteration.

5.5. Update of speaker weight projections: more exact version

As mentioned, we describe two versions of the speaker-space weight projections $\mathbf{u}_i$. We first describe the more exact version. Three types of statistics are required for this update. Two of these are per-speaker statistics, and are required in the update phase, so these would have to be stored as a list. This is a qualitiatively new aspect to the update procedure, as previously we were able to avoid any per-speaker quantities being needed in the update phase.

The first type of statistic required is $a_i^{(s)}$, as defined in Equation (30). We store these as a list, for all speakers. Also note that we would use the final, speaker-adapted alignment probabilities and speaker-dependent quantities to compute these statistics, so the value of $a_i^{(s)}$ stored in the list would not be the same as the value used to compute the speaker vectors $\mathbf{v}^{(s)}$.

The second type of statistic required is $\mathbf{v}^{(s)}$, the speaker vectors. Again, these are stored as a list.

The third type of statistic required is:

$$\mathbf{s}_i = \sum_s \gamma_i^{(s)} \mathbf{v}^{(s)}. \tag{45}$$

This requires us to store $\gamma_i^{(s)} = \sum_{t \in \mathcal{T}(s), j, m} \gamma_{jmi}(t)$, given the final, speaker-adapted alignments. This quantity is already needed for some of the other computations described in [1].

The auxiliary function in $\{\mathbf{u}_i, 1 \leq i \leq I\}$ is:

$$\mathcal{Q}_1 = \sum_i \mathbf{u}_i^T \mathbf{s}_i - \sum_s \gamma^{(s)} \log \sum_i a_i^{(s)} \exp \mathbf{u}_i^T \mathbf{v}^{(s)} \tag{46}$$

In order to separate the auxiliary function over the different values of i, and thus simplify the problem, we use the inequality $-\log(x) \geq -x + 1$ and write:

$$\mathcal{Q}_2(\mathbf{u}_i) = \mathbf{u}_i^T \mathbf{s}_i - \sum_s a_i^{(s)} \exp \mathbf{u}_i^T \mathbf{v}^{(s)}. \tag{47}$$

To obtain this, we use $\sum_i a_i^{(s)} \exp(\mathbf{u}_i^T \mathbf{v}^{(s)}) = \gamma^{(s)}$ and the $\gamma^{(s)}$ cancels. The optimization process is an iterative one where on each iteration $1 \leq p \leq P$ we compute linear and quadratic terms $\mathbf{g}_i^{(p)}$ and $\mathbf{F}_i^{(p)}$ and maximize the corresponding quadratic objective function. On each iteration we check that the auxiliary function did not decrease.

The optimization procedure for a particular value of i is as follows: Set $\mathbf{u}_i^{(0)} \leftarrow \mathbf{u}_i$ (i.e. the value at input). For $p = 1 \ldots P$ (e.g. $P = 3$), do:

$$\mathbf{g}_i^{(p)} \leftarrow \mathbf{s}_i - \sum_s a_i^{(s)} \exp(\mathbf{u}_i^{(p-1)\,T} \mathbf{v}^{(s)}) \mathbf{v}^{(s)} \tag{48}$$

$$\mathbf{F}_i^{(p)} \leftarrow \sum_s a_i^{(s)} \exp(\mathbf{u}_i^{(p-1)\,T} \mathbf{v}^{(s)}) \mathbf{v}^{(s)} \; \mathbf{v}^{(s)\,T} \tag{49}$$

Then the candidate new value of $\mathbf{u}_i^{(p)}$ is:

$$\mathbf{u}^{\mathrm{tmp}} = \mathbf{u}_i^{(p-1)} + \mathbf{F}_i^{(p)\,-1} \mathbf{g}_i^{(p)}, \tag{50}$$

or more safely

$$\mathbf{u}^{\mathrm{tmp}} = \mathbf{u}_i^{(p-1)} + \mathrm{solve\_vec}(\mathbf{F}_i^{(p)}, \mathbf{g}_i^{(p)}, \mathbf{0}, K^{\max}) \tag{51}$$

with solve_vec as defined in [1], and then we do as follows: while $\mathcal{Q}_2(\mathbf{u}^{\mathrm{tmp}}) < \mathcal{Q}_2(\mathbf{u}_i^{(p-1)})$, with $\mathcal{Q}_2$ defined as in Equation (47), set

$$\mathbf{u}^{\mathrm{tmp}} \leftarrow \frac{1}{2}(\mathbf{u}^{\mathrm{tmp}} + \mathbf{u}^{(p-1)}). \tag{52}$$

Then (once the auxiliary function is no longer worse than before), set $\mathbf{u}^{(p)} \leftarrow \mathbf{u}^{\mathrm{tmp}}$.

At the end we set $\hat{\mathbf{u}}_i \leftarrow \mathbf{u}_i^{(P)}$.

5.6. Update of speaker weight projections: less exact version

For the less exact version of the computation of the speaker weight projections, we avoid storing any lists of speaker-specific quantities and instead accumulate statistics sufficient to form a local quadratic

approximation of the auxiliary function, which we directly maximize in the update phase. In this case we store the following statistics:

$$\mathbf{t}_i = \sum_s \left(\gamma_s^{(i)} - a_i^{(s)} b_i^{(s)} \right) \mathbf{v}^{(s)} \quad (53)$$

$$\mathbf{U}_i = \sum_s a_i^{(s)} b_i^{(s)} \mathbf{v}^{(s)} \; \mathbf{v}^{(s)\,T}. \quad (54)$$

The auxiliary function we maximize is as follows, where Δ_i is the change in $\mathbf{u}_i$:

$$\mathcal{Q}_3(\Delta_i) = \mathbf{t}_i^T \Delta_i - \frac{1}{2} \Delta_i^T \mathbf{U}_i \Delta_i, \quad (55)$$

and our update equation is $\hat{\mathbf{u}}_i \leftarrow \mathbf{u}_i + \Delta_i$, or more generally, to handle the singular cases,

$$\hat{\mathbf{u}}_i \leftarrow \mathbf{u}_i + \mathrm{solve\_vec}(\mathbf{U}_i, \mathbf{t}_i, \mathbf{0}, K^{\max}), \quad (56)$$

with the function $\mathrm{solve\_vec}$ as defined in [1].

6. REFERENCES

[1] D. Povey, Lukáš Burget, et al., "The Subspace Gaussian Mixture Model – a Structured Model for Speech Recognition," *Computer Speech and Language (accepted)*, 2010.