Robust Estimation and Adaptation of Subspace Gaussian
Mixture Models for Automatic Speech Recognition

Liang Lu

October 21, 2011
Abstract

In conventional hidden Markov model (HMM) based speech recognisers, the emitting HMM states are modelled by Gaussian Mixture Models (GMMs), with parameters been estimated directly from the training data. However, in Subspace Gaussian mixture model (GMM) based acoustic modelling, the parameters of each state model are derived from the globally shared model subspaces which are normally low dimensional. This leads to significant reduction in terms of total parameters to be estimated while allowing larger number of Gaussians to be used to increase the model capacity. Considerable performance gains are observed in several speech recognition tasks compared to conventional acoustic modelling. Though relatively compact, SGMM acoustic models still suffer from model overfit problem given small amount of training data. In this report, we will discuss the model estimation with regularizations to improve robustness, and model estimation from out-domain data to increase the model accuracy. We will also show that these techniques can be successfully applied to develop a target language system with low resource by cross-lingual setting. In addition, to address the mismatch between model training and testing, we will also present the adaptation techniques of SGMM acoustic model. In particular, we will show how the model estimated from out-domain data can be adapted by maximum a posteriori (MAP) criteria to fit the target system better, and also the model trained in clean condition can be adapted with noise compensation techniques to work well in noisy environment.
## Contents

1 Introduction 3

2 Overview of SGMM Acoustic Model 4
   2.1 Likelihood and Posterior  ........................................ 4
   2.2 Sufficient Statistics ............................................. 6
   2.3 Maximum Likelihood Model Estimation .......................... 6
   2.4 Summary ............................................................ 8

3 Model Estimation with Regularization 8
   3.1 Regularized state vector estimation ............................ 9
   3.2 Optimization ....................................................... 10
   3.3 Modified Regularization .......................................... 11
   3.4 Summary ............................................................ 12

4 Model Estimation from Out-domain Data 12
   4.1 Parameters Sharing Across Systems ............................ 13
   4.2 Application to Cross-lingual Systems ........................... 13
   4.3 Summary ............................................................ 14

5 MAP Adaptation of Model Subspace 14
   5.1 Auxiliary Function of MAP Criteria ............................. 15
   5.2 Matrix Variate Gaussian Prior ................................... 15
   5.3 Prior Distribution Estimation ................................. 16
   5.4 MAP Estimation of the Phonetic Subspace .................... 17
   5.5 Summary ............................................................ 18

6 Noise Adaptation of SGMM Acoustic Model 18
   6.1 Mismatch Function ................................................. 19
   6.2 Vector Taylor Series ............................................... 20
   6.3 Joint Uncertainty Decoding ..................................... 20
   6.4 VTS and JUD for SGMM Acoustic Model ........................ 21
      6.4.1 Noise model estimation .................................... 22
   6.5 Summary ............................................................ 23

7 Experiments 23
   7.1 Experiments of Regularized Model Estimation ............... 23
      7.1.1 Baseline System ............................................. 24
      7.1.2 SGMM results with smoothing and renormalization ...... 24
      7.1.3 SGMM results with regularization .......................... 24
   7.2 Experiments of Cross-lingual Systems ......................... 26
      7.2.1 GlobalPhone Corpus ......................................... 26
      7.2.2 Baseline Monolingual Systems .............................. 27
      7.2.3 Cross-lingual experiments .................................. 28
7.2.4 Sparsity Analysis .................................................. 33
7.3 Experiments of MAP Adaptation ................................... 34
  7.3.1 Baseline results ................................................. 34
  7.3.2 MAP adaptation results ....................................... 34
7.4 Experiments of Noise Adaptation ................................. 37
  7.4.1 Results of noise compensation on GMM based systems .... 37
  7.4.2 Results of noise compensation on SGMM based systems .... 38

8 Plan for Future Work ................................................. 40

9 Summary ..................................................................... 40
1 Introduction

Thanks to decades of research, automatic speech recognition (ASR) has achieved great progress in recent years, and successful applications have been deployed in internet or mobile network. Currently, the prevalent acoustic modelling approach relies on the hidden Markov models (HMMs) with Gaussian Mixture Models (GMMs) to model the emitting states [1]. For large vocabulary tasks, context dependent HMM topology is employed, in which, rather than modelling the individual phoneme unit of speech, the emitting states model the triphones or even quinphones according to the context. Context dependent models can capture the finer characters in speech, especially for utterances of fast speech rates. On the other hand, however, it also leads to remarkable increase in model size which makes it difficult to train the model. State tying is proposed to reduce the model size by clustering the “similar” context dependent states into classes, typically with a regression tree, and the model parameters are shared within each class [2]. For a typical large vocabulary speech recognition system, the number of HMM emitting states can be thousands or tens of thousands after state tying, which is normally a small proportion of the total context dependent states.

In HMM/GMM framework, each HMM emitting state is modelled by a Gaussian Mixture Model (GMM). In a typical large vocabulary speech recogniser, the number of Gaussian components is normally tens to hundreds according to the amount of data that available. In conventional acoustic modelling, the GMM parameters, i.e. mixture weights, means and covariances, for each state are estimated directly, which leads to at least millions of active parameters in total in a system. Though according to previous experience, such kind of modelling works well in many applications, the recently proposed subspace GMM (SGMM) based acoustic model shows that, it’s possible to reduce the total number of active parameters by relying on the correlations between HMM state models [3]. In SGMM based acoustic models, the low dimensional model subspaces are introduced which capture the phone and speaker correlations. The model subspaces are globally shared which means they do not depend on the HMM states, and the parameters of state GMMs are derived from the model subspaces with relatively small number of state dependent parameters. This considerably reduces the total number of parameters to be estimated, and meanwhile, allows larger number of Gaussian mixtures to be used for each HMM states to model the acoustic variations, and hence leads to significant performance gains according to experiments in several speech recognition tasks.

In this report, we present our own work in SGMM based acoustic modelling. After the seminal paper of Povey et. al [3], SGMM has risen to be an active research topic in field of ASR, which covers many aspects. Our contribution in this topic up to now includes

- Model estimation with regularization penalties to improve robustness
- Model estimation from out-domain data to improve accuracy and its application to cross-lingual speech recognition
• Maximum a posteriori (MAP) adaptation of model subspace and its application to cross-lingual speech recognition
• Applying model based noise compensation techniques for noise robust SGMM acoustic model

In the rest of the report, we will present detailed discussion of these work with experimental studies, and then point out the future work for our 3rd year Ph.D research.

2 Overview of SGMM Acoustic Model

SGMM acoustic models are similar to conventional Gaussian mixture model (GMM) systems, in that the output pdf of each HMM state is a GMM. The principal difference is that the Gaussian means and the mixture component weights are derived from the phonetic and speaker subspaces, in order to capture the corresponding variability together with the weight projection [3]. In addition, the covariance matrices (which are normally full rather than diagonal) are shared between all the HMM states. The model may be expressed formally as:

\[
p(x_t|j, s) = \sum_{m=1}^{M_j} c_{jm} \sum_{i=1}^{I} w_{jmi} N(x_t|\mu_{jmi}, \Sigma_i) \quad (1)
\]

\[
\mu_{jmi} = M_i v_{jm} + N_i v^{(s)} \quad (2)
\]

\[
w_{jmi} = \frac{\exp w^T_{jmi} v_{jm}}{\sum_{i'=1}^{I} \exp w^T_{jmi'} v_{jm}} \quad (3)
\]

where \( x_t \in \mathbb{R}^F \) denotes the \( t \)-th \( F \)-dimensional acoustic frame, \( j \) is the HMM state index, \( m \) is a sub-state [3], \( I \) is the number of Gaussians, and \( \Sigma_i \) is the \( i \)-th covariance matrix. \( v_{jm} \in \mathbb{R}^S \) is referred to as the sub-state vector, and \( S \) denotes the subspace dimension. The matrices \( M_i \) and the vectors \( w_i \) span the model subspaces for Gaussian means and weights respectively, and are used to derive the GMM parameters given sub-state vectors (equations (2) and (3)). Similarly, \( N_i \) defines the speaker subspace for Gaussian means, and \( v^{(s)} \in \mathbb{R}^T \) is referred as the speaker vector where \( T \) denotes the dimension of the speaker subspace.

2.1 Likelihood and Posterior

Since the SGMM based acoustic model is in principle the same with the conventional GMM based one, algorithms of computing the likelihood and posterior for model training and decoding is also similar to conventional ones. However, we can utilise the structure of SGMM acoustic model to speed up the computation by caching some global terms. This is important in practice as the total number of Gaussian in SGMM acoustic model
is much larger which makes both model training and decoding impractical without these strategies.

Firstly, the likelihood for state $j$, mixture $m$ and Gaussian $i$ can be factorized into three terms, i.e.

$$
\log p(x_t, m, i|j) = n_i(t) + n_{jmi} + z_i(t)^T v_{jm},
$$

(4)

where $n_{jmi}$ denotes the global term which does not depend on the acoustic frame, hence only need to be computed once as

$$
n_{jmi} = \log c_{jm} + \log w_{jmi} - 0.5 \left( \log |\Sigma_i| + D \log(2\pi) + \mu_{jmi}^T \Sigma_i^{-1} \mu_{jmi} \right)
$$

(5)

Here we use $\mu_{jmi} = M_i v_{jm}$ without speaker offset in equation (2). $n_i(t)$ and $z_i(t)^T v_{jm}$ are frame dependent terms which are defined as

$$
z_i(t) = M_i^T \Sigma_i^{-1} x_i(t)
$$

(6)

$$
n_i(t) = \log |A^{(s)}| - 0.5 x_i(t)^T \Sigma_i^{-1} x_i(t)
$$

(7)

where $x_i(t)$ is the speaker-adapted feature frame

$$
x_i(t) = A^{(s)} x(t) + b^{(s)} - N_i v^{(s)}.
$$

(8)

$A^{(s)}$ and $b^{(s)}$ is speaker dependent constraint maximum likelihood linear regression (CMLLR) transforms, and $N_i v^{(s)}$ is speaker mean offset defined in equation (2). The total likelihood for state $j$ of equation (1) is

$$
\log p(x_t|j) = \log \sum_{m,i} p(x_t, m, i|j)
$$

(9)

In practice, the summarisation is done using a “log add” function that computes $f(a, b) = \log(\exp a + \exp b)$ without ever computing $\exp a$ or $\exp b$ directly in case of underflow.

Except the pre-computation, we can make the system more efficient by pruning the Gaussians. For this, a universal background model (UBM) is introduced, and for each frame $x_t$, only a small number of Gaussians $P$ (e.g. $P = 15$) are switched on according to the likelihood on the UBM. Hence, rather than compute the likelihood of (4) for all the Gaussian, the likelihood will only be computed for the active Gaussians. This will significantly speedup the model training and decoding. In addition, if the UBM is fixed during model estimation, the Gaussian pruning process is only required to be done once.

Given the per-Gaussian and per-state likelihood of equation (4) and (9), the state posterior $p(j|t) = \gamma_j(t)$ can be obtained by standard forward-backward or Viterbi algorithm. The Gaussian posterior can be derived from the Bayesian rule as

$$
\gamma_{jmi}(t) = p(j, m, i|t)
$$

$$
= \gamma_j(t) \frac{p(x_t, m, i|j)}{p(x_t|j)}
$$

(10)

(11)
A further pruning on the Gaussian posteriors can be applied as the values for most of Gaussians are relatively very small. Given the likelihood and posterior, we will define the sufficient statistics first before presenting the formulations of model estimation in maximum likelihood fashion.

### 2.2 Sufficient Statistics

The sufficient statistics to update the GMM parameters are the zeroth, first and second order of statistics. As a special type of GMM, except for these statistics, we also need some variants of these statistics to update the SGMM acoustic model. The zeroth order statistics are Gaussian and sub-state posteriors as

\[
\gamma_{jmi} = \sum_t \gamma_{jmi}(t), \quad \gamma_{jm} = \sum_i \gamma_{jmi}
\]

(12)

The first order of statistics to update sub-state vector \(v_{jm}\) and the model subspace \(M_i\) are

\[
y_{jm} = \sum_{t,i} \gamma_{jmi}(t)z_i(t)
\]

(13)

\[
Y_i = \sum_{t,j,m} \gamma_{jmi}(t)x_i(t)v_{jm}^T
\]

(14)

where \(z_i(t)\) and \(x_i(t)\) are defined as equation (6) and (8), respectively. The second order statistics are defined as

\[
S_i = \sum_{t,j,m} \gamma_{jmi}(t)x_i(t)x_i(t)^T
\]

(15)

which is used to update the global covariance matrix \(\Sigma_i\). In addition, we further define the following statistics which are used to update the \(v_{jm}\) and \(M_i\)

\[
H_i = M_i^T \Sigma_i^{-1} M_i
\]

(16)

\[
Q_i = \sum_{j,m} \gamma_{jmi}v_{jm}v_{jm}^T
\]

(17)

The statistics required to update the speaker subspace parameters \(N_i\) and \(v^{(s)}\) are analogous to those for \(M_i\) and \(v_{jm}\), which are included in this report. Detail for this can be found in [3], which also include the estimation of speaker dependent CMLLR transforms \(A^{(s)}\) and \(b^{(s)}\).

### 2.3 Maximum Likelihood Model Estimation

Compared to conventional GMM acoustic model, SGMM based approach organises the total parameters in structured fashion. This leads to benefits in terms of reduction of
total number of parameters, capturing model correlations and factorisation of different
attribute, et. al. However, this also make it complex to estimate the model parameters
compared to conventional modelling. Nevertheless, in this section, we will show that
neat update formulation can be derived from expectation-maximisation (EM) algorithm
to estimate the model in maximum likelihood (ML) criteria.

Given the observed data samples $X$ with latent labels $Z$ (which we consider discrete case
here), the likelihood of current model parameters $\theta$ is expressed as

$$\log p(X|\theta) = \log \sum_Z p(X, Z|\theta). \tag{18}$$

Since the latent variables $Z$ are not observed, it difficult to get the exact expression of
the right side of equation (18). However, we can obtain the lower bound of the complete
likelihood $\log p(X|\theta)$ as

$$\log p(X|\theta) = \sum_Z p(Z|X, \theta^{old}) \left\{ \log p(X, Z|\theta) - \log p(Z|X, \theta) \right\} \tag{19}$$

$$\geq \sum_Z p(Z|X, \theta^{old}) \log p(X, Z|\theta). \tag{20}$$

Equation (19) is derived from the product rule of probability

$$\log p(X, Z|\theta) = \log p(Z|X, \theta) + \log p(X|\theta), \tag{21}$$

and the inequality (20) is obtained by the fact that

$$-\sum_Z p(Z|X, \theta^{old}) \log p(Z|X, \theta) \geq 0 \tag{22}$$

In SGMM based acoustic models, the latent variables are state index $j$, sub-state index
$m$ and Gaussian index $i$. In the previous subsections, we have shown how to estimate
these posteriors, and the formulate of complete likelihood. Hence, it’s straightforward
to derive the EM auxiliary functions to update the SGMM model parameters.

To be specific, the auxiliary function to update the sub-state vector $v_{jm}$ is

$$Q(v_{jm}) = \sum_{i,t} p(i, m, j|x_t) \log p(x_t, i, m, j|v_{jm}) \tag{23}$$

$$= -0.5v_{jm}^T H_{jm} v_{jm} + v_{jm}^T g_{jm} + \text{const}, \tag{24}$$

where

$$g_{jm} = y_{jm} + \sum_i \tilde{w}_i (\gamma_{jmi} - \gamma_{jm} \tilde{w}_{jmi} - \max(\gamma_{jmi}, \gamma_{jm} \tilde{w}_{jmi}) \tilde{w}_i^T v_{jm}) \tag{25}$$

$$H_{jm} = \sum_i \gamma_{jmi} H_i + \max(\gamma_{jmi}, \gamma_{jm} \tilde{w}_{jmi}) \tilde{w}_i \tilde{w}_i^T \tag{26}$$
where $y_{jm}$ and $H_i$ are defined in equation (13) and (16), $\hat{w}_{jmi}$ denotes the weight derived from equation (3) with updated weight projection $w_i$ and “old” sub-state vector $v_{jm}$. Similarly, the auxiliary function to update the phonetic subspace $M_i$ is

$$Q(M_i) = tr(M_i^T \sum_1^{-1} Y_i) - 0.5tr(\sum_1^{-1} M_i Q_i M_i^T) + const$$

(27)

where $Y_i$ and $Q_i$ are defined in equation (14) and (17). The auxiliary function for sub-state weight and weight projection $w_i$ are

$$Q(c_{jm}) = \sum_{jm} \gamma_{jm} \log c_{jm} + const$$

(28)

$$Q(w_i) = \sum_{j,m,i} \gamma_{jmi} \log w_{jmi} + const$$

(29)

Finally, the auxiliary function for the globally shared covariance matrix $\sum_i$ is

$$Q(\sum_i) = -0.5 \sum_{j,m,t} \gamma_{j,m,i} (t) \left( \log |\sum_i| + (x_i(t) - \mu_{jmi})^T \sum_1^{-1} (x_i(t) - \mu_{jmi}) \right) + const$$

(30)

where $x_i(t)$ is defined in equation (8) and $\mu_{jmi} = M_i v_{jm}$. The auxiliary functions for speaker subspace $N_i$ and speaker vector $v^s$ are analogous to that of $M_i$ and $v_{jm}$ which are not presented here. Refer [3] for this part and the solutions of maximising the auxiliary functions.

2.4 Summary

In this section, we present the overview of the SGMM acoustic model. In SGMMs, the total model parameters are factorized into different categories, e.g. the globally shared model subspace and state dependent parameters which are combined to derive the HMM state GMM model. Such kind of factorization reduces the total parameters to be estimated, and allows much larger number of Gaussians to be used for each states. This increase the model capacity, but also brings up the computation issue. We show how to speed up the system by pre-computation and Gaussian pruning. In addition, we also show how to estimate the model using EM algorithm based on maximum likelihood criteria.

3 Model Estimation with Regularization

In Section 2, we present an overview of SGMM acoustic model, and formulate the model estimation by EM with maximum likelihood (ML) criteria. Though in SGMM based acoustic model, the total parameters is much smaller than that in convention system, ML training may still suffer from overfitting with insufficient training data. This is especially true for the state-dependent parameters, as the amount of acoustic data attributed to
each state tends to be small. To be specific, the auxiliary function for a particular (sub-)state vector $v$ is shown in equation (24) and it’s rewritten here without index while do not cause confusions:

$$Q(v) = -\frac{1}{2}v^T Hv + g^T v + \text{const},$$  \hspace{1cm} (31)

As stated previously, $g$ is a $S$-dimensional vector and $H$ is a $S \times S$ matrix, representing the first- and second-order statistics respectively. Although the state vectors are normally low-dimensional, the amount of data for computing the statistics $H$ and $b$ may still be insufficient. Some heuristic approaches may be applied, for instance $H$ and $b$ may be smoothed by the global statistics:

$$\hat{H} = H + \tau H^{sm}, \quad \hat{b} = b + \tau b^{sm},$$ \hspace{1cm} (32)

where $H^{sm}$ and $b^{sm}$ denotes the smoothing term calculated based on all the HMM states (see [4] for details), and $\tau \in \mathbb{R}$ is the tuning parameter. Povey et al. [3] also discuss some numeric controls to tackle the poor condition of $H$. In this section, we will discuss the regularized model estimation for robustness given insufficient training data [5]. In particular, we focus on the regularized estimation of (sub-)state vectors, however, this approach can be easily extended to other parameter types.

### 3.1 Regularized state vector estimation

To regularize the estimation of the (sub-)state vectors, we introduce an element-wise penalty term to the original ML auxiliary function in order to smooth the output variables, giving:

$$v = \arg\max_v Q(v) - J_\lambda(v).$$ \hspace{1cm} (33)

$J_\lambda(v)$ denotes the regularization function for $v$ parametrised by $\lambda$. We may interpret $-J_\lambda(v)$ as a log-prior for the state vector, in which case we can interpret (33) as a MAP estimate. However, here we treat the problem more in terms of the design and analysis of regularization functions, rather than giving an explicit Bayesian treatment as used in JFA-based speaker recognition where Gaussian priors are applied to both speaker and channel factors [6].

We may formulate a family of regularization penalties in terms of a penalty parameter $\lambda$, and an exponent $q \in \mathbb{R}$:

$$J_\lambda(v) = \sum_i |v_i|^q \text{ s.t. } \lambda \geq 0.$$ \hspace{1cm} (34)

The case $q = 1$ corresponds to $\ell_1$-norm regularization, sometimes referred to as the lasso [7], and the case $q = 2$ corresponds to $\ell_2$-norm regularization, which is referred to as ridge regression [8] or weight decay.

Both $\ell_1$- and $\ell_2$-norm penalties perform an element-wise shrinkage of $v$ towards zero in the absence of an opposing data-driven force [8], which enables more robust estimation.
The $\ell_1$-norm penalty has the effect of driving some elements to be zero, thus leading to a kind of variable selection, and inspiring its application in sparse representation of speech features [9, 10]. It is possible to seek a compromise between the $\ell_1$ and $\ell_2$ penalties by simply setting $1 < q < 2$ which is sometimes referred to as a bridge penalty. However, the nonlinearity of the bridge penalty brings increased computational complexity. Alternatively, the $\ell_1$- and $\ell_2$-norm penalties can both be applied, as in elastic net regularization [11]:

$$J_\lambda(v) = \lambda_1 \sum_i |v_i| + \lambda_2 \sum_i |v_i|^2,$$

$$s.t. \lambda_1, \lambda_2 \geq 0.$$  \hspace{1cm} (35)

This is much less computationally demanding than the bridge penalty. In this section, we investigate the $\ell_1$-norm, $\ell_2$-norm and elastic net regularization for the estimation of SGMM (sub-)state vectors.

### 3.2 Optimization

Given the regularized objective function for state vector estimation (33), a closed form solution is readily available for the $\ell_2$-norm penalty:

$$\hat{v} = \arg \max_v -\frac{1}{2} v^T Hv + b^T v - \lambda \|v\|_{\ell_2}$$

$$= (H + \lambda I)^{-1} b$$

However, there is no such closed form solutions for the $\ell_1$-norm and elastic net penalties as the derivatives of their objective functions are not continuous. In both the optimization and signal processing fields, there have been numerous approaches proposed to solve the $\ell_1$-norm penalty problem and here we adopt the gradient projection algorithm of Figueiredo et al. [12]. The same approach may be applied to the elastic net penalty as it can be formulated in terms of the $\ell_1$ penalty:

$$\hat{v} = \arg \max_v -\frac{1}{2} v^T (H + \lambda_2 I)v + b^T v - \lambda_1 \|v\|_{\ell_1},$$

given the regularization parameters $\lambda_1$ and $\lambda_2$. A proper scaling factor should applied to the result of (36) to get the exact elastic net solution, but we did not do it in this work which corresponds to the naive elastic net [11].

Expressing (33) with the $\ell_1$ penalty results in the following objective function:

$$\hat{v} = \arg \min_v \frac{1}{2} v^T Hv - b^T v + \lambda \|v\|_{\ell_1}, \lambda > 0.$$  \hspace{1cm} (37)

As the derivative of the objective function is not continuous, which makes the search of global optimum difficult, we introduce two auxiliary vectors $x$ and $y$ such that:

$$v = x - y, \quad x \geq 0, y \geq 0,$$

$$s.t. \lambda_1, \lambda_2 \geq 0.$$
where, \( x = [v]_+ \) which takes the positive entries of \( v \) while keeping the rest as 0, i.e. \( x_i = \max\{0, v_i\} \) for all \( i = 1, \ldots, S \). Similarly, \( y = [-v]_+ \). In this case, equation (37) can be rewritten as

\[
(\hat{x}, \hat{y}) = \arg\min_{x, y} \frac{1}{2}(x - y)^T H (x - y) \\
- b^T (x - y) + \lambda_1 S x + \lambda_2 S y \\
s.t. \quad x \geq 0, y \geq 0
\]  

(39)

where \( 1_S \) denotes an \( S \)-dimensional vector whose elements are all 1. We can reformulate (39) further as a more standard bound-constraint quadratic program

\[
\hat{z} = \arg\min_z \frac{1}{2} z^T B z + c^T z \quad s.t. \quad z \geq 0
\]  

(40)

where we have set

\[
z = \begin{bmatrix} x \\ y \end{bmatrix}, \quad c = \lambda_1 2S + \begin{bmatrix} -b \\ b \end{bmatrix}, \quad \text{and } B = \begin{bmatrix} H & -H \\ -H & H \end{bmatrix}.
\]

The objective function (40) does not suffer the nonlinearity problem of the original objective function (37), and its gradient is readily available as

\[
G(z) = Bz + c,
\]  

(41)

which forms the basis of the gradient projection algorithm (see [12] for details).

The regularization parameters in (34) or (35) should vary according to the size of training data and the model complexity, however, in order to simplify the model training procedure, we adopt global and constant regularization parameters in this work. It also have to note that as the state vector \( v_j \) and subspace parameters \( M, w \) are interdependent. In principle, the shrinkage of state vectors by regularization may be undone by a corresponding scaling of \( M \) or \( w \). This can be addressed by the renormalizing the phonetic subspaces, as described in [4] [Appendix K], such that the state vectors \( v_j \) always have unit variance after each iteration.

### 3.3 Modified Regularization

Regularizing all coefficients in the sub-state vectors (33), forces the sub-state vector to shrink towards zero, corresponding to a prior on the Gaussian means centered at the origin. Intuitively, we would prefer a prior that fits the general distribution of the data. To achieve this, we modify the regularization such that the Gaussian means shrink towards a universal background model (UBM). This is done by setting the first coefficient of sub-state vector \( v \) to be 1, which also forces the first column of phonetic subspace matrices \( M_i \) to be the UBM means during model update. The regularization penalty is
applied to the remaining sub-state coefficients. For \( \ell_1 \)-norm regularization, the objective function becomes:

\[
\hat{v} = \arg \max_v \frac{1}{2} v^T H v + v^T y - \lambda \|v\|_{\ell_1},
\]

s.t. \( \lambda \geq 0, \ v[1] = 1. \)  

(42)

where we are fixing the first coefficient of \( v \) to be exactly one. If we adopt the following expressions:

\[
\hat{v} = \begin{bmatrix} 1 \\ \hat{v}^* \end{bmatrix}, v = \begin{bmatrix} 1 \\ v^* \end{bmatrix}, y = \begin{bmatrix} a \\ y^* \end{bmatrix}, H = \begin{bmatrix} b & h^T \\ h & H' \end{bmatrix},
\]

then equation (42) is equivalent to

\[
\hat{v}^* = \arg \max_{v^*} \frac{1}{2} v^{*T} H^* v^* + v^{*T} (y^* - h) - \lambda \|v^*\|_{\ell_1},
\]

s.t. \( \lambda \geq 0. \)  

(43)

This modified regularization, does not involve any change in accumulating the statistics of \( H \) and \( y \), and the code for the original regularization [5] can be reused.

3.4 Summary

In this section, we discuss the model estimation with regularizations to overcome the overfit problem in case of limited training data. We present the \( \ell_1 \)- and \( \ell_2 \)-norm regularization penalties for (sub-)state vectors, and show the optimization algorithm to obtain the solution. Similar regularization algorithm can be applied to other type of parameters, e.g. the globally shared parameters \( M_i, w_i \) and \( \Sigma_i \). However, in the following section, we will show that it may be better to estimate these globally shared parameters from out-domain data when target training data is limited. We will also show in the experimental section that, the regularized (sub-)state vector estimation and model estimation with out-domain training data can work well together when applied to cross-lingual speech recognition systems with low develop resource.

4 Model Estimation from Out-domain Data

In condition of sparse training data, we have shown that regularization can help to improve model robustness and avoid model overfit. Another way to handle the data insufficiency is to leverage on the availability of out-domain or out-language resources. There has been consistent work along this line in the filed of multilayer perception (MLP) based speech recognition systems [13, 14, 15, 16], in which, the out-domain training data are used to build the MLP models, and the posterior features derived for the target training
data with the pre-trained MLPs. This approach can be viewed as feature level knowledge sharing among different domains or languages. In this section, we will show that similar methodology can be applied to SGMM based acoustic models. However, in contrast to sharing knowledge in the feature level, we will show that the structure of SGMMs allows us to exploit the out-domain data in model level by sharing parameters across multiple SGMM systems.

4.1 Parameters Sharing Across Systems

For SGMM based acoustic models, thought it’s possible to share the training data on the state level [17], in this section, we only focus on jointly estimating the globally shared parameters, i.e. $M_i, w_i$ and $\Sigma_i$, across multiple systems [18, 19], as we consider that the state dependent parameters can be estimated with regularizations to improve robustness and accuracy. Here, we assume that the resource are available for $N$ systems, including lexicons, transcriptions and acoustic data. We denote the indexes are $\{S_1, \ldots, S_N\}$ for these systems. Tying the parameter set $M_i, w_i$ and $\Sigma_i$ across these systems can be represented as

$$\{M_i, w_i, \Sigma_i\} = \arg \max_{M_i, w_i, \Sigma_i} \sum_{n=1}^{N} p(X(S_n)|M_i, w_i, \Sigma_i)$$

(44)

where $X(S_n)$ denotes the acoustic frames for system $S_n$. Similar to the model estimation in section 2, the parameters $M_i, w_i, \Sigma_i$ can be updated iteratively based EM algorithm. The auxiliary functions are analogous to the one in section 2, for example, the auxiliary function for $M_i$ now becomes

$$\hat{Q}(M_i) = \sum_{n=1}^{N} \left( tr(M_i^T \Sigma_i^{-1} Y_{i,n}) - 0.5 tr(\Sigma_i^{-1} M_i Q_{i,n} M_i^T) \right) + \text{const}$$

(45)

which is derived from the original auxiliary function (27), where $Y_{i,n}$ and $Q_{i,n}$ are the corresponding statistics for system $S_n$. Similar form of auxiliary functions can be obtained for $w_i$ and $\Sigma_i$. As seen from equation (45), tying parameters across systems lead to that the parameters are updated by all the statistics from multiple systems, hence lead to better model estimation for system with insufficient training data. This technique is particularly useful to development a target language system with very limited resource which will be discussed later.

4.2 Application to Cross-lingual Systems

Large vocabulary continuous speech recognition systems rely on the availability of substantial resources including transcribed speech for acoustic model estimation, text for language model estimation, and a pronunciation dictionary. Building a speech recognition system from scratch for a new language thus requires considerable investment in
such resources. Cross-lingual acoustic modelling has the aim of significantly reducing the amount of acoustic training data for a new target language, by leveraging on existing acoustic models for other source languages. However, owing to differences such as different sets of subword units, this is not a straightforward task. There have been three main approaches to cross-lingual acoustic modelling: the use of global phone sets \[20, 21, 22, 23\], cross-lingual phone/acoustic mapping \[24, 25, 26, 27\], and cross-lingual tandem features \[13, 14, 15, 16\].

However, recent works have show that the cross-lingual acoustic modelling based on SGMMs can achieve significant performance gains. We have shown that the globally shared parameters $M_i$, $w_i$ and $\Sigma_i$ can be estimated jointly across multiple systems. In cross-lingual settings, they are estimated from source language system(s) \[18, 19\]. The parameters estimated from out-language data are then borrowed for the target language system. Only state dependent parameters are retrained while the out-domain parameters can be fixed during model training, or be adapted by the target language training data (cf. section 5). We have shown in \[19\] that the out-domain globally shared parameters can be successfully applied to target language system, and the parameters estimated in multiple source language systems as in subsection 4.1 considerably outperform that estimated from monolingual systems. In addition, we also show that, for state-dependent parameters, regularized estimation as in section 3 can improve the numerical accuracy and allow the use of larger number of basis in $M_i$ and $w_i$, hence lead to performance gains.

### 4.3 Summary

In this section, we have discussed the parameter estimation of SGMM acoustic model from out-domain data to improve model estimation accuracy in case of limited training data. In particular, we have show that the globally shared parameters $M_i$, $w_i$ and $\Sigma_i$ in SGMMs can be tied across multiple systems. In addition, We have also shown that the parameter sharing approach can be successfully applied to cross-lingual speech recognition systems, where the resource for target language is very low. However, a problem in this setting is that it may give rise to a potential mismatch between the parameters estimated from out-domain data and target language system. In the following section, we will discuss that, this potential mismatch can be alleviated by adaptation techniques.

### 5 MAP Adaptation of Model Subspace

It has been shown in previous section that, the globally shared parameters in SGMMs can be estimated by tying across multiple languages/systems to improve estimation accuracy. It has also been used in cross-lingual settings \[18, 19\], where the global parameters were
reused by the target language system, with only state dependent parameters being re-
estimated. Experiments have shown that significant performance improvements could
be achieved when training data for the target language is very limited, since the number
of parameters to be estimated is much smaller [19].

However, sharing the global parameter set across multiple languages can introduce a
mismatch with the target language system, owing to differences in phonetic character-
istics, corpus recording conditions, and speaking styles. Since the amount of training
data may not be sufficient to allow the global parameters to be updated using maximum
likelihood (ML), in this work we employ maximum a posteriori (MAP) adaptation [28].
In particular, we train the target language system using MAP adaptation of the pho-
netic subspace parameters with a matrix variate Gaussian prior distribution based on
the phonetic subspace parameters estimated in the multilingual system.

5.1 Auxiliary Function of MAP Criteria

In ML estimation of the phonetic subspace [3], the auxiliary function for $M_i$ is given by
equation (27). If a prior term is introduced, then the auxiliary function becomes:

$$\hat{Q}(M_i) = Q(M_i) + \tau \log P(M_i),$$

where $P(M_i)$ denotes the prior distribution of matrix $M_i$, and $\tau$ is the smoothing pa-
rameter which balances the relative contributions of the likelihood and prior. Although
any valid form of $P(M_i)$ may be used, in practical MAP applications a conjugate prior
distribution is often preferred for reasons of simplicity. In this work, $P(M_i)$ is set to be
a Gaussian distribution which is conjugate to the auxiliary $Q(M_i)$.

5.2 Matrix Variate Gaussian Prior

The Gaussian distribution of random matrices is well understood [29]. A typical ex-
ample of its application in speech recognition is maximum a posteriori linear regression
(MAPLR) [30] for speaker adaptation, in which a matrix variate prior was used for the
linear regression transformation matrix. The Gaussian distribution of a $D \times S$ matrix
$M$ is defined as:

$$\log P(M) = -\frac{1}{2} \left( DS \log(2\pi) + D \log |\Omega_r| + S \log |\Omega_c| \\
+ \text{tr}(\Omega_r^{-1}(M - \bar{M})\Omega_c^{-1}(M - \bar{M})^T) \right),$$

where $\bar{M}$ is a matrix containing the expectation of each element of $M$, and $\Omega_r$ and $\Omega_c$ are $D \times D$ and $S \times S$ positive definite matrices representing the covariance between the
rows and columns of $M$, respectively. $| \cdot |$ and $\text{tr}(\cdot)$ denote the determinant and trace
of a square matrix. This prior distribution is conjugate to auxiliary function (27). This matrix density Gaussian distribution may be written as:

\[ \text{Vec}(M) \sim N(\text{Vec}(\bar{M}), \Omega_r \otimes \Omega_c), \quad (48) \]

where \( \text{Vec}(\cdot) \) is the vectorization operation which maps a \( D \times S \) matrix into a \( DS \times 1 \) vector, and \( \otimes \) denotes the Kronecker product of two matrices. In this formulation, only \( \Omega_r \otimes \Omega_c \) is uniquely defined, and not the individual covariances \( \Omega_r \) and \( \Omega_c \), since for any \( \alpha > 0 \), \( (\alpha \Omega_r, \frac{1}{\alpha} \Omega_c) \) would lead to the same distribution. However, this is not of concern in the current application to MAP adaptation.

### 5.3 Prior Distribution Estimation

For MAP estimation, the prior distribution \( P(M_i) \) for each \( M_i \), should be estimated first. This requires the estimation of the mean matrices \( \bar{M}_i \), and the row and column covariances \( \Omega_r \) and \( \Omega_c \). Given a set of samples generated by \( P(M_i) \), the ML estimation of the mean, and the row and column covariances, is described by Dutilleul [31]. In MAPLR such samples are derived from clusters in the speaker independent model based on a regression tree [30]. In the case of cross-lingual SGMMs, the MAP formulation is based on the assumption that the multilingual estimate of the global subspace parameters serves a good starting point, which has been empirically verified earlier [19]. Recall that in the current cross-lingual system, the subspace parameters are obtained from an initial multilingual system trained on the source languages, and fixed during training of the state-specific parameters on the target language data. For its MAP counterpart, we set these multilingual parameters to be the mean of the prior \( P(M_i) \) and update both the state-specific \( v_{jm} \) and the global \( M_i \). With a sufficiently large value of \( \tau \) in (46), we can shrink the system back to the cross-lingual baseline, whereas \( \tau = 0 \) corresponds to the ML update.

The covariance matrices for each \( P(M_i) \) are global, estimated from the multilingual parameters by ML [31]. To be specific, suppose the set of multilingual phonetic subspace matrices is \( \{ \bar{M}_i, i = 1, \ldots, I \} \). We first compute the global mean as \( \bar{M} = \frac{1}{I} \sum_{i=1}^{I} M_i \). The two covariance matrices, \( \Omega_r \) and \( \Omega_c \), are then estimated by computing the following two equations iteratively until convergence:

\[
\begin{align*}
\Omega_r &= \frac{1}{ID} \sum_{i=1}^{I} (M_i - \bar{M}) \Omega_r^{-1} (M_i - \bar{M})^T \\
\Omega_c &= \frac{1}{IS} \sum_{i=1}^{I} (M_i - \bar{M})^T \Omega_c^{-1} (M_i - \bar{M}).
\end{align*}
\quad (49)
\]

\( \Omega_r, \Omega_c \) can be initialised as identity matrices, and several iterations are found to be sufficient for convergence. Hence the prior \( P(M_i) \) is parameterized by \( \bar{M}_i, \Omega_r, \) and \( \Omega_c \).

Povey [4] has discussed using a global prior over all the subspace matrices and presented a similar formulation. The principal differences in this work are that we are using multilingual subspace parameters as priors, and we have applied it in a cross-lingual
setting. In addition, we also note that it is possible to estimate the covariances using a data driven approach. For instance, a fully Bayesian treatment [32] can be applied, by which covariances can be estimated by maximizing the marginal likelihood

$$\arg \max_{\Omega, \Omega_c} \sum_i \int P(\mathbf{X} | \mathbf{M}_i) P(\mathbf{M}_i | \bar{\Omega}_i, \Omega_c) d\mathbf{M}_i,$$

(50)

where \( \mathbf{X} \) denotes all the acoustic frames. The likelihood \( P(\mathbf{X} | \mathbf{M}_i) \) can be approximated by its lower bound, i.e. the auxiliary function (27), and as we use the conjugate prior to the auxiliary function, the analytical form of the marginal likelihood is available. Hence, this approach is expected to be feasible in practice. We have not experimentally investigated this approach in this work.

5.4 MAP Estimation of the Phonetic Subspace

The detailed analytical solution of the MAP estimate of subspace parameters with Gaussian prior is given by Povey [4] (App. J). Here, we summarize the main ideas. By substituting (27) and (47) into (46), the auxiliary function of MAP can be rewritten as:

$$\tilde{Q}(\mathbf{M}_i) \propto \text{tr} \left( \mathbf{M}_i^T \Sigma^{-1}_i \mathbf{Y}_i + \tau \mathbf{M}_i^T \Omega^{-1}_r \bar{\mathbf{M}}_i \Omega^{-1}_c \right) - \frac{1}{2} \text{tr} \left( \Sigma^{-1}_i \mathbf{Q}_i \mathbf{M}_i^T + \tau \Omega^{-1}_r \mathbf{M}_i \Omega^{-1}_c \mathbf{M}_i^T \right).$$

(51)

The solution is not readily available by taking the derivative of \( \tilde{Q}(\mathbf{M}_i) \) with respect to \( \mathbf{M}_i \) and setting it to be zero. Instead, we introduce an intermediate transform \( \mathbf{T} = \mathbf{U}^T \mathbf{L}^{-1} \) that simultaneously diagonalises \( \Sigma^{-1}_i \) and \( \Omega^{-1}_r \), where

\[ \Sigma^{-1}_i = \mathbf{L} \mathbf{L}^T, \quad (\text{Cholesky decomposition}), \]

(52)

\[ \mathbf{S} = \mathbf{L}^{-1} \Omega^{-1}_r \mathbf{L}^{-T}, \]

(53)

\[ \mathbf{S} = \mathbf{U} \Lambda \mathbf{U}^T, \quad (\text{Eigenvalue decomposition}). \]

(54)

It is the case that \( \mathbf{T} \Sigma^{-1}_i \mathbf{T} = \mathbf{I} \) and \( \mathbf{T} \Omega^{-1}_r \mathbf{T} = \Lambda \), where \( \mathbf{I} \) is the identity matrix, and \( \Lambda \) is a diagonal matrix holding the eigenvalues of matrix \( \mathbf{S} \). If we further define \( \mathbf{M}_i = \mathbf{T} \mathbf{M}_i' \), then equation (51) can be rewritten as

$$\tilde{Q}(\mathbf{M}_i) \propto \text{tr} \left( \mathbf{M}_i'^T \mathbf{T} (\Sigma^{-1}_i \mathbf{Y}_i + \tau \Omega^{-1}_r \bar{\mathbf{M}}_i \Omega^{-1}_c) \right) - \frac{1}{2} \text{tr} \left( \mathbf{M}_i' \mathbf{Q}_i \mathbf{M}_i'^T + \tau \Lambda \mathbf{M}_i' \Omega^{-1}_c \mathbf{M}_i'^T \right).$$

(55)

Now we can take the derivative of \( \tilde{Q}(\mathbf{M}_i) \) with respect to \( \mathbf{M}_i' \):

$$\frac{\partial \tilde{Q}(\mathbf{M}_i)}{\partial \mathbf{M}_i'} = \mathbf{T} (\Sigma^{-1}_i \mathbf{Y}_i + \tau \Omega^{-1}_r \bar{\mathbf{M}}_i \Omega^{-1}_c) - \mathbf{M}_i' \mathbf{Q}_i - \tau \Lambda \mathbf{M}_i' \Omega^{-1}_c.$$

(56)
Setting this derivative to be zero, we obtain the row by row solution of $\mathbf{M}_i'$ as

$$m'_n = g_n (Q_i + \tau \lambda_n \Omega^{-1}_c)^{-1},$$

(57)

where $m'_n$ is the $n$th row of $\mathbf{M}_i'$, $\lambda_n$ is the $n$th diagonal element of $\mathbf{A}$, and $g_n$ is the $n$th row of matrix $T(\mathbf{\Sigma}^{-1}_i \mathbf{Y}_i + \tau \Omega^{-1}_r \mathbf{M}_i \Omega^{-1}_c)$. The final solution of $\mathbf{M}_i$ can then be obtained by $\mathbf{M}_i = \mathbf{T}^T \mathbf{M}_i'$.

As noted before, by setting $\tau \to \infty$, we shrink the system back to the cross-lingual baseline, and $\tau = 0$ corresponds to the ML estimate. If $\Omega_r = \Omega_c = \mathbf{I}$, the MAP estimate is equivalent to applying $\ell_2$-norm regularization on $\mathbf{M}_i$ with the model origin set to be the multilingual estimate (cf. equation (51)). In the experimental section, we will also study the roles of $\Omega_r$ and $\Omega_c$ individually, by setting the other to be $\mathbf{I}$.

### 5.5 Summary

In this section, we discussed the MAP adaptation of the phonetic subspace parameters in an SGMM acoustic model for cross-lingual speech recognition. In this approach, a matrix variate Gaussian prior is introduced to the subspace parameter estimation in order to avoid model overfitting in limited resource conditions. In this work, the phonetic subspace parameters estimated in the multilingual system are served as priors for the target language systems. Note that the MAP adaptation is formulated for the phonetic subspace $\mathbf{M}_i$, however, it’s equally applicable to speaker subspace $\mathbf{N}_i$ as the two parameter sets are mirror images of each other. In addition, it’s worthwhile to explore the Bayesian estimation of the prior parameters as well as the adaptation of the weight projections in future work.

### 6 Noise Adaptation of SGMM Acoustic Model

Robust speech recognition in noisy environment is one of the main research focuses due to the requirement of practical speech recognition applications in the real world. Over the past decade or two, many approaches have been proposed to improve the robustness of speech recognisers, which apply to both feature or model domains. Owning to utilising the power of modelling, model based noise compensation can normally achieve better performance. In particular, noise compensation based on vector Taylor series (VTS) has been successfully applied in HMM/GMM based speech recognition system [33]. However, VTS based noise compensation is computationally expensive as every Gaussian component in the acoustic model has to be adapted. This problem can be addressed by Joint Uncertainty Decoding (JUD) [34], in which the whole set of Gaussian components are clustered into a small number of classes by a regression tree. The mapping between clean and noise corrupted speech model is estimated from the regression model and the same mapping is applied all the components than belong to the same regression class.
has been shown in [35] that VTS and JUD are equivalent when the number of regression class is the same with the total number of Gaussians.

In this section, we investigate applying noise compensation by VTS and JUD to SGMM based acoustic model [3]. In an SGMM, the parameters of each Gaussian component are not estimated directly, but derived from a low dimensional model subspace. This allows a much larger number of Gaussians to be used by each HMM state while the total parameters to be estimated can be relatively smaller compared to conventional modelling. However, using larger number of Gaussians in SGMM brings challenges to apply VTS and JUD for noise compensation. Firstly, the computation burden will be much heavier for VTS-based approach, and for JUD, clustering the all the Gaussians is computationally expensive. In addition, the techniques such as pre-computation and Gaussian pruning may not work well after the noise adaptation, which also brings efficiency issues. In this section, we discuss our preliminary work address these challenges and present the results of noise adaptation with an ad hoc setting.

6.1 Mismatch Function

In this work, the mismatch function between clean and noise corrupted speech used for VTS and JUD is

\[
y_s = x_s + h + C \log \left(1 + \exp \left(C^{-1}(z - x_s - h)\right)\right) + 2 \alpha \cdot \exp \left(C^{-1}(z - x_s - h)/2\right) = f(x_s, h, z), \tag{58}
\]

where \(x_s\) and \(y_s\) denote the clean and noise corrupted static feature, \(z\) and \(h\) are additive and convolutional noise, respectively. \(C\) is the discrete cosine transform (DCT) matrix. \(\alpha\) is a vector of phase factors [36, 37]. In most applications, the phase term is not included. However, in this work, we found the phase term significantly affects the performance of JUD and VTS in SGMM acoustic model. It’s probably due to the strategy for clustering that we used. This will be discussed in detail in experimental section.

In most noise compensation schemes, the noise is normally modelled as

\[
z \sim N(u_n, \Sigma_n), \quad h = \mu_h, \tag{59}
\]

and as in most speech recognition systems, the clean speech is assumed to be Gaussian distributed, the noise corrupted speech is still approximated by Gaussian distribution in order to be compatible with recogniser, thought it’s “true” distribution may be very complex. In this case, the aim of noise adaptation is to estimate the mean \(\mu_y\) and covariance \(\Sigma_y\) of the noise corrupted speech.
6.2 Vector Taylor Series

In VTS, the noise corrupted speech model is estimated by first order Taylor series approximation, e.g. for \( m \)th Gaussian component the parameters for static feature is

\[
\mu^m_{ys} = f(\mu^m_{xs}, \mu_h, \mu_z)
\]

(60)

\[
\Sigma^m_{ys} = J^m\Sigma^m_{xs}J^mT + (I - J^m)\Sigma_n(I - J^m)^T
\]

(61)

where \( \mu_s \) and \( \Sigma_s \) denote the mean and covariance for static coefficient. Similar notations are used for delta and double-delta coefficients. \( J_m \) is the derivative of \( y_s \) with respect to \( x_s \) at the expansion point \( \mu_x \)

\[
J^m = \frac{\partial y_s}{\partial x_s}
\]

| \( F^m \) is a diagonal matrix with the \( i \)th element as

\[
f^m_{ii} = \frac{1 + \alpha_i \exp(B_i/2)}{1 + \exp(B_i/2) + 2\alpha_i \exp(B_i/2)}
\]

(63)

\[
B_i = [C^{-1}]i(z - x_s - h)
\]

(64)

where \( [C]_i \) is the \( i \)th row of \( C \). For dynamic features, continuos-time approximation [38] is used to derive their parameters, e.g. for delta features,

\[
\mu^m_{yd} \approx J^m\mu_{xd}
\]

(65)

\[
\Sigma^m_{yd} \approx J^m\Sigma^m_{xd}J^mT + (I - J^m)\Sigma_n(I - J^m)^T
\]

(66)

and same approach is used for delta-delta coefficients. In recent works, rather than using first order Taylor series approximation, higher order of VTS to both model and feature domains is investigated and shown good performance [39, 40]. It also notes that in conventional HMM/GMM systems, the covariance of clean speech model \( \Sigma_x \) is diagonal, while it's full in SGMMs. After noise compensation, \( \Sigma_y \) will be always full as the transform \( J^m \) is full matrix. However, \( \Sigma_y \) is normally diagonalised after compensation for efficiency in conventional setups.

6.3 Joint Uncertainty Decoding

The major issue with VTS-based noise compensation lies in its computational requirement. Even do not consider the cost of noise model estimation, since every Gaussian component should be adapted as equation (60) and (61), it is very computationally expensive to apply VTS noise adaptation for large vocabulary task, especially when the number of Gaussians is large. Joint uncertainty decoding (JUD) is proposed to approximate VTS while saving the computation by clustering the total Gaussian component.
into much smaller number of classes, and for $m$th component, the approximation is

$$p(y|m) = \int p(y|x,m)p(x|m)dx \quad (67)$$

$$\approx \int p(y|x,r_m)p(x|r_m)dx \quad (68)$$

where $r_m$ denotes the regression class that component $m$ belongs to, and the conditional distribution $p(y|x,r_m)$ is derived from the joint distribution of clean and noise corrupted speech which is assumed to be Gaussian, for $r_{th}$ regression class

$$p\left(\begin{bmatrix} x \\ y \end{bmatrix} | r \right) \sim \mathcal{N}\left(\begin{bmatrix} \mu^r_x \\ \mu^r_y \end{bmatrix}, \begin{bmatrix} \Sigma^r_x & \Sigma^r_{yx} \\ \Sigma^r_{xy} & \Sigma^r_y \end{bmatrix}\right), \quad (69)$$

which gives conditional distribution $p(y|x,r_m)$ as

$$\mu^r_{y|x} = \mu^r_y + \Sigma^r_x \Sigma^{-1}_{xy} (x - \mu^r_x) \quad (70)$$

$$\Sigma^r_{y|x} = \Sigma^r_y - \Sigma^r_{yx} \Sigma^{-1}_{xy} \Sigma^r_{xy} \quad (71)$$

By marginalising the likelihood in expression (68), the likelihood of corrupted speech on $m_{th}$ component can be approximated as

$$p(y|m) \approx |A^r| \mathcal{N}\left( A^r y + b^r; \mu_m, \Sigma_m + \Sigma_b^r \right) \quad (72)$$

where

$$A^r = \Sigma_x^r \Sigma^{-1}_{xy} \quad (73)$$

$$b^r = \mu_x^r - A^r \mu_x^r \quad (74)$$

$$\Sigma_b^r = A^r \Sigma_y^r A^T x - \Sigma_x^r \quad (75)$$

The JUD transforms $\{A^r, b^r, \Sigma^r\}$ are computed for each regression class, and the Gaussian components belonging to the same class share the same JUD transforms. Note that the JUD transform is similar to CMLLR transforms, except for a bias term $\Sigma_b^r$ for covariance, which intuitively means after noise compensation, the uncertainty of the model increase. The cross covariance $\Sigma^{-1}_{xy}$ in (73) can be obtained by VTS algorithm as [35]

$$\Sigma^{-1}_{xy} = diag(\Sigma_{xs}^r J^T, \Sigma_{xd}^r J^T, \Sigma_{xdd}^r J^T) \quad (76)$$

where $J^r$ is derived from regression class $r$. “diag” denotes the block-diagonal operation of the three matrices which corresponding to static, delta and double-delta coefficients.

6.4 VTS and JUD for SGMM Acoustic Model

Applying VTS and JUD to SGMM based acoustic model is not a trivial task. A typical feature of SGMMs is much larger number of Gaussian components are used for each
HMM state. For example, the GMM system we used on Aurora 4 task has about 50k Gaussian components, however, the corresponding SGMM system has 16k sub-states, and each is modelled by 400 Gaussians \((I = 400)\), which gives 6.4 million Gaussian components in total. In addition, unlike conventional GMM systems, SGMM acoustic model uses full covariance matrices. Thought they are globally shared, after VTS compensation however, they will no longer be as in equation (61) the transform \(J^m\) depends on each Gaussian component. In addition, the component mean will be expended as well as the compact representation form as equation (2) will be broken. This will bring huge burden to both computation and memory.

However, this problem can be circumvented by JUD style compensation as the model does not need to be expended, and the noise compensation is done by JUD transforms on feature level. Though the covariance should be updated, \(\Sigma^b\) depends on the regression class which is normally much smaller than the number of Gaussians. The key question here is how to obtain the regression classes. It is costly, but not impossible to apply the conventional clustering by a regression tree as in GMM system to the total Gaussian components (e.g. 6.4 million in our case) in SGMM acoustic model. In this work, however, we assume the UBM itself which is used for Gaussian pruning is good clustering of all the SGMM Gaussian components as the one to one mapping from the UBM to the acoustic model is established during model training. Another consideration for this is after noise compensation, we still need the noise adapted UBM for Gaussian selection to speed up the decoding, hence the mapping between UBM and SGMM should be kept well. In addition, using UBM as clustering only require small modification of current decoder.

### 6.4.1 Noise model estimation

As the noise is latent, its parameters can not be estimated directly, however, this can be addressed by expectation-maximisation (EM) algorithm which iteratively update the noise and noise corrupted acoustic model. Denote the noise model as \(\mathcal{M}_n = \{\mu_n, \Sigma_n, \mu_h\}\), the auxiliary function for noise model update is

\[
Q(\tilde{\mathcal{M}}_n, \mathcal{M}_n) = \sum_{t,m} \gamma_{mt} \log(p_y(t|m; \mathcal{M}_n, \tilde{\mathcal{M}}_n)),
\]

where \(\tilde{\mathcal{M}}_n\) is the “new” noise model. The auxiliary function will depend on either VTS or JUD is used for noise compensation. For instance, the likelihood will be approximated by equation (72) in JUD based approach, while for VTS, the likelihood is in standard form as the acoustic model parameters have been updated by noise model.

The close form solution of updating the noise parameters is normally not available by maximising the auxiliary function (77). In this case, gradient based optimisation scheme
can be use, e.g. for additive noise mean $\mu_n$

$$\mu_n = \mu_n - \zeta \left( \frac{\partial^2 Q(\cdot)}{\partial^2 \mu_n} \right)^{-1} \left( \frac{\partial Q(\cdot)}{\partial \mu_n} \right)$$

(78)

where $\zeta$ is the learning rate. Same approach can be used for other noise parameters. Optimally, the noise model should be estimated in conjunction with the acoustic model that to be compensated, i.e. the SGMM acoustic model should be used in the auxiliary function (77). In this work, however, we use the UBM model to estimate the noise parameters as the UBM is treated as the regression model of all the SGMM components. Using the regression model to estimate the noise model parameters is a standard setup for JUD system, and it will serve as a baseline for the future work.

6.5 Summary

In this section, we have discussed the noise compensation of SGMM acoustic model based on vector Taylor series (VTS) and joint uncertainty decoding (JUD) for noise robust speech recognition. After reviewing the VTS and JUD algorithms, we discussed the challenges of applying VTS and JUD to SGMM acoustic model due to large number of Gaussian components. We have also shown the motivation of using UBM as regression model for both JUD compensation and noise model estimation. Further discussion will be given in the experimental section.

7 Experiments

In this section, we report the experimental results of regularized model estimation, cross-lingual speech recognition systems, MAP adaptation and noise compensation for SGMM acoustic models discussed in this report. The experiments were conducted on three different corpus, namely, Wall Street Journal (WSJ) for regularization experiments, GlobalPhone corpus for cross-lingual and MAP adaptation experiments and Aurora 4 for noise compensation experiments.

7.1 Experiments of Regularized Model Estimation

Experiments for regularized model estimation were carried out on the WSJ-5k dataset. We follow the setup described in [41]. The training set contains 7137 utterances with a total duration of about 14 hours (after removing silence). For testing, we use subset of the WSJ1-5k development set obtained by deleting sentences with out-of-vocabulary words giving a total of 248 sentences from 10 speakers. We use the standard 5k
non-verbalised punctuation bigram language model (LM) for decoding. Standard 13-dimension MFCC+Δ+ΔΔ features were used with cepstral mean and variance normalisation. The following results were obtained by tuning the LM scaling factor and word insertion penalty to get the best word error rate (WER).

### 7.1.1 Baseline System

We first train a conventional HMM-GMM baseline recognizer using the HTK speech recognition toolkit [42]. The baseline system has 3093 tied cross-word triphone states, each with a 16-component GMM with diagonal covariance. Our baseline result of 10.3% WER on the test set is comparable to the 10.48% WER reported in [41] using a similar configuration. Starting from the HTK baseline system, we train the SGMM system according to the recipe using the Kaldi software described in [3], using 400 Gaussian components in the universal background model (UBM) and 40-dimensional phonetic subspace (i.e., $S = 40$). State splitting was applied to increase the number of sub-states for large model capacity. The best performance of SGMM baseline is 8.6%, which gives more than 15% relative improvement compared to the conventional system.

### 7.1.2 SGMM results with smoothing and renormalization

We first compare the performance of ad-hoc smoothing shown in equation (32). The results are given in Table 1 for different values of the smoothing parameter $\tau$. We also present the results by renormalization denoted as $R(v)$ in Table 1. While we do not observe much improvements from the ad-hoc smoothing approach, from the results of using a small smoothing term ($\tau = 5$) compared to the non-smoothed case ($\tau = 0$), the smoothing terms can indeed help to address the overfitting issue, albeit rather mildly. Renormalization, however, is beneficial to both system performance and model robustness. While theoretically, renormalization does not change the model, in practice it makes a difference due to issues like numerical stability of the updates, flooring, condition limiting of matrices, etc.

### 7.1.3 SGMM results with regularization

Here the regularization is applied at the sub-state level for systems with sub-state splitting. The regularization parameter is set to be global and constant for different numbers of sub-states, and except for regularized estimation of the sub-state vectors, the SGMM training follows the recipe in [3].

Table 2 shows the results of regularization with $\ell_1$, $\ell_2$ as well as elastic net penalty for systems with and without renormalization. For the systems without renormalization, the regularization parameters are set to be 10 for all $\ell_1$, $\ell_2$ and elastic net systems (i.e.
Table 1: Word error rates of SGMM acoustic model with ad-hoc smoothing or renormalization, $S = 40$

<table>
<thead>
<tr>
<th>#Substates</th>
<th>$R(\nu)$</th>
<th>$\tau = 0$</th>
<th>$\tau = 5$</th>
<th>$\tau = 10$</th>
<th>$\tau = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3k</td>
<td>9.7</td>
<td>9.8</td>
<td>9.9</td>
<td>10.0</td>
<td>10.1</td>
</tr>
<tr>
<td>4.5k</td>
<td>9.7</td>
<td>9.6</td>
<td>9.7</td>
<td>9.7</td>
<td>9.8</td>
</tr>
<tr>
<td>6k</td>
<td>9.7</td>
<td>9.4</td>
<td>9.4</td>
<td>9.5</td>
<td>9.6</td>
</tr>
<tr>
<td>12k</td>
<td>9.0</td>
<td>8.8</td>
<td>8.9</td>
<td>9.1</td>
<td>9.1</td>
</tr>
<tr>
<td>16k</td>
<td>8.8</td>
<td>8.6</td>
<td>8.8</td>
<td>8.9</td>
<td>8.6</td>
</tr>
<tr>
<td>20k</td>
<td>8.8</td>
<td>8.7</td>
<td>8.7</td>
<td>9.3</td>
<td>8.9</td>
</tr>
<tr>
<td>24k</td>
<td>8.3</td>
<td>8.8</td>
<td>8.6</td>
<td>9.1</td>
<td>8.8</td>
</tr>
<tr>
<td>28k</td>
<td>8.5</td>
<td>8.7</td>
<td>8.7</td>
<td>9.1</td>
<td>8.8</td>
</tr>
<tr>
<td>32k</td>
<td>8.7</td>
<td>9.0</td>
<td>8.5</td>
<td>9.4</td>
<td>9.7</td>
</tr>
</tbody>
</table>

GMM baseline: 10.3

Table 2: Comparison of SGMM acoustic model with regularized (sub-)state vector estimation, $S = 40$

<table>
<thead>
<tr>
<th>#Substates</th>
<th>without renormalization</th>
<th>with renormalization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$eNet$</td>
<td>$eNet$</td>
</tr>
<tr>
<td>3k</td>
<td>-</td>
<td>9.7 10.2 9.7 9.9</td>
</tr>
<tr>
<td>4.5k</td>
<td>-</td>
<td>9.7 9.8 9.7 9.9</td>
</tr>
<tr>
<td>6k</td>
<td>-</td>
<td>9.7 9.7 9.4 9.6</td>
</tr>
<tr>
<td>12k</td>
<td>-</td>
<td>9.0 8.8 9.1 9.5</td>
</tr>
<tr>
<td>16k</td>
<td>8.6 8.8 8.4 8.7</td>
<td>8.8 8.9 8.9 9.1</td>
</tr>
<tr>
<td>20k</td>
<td>8.7 8.3 8.8 8.6</td>
<td>8.8 8.7 8.4 9.2</td>
</tr>
<tr>
<td>24k</td>
<td>8.8 8.4 8.7 8.5</td>
<td>8.3 8.5 8.6 9.0</td>
</tr>
<tr>
<td>28k</td>
<td>8.7 8.4 8.5 8.5</td>
<td>8.5 8.4 8.7 9.2</td>
</tr>
<tr>
<td>32k</td>
<td>9.0 8.5 8.5 8.8</td>
<td>8.7 8.3 9.0 9.2</td>
</tr>
</tbody>
</table>
Table 3: The number of phones and speakers, the amount of training and development data (hours) for the 4 languages used in this paper.

<table>
<thead>
<tr>
<th>Language</th>
<th>#Phones</th>
<th>#Speakers</th>
<th>Trn(h)</th>
<th>Dev(h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>German (GE)</td>
<td>44</td>
<td>77</td>
<td>14.8</td>
<td>2.0</td>
</tr>
<tr>
<td>Spanish (SP)</td>
<td>43</td>
<td>97</td>
<td>17.2</td>
<td>2.0</td>
</tr>
<tr>
<td>Portuguese (PT)</td>
<td>48</td>
<td>101</td>
<td>22.6</td>
<td>1.5</td>
</tr>
<tr>
<td>Swedish (SW)</td>
<td>52</td>
<td>98</td>
<td>17.4</td>
<td>2.0</td>
</tr>
</tbody>
</table>

$\lambda_1 = \lambda_2 = 10$ in equation 35). Compared to the baseline, the SGMM system with regularization is less likely to suffer from overfitting, as the best results are achieved by models with large capacity, and also obtain moderate improvement, which agrees with the argument of regularization in this paper. We do not observe significant difference between $\ell_1$ and $\ell_2$-norm penalty in terms of performance, and elastic net do not give further gains. In our experiments, $\ell_1$ penalty does not give sparse solution when the number of sub-states is small, however, with further sub-state splitting, a considerable amount of sub-state vectors are driven to be sparse, e.g. the proportion of zero entries can be 10%-20% for some of them.

With renormalization, the regularization is still efficient in avoiding model overfitting with larger models, as shown by the results in Table 2. However, we do not observe performance gains in this case. This shows that, in the previous setting, regularization was providing better performance by improving the numerical stability of the updates. It is worth noting that with renormalization, the regularization parameters need to be much smaller, for example we use $\lambda_1 = \lambda_2 = 2$ for these experiments. Also, the system is more sensitive to the choice of the regularization parameters. This corroborates with the assumption that without renormalization, the updates of the globally-shared parameters $M$ and $w$ can ameliorate over-penalization of the state-vectors to an extant.

7.2 Experiments of Cross-lingual Systems

Cross-lingual speech recognition experiments were performed on the GlobalPhone corpus [23], with German as the target language. Our main experiments have investigated low-resource cases, in which we have used one hour and five hours of target language acoustic training data. Before presenting these cross-lingual experiments, we give a brief description of the corpus and system configuration for our experiments, and give results for the baseline monolingual systems, trained on the complete data sets.

7.2.1 GlobalPhone Corpus

The GlobalPhone corpus [23] contains up to 20 languages including English, Arabic, Chinese and a number of European languages, and consists of recordings of a range of
Table 4: Word Error Rates (WER %) of GMM and SGMM baseline of target and source languages on Dev dataset.

<table>
<thead>
<tr>
<th>Language</th>
<th>LM</th>
<th>PPL</th>
<th>OOV</th>
<th>Dict</th>
<th>GMM</th>
<th>SGMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>GE</td>
<td>Trigram</td>
<td>422</td>
<td>5.2%</td>
<td>17k</td>
<td>25.7</td>
<td>24.0</td>
</tr>
<tr>
<td>SP</td>
<td>Bigram</td>
<td>306</td>
<td>4.8%</td>
<td>17k</td>
<td>33.7</td>
<td>30.4</td>
</tr>
<tr>
<td>PT</td>
<td>Bigram</td>
<td>393</td>
<td>4.3%</td>
<td>52k</td>
<td>29.3</td>
<td>25.9</td>
</tr>
<tr>
<td>SW</td>
<td>Trigram</td>
<td>940</td>
<td>0%</td>
<td>23k</td>
<td>47.2</td>
<td>40.8</td>
</tr>
</tbody>
</table>

Table 5: Total trace of covariance and subspace matrices given by the source SGMM systems, $S = 40$.

<table>
<thead>
<tr>
<th></th>
<th>SP</th>
<th>PT</th>
<th>SW</th>
<th>Multilingual</th>
</tr>
</thead>
<tbody>
<tr>
<td># of states</td>
<td>2298</td>
<td>3140</td>
<td>3153</td>
<td>-</td>
</tr>
<tr>
<td># of sub-states</td>
<td>20k</td>
<td>20k</td>
<td>20k</td>
<td>-</td>
</tr>
<tr>
<td>$\sum_i tr(\Sigma_i)/10^3$</td>
<td>8.02</td>
<td>8.07</td>
<td>8.14</td>
<td>8.15</td>
</tr>
<tr>
<td>$\sum_i tr(M_iM_i^T)/10^3$</td>
<td>16.1</td>
<td>12.9</td>
<td>11.9</td>
<td>11.2</td>
</tr>
</tbody>
</table>

speakers reading newspapers in their native language. There are about 100 speakers for each language, and recordings were made under a range of ‘quiet’ conditions, resulting in about 15–20 hours of high quality speech for each language. However, since the recording locations vary, acoustic conditions also vary both within and between each language. Hence, corpus mismatch may degrade the performance of cross-lingual systems.

In these experiments, German (GE) was used as the target language, and Spanish (SP), Portuguese (PT), and Swedish (SW) as the source languages. Table 3 describes the data for each language used in the experiments in terms of the number of phonemes and speakers, and the amount of training and development data. Our baseline monolingual systems, described below, used the complete training sets for each language. In the cross-lingual experiments (section 7.2.3), we used the full training sets for the source languages, but limited training sets (1 hour and 5 hours) for the target language.

### 7.2.2 Baseline Monolingual Systems

We constructed GMM-based baseline systems for each of the four languages, based on the system of Lal [16]. We used 12th order mel-frequency cepstral coefficients, plus energy, with first and second derivatives, to give a 39-dimension acoustic feature vector. We applied cepstral mean and variance normalization, and used HTK to build the acoustic models. For the German GMM baseline, we used 3125 triphone states, and 16 mixture components.

The baseline monolingual SGMM systems used the same acoustic feature vectors and the same context dependent phone clustering as the corresponding baseline GMM system. We set the number of Gaussians $I = 400$, and the sub-state vector dimension $S = 40$.  

27
The open source Kaldi software was used for the SGMM systems in this paper. Table 4 gives the baseline monolingual WERs for each of the four languages. We used trigram language models for all experiments, except the baseline monolingual systems for Spanish and Portuguese.

Following Burget et al. [18], a multilingual SGMM was constructed by tying $\Sigma_i$, $M_i$ and $w_i$ across the source language (SP, PT, SW) monolingual SGMM systems, i.e. those parameters were updated by summing up all the related statistics across the source languages. This means the number of parameters in multilingual and monolingual system are the same for each language, and the only difference is in estimating the globally shared parameters. In this paper, we renormalized the phonetic subspace [4] (Appendix K) to concentrate the most important variation in the lower-numbered dimensions. This also allowed us to use a lower dimension subspace for cross-lingual systems without retraining the subspace parameters $M_i$ and $w_i$. Table 5 shows the size of $\Sigma_i$ and $M_i$ estimated in both monolingual and multilingual fashion.

### 7.2.3 Cross-lingual experiments

Here, we present results of the cross-lingual acoustic modelling experiments, in cases where we have limited target language resources. Our experiments use German as the target language and we have investigated two levels of limited acoustic training data, 1 hour and 5 hours, in order to provide an estimate of the amount of transcribed speech needed for an acceptable recognition accuracy. These training data subsets were randomly selected from the complete German training set, from which we selected 7–8 minutes of recorded speech from each of 8 and 40 speakers for the 1 hour and 5 hour systems respectively. The globally-shared SGMM parameters for these cross-lingual systems were obtained from monolingual source language systems, or from the multilingually-trained SGMM using all the source languages.

With 1 hour training data, we trained the target language (GE) GMM baseline system which had 620 triphone states, each of which was modelled using 4-component GMMs. The baseline and the cross-lingual SGMM systems used the same context-dependent phonetic clustering as the GMM system, and the dimension of sub-state vector is set to be $S = 20$ for all the SGMM systems. In the baseline SGMM systems, all the parameters in equations (1–3) were updated: the sub-state vectors $v_{jm}$ and the globally shared parameters $M_i$, $w_i$ and $\Sigma_i$. In the cross-lingual systems, only the sub-state vectors $v_{jm}$ were re-estimated, with the globally shared parameters taken from the source language or multilingual systems. Hence, with the same number of sub-states, the number of active parameters are the same for all cross-lingual systems, but there are much smaller than that in the corresponding baseline SGMM system. As discussed earlier, the subspace parameters were renormalized and we only used a 20-dimension subspace for the cross-lingual system without retraining these parameters.

In contrast to Burget et al. [18], the number of tied states was fixed rather than being
Figure 1: 1 Hour Training Data: WER of baseline GMM and SGMM system (41.2% vs. 38.0%) as well as cross-lingual systems. For all SGMM systems, the dimensionality of the sub-state vectors is set to be $S = 20$. The lowest WER (35%) is obtained from the multilingual subspace system (w/Mul).

Figure 2: 1 Hour Training Data: Training of 40-dimensional sub-state vectors showed numerical instability after a few iterations. Condition number of the covariance matrix of the sub-state vectors start increasing rapidly and the estimation failed, leading to a decrease in the log-likelihood. This was fixed using a regularized sub-state vector update.
Figure 3: 1 Hour Training Data: WER of cross-lingual systems with and without $\ell_1$-norm regularization. Regularization does not bring performance gains but slight degradation for systems with 20-dimension state vectors, and there are omitted in the figure for clarity. For the multilingual subspace system (w/Mul) with $S = 40$, the best performance is 32.7% by original regularization and 31.9% by modified regularization (M).

Table 6: Mean and Variance of the 1st coefficient in the state vectors of the cross-lingual systems without regularization.

<table>
<thead>
<tr>
<th>System</th>
<th>w/SW</th>
<th>w/SP</th>
<th>w/PT</th>
<th>w/Mul</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>1.006</td>
<td>0.889</td>
<td>0.946</td>
<td>1.019</td>
</tr>
<tr>
<td>V arance ($\times 10^{-3}$)</td>
<td>9.75</td>
<td>16.83</td>
<td>17.53</td>
<td>9.33</td>
</tr>
</tbody>
</table>

increased (from 500 (GMM) to 1000 (SGMM) to 1500 (multilingual SGMM), in their experiments on CallHome). This clearly demonstrates that the improvements are due to better estimation of the parameters, and not because the SGMM systems allows for estimation of a larger number of context-dependent models for the same amount of data. We expect that the results reported here will improve further by increasing the number of tied states.

The WERs for the monolingual GMM and SGMM systems trained on 1 hour of data were 41.2% and 38.0% respectively, a significant increase in WER compared with the case when they were trained with the complete 14.8 hour training data set. The SGMM system again has a considerably lower WER than the GMM system, as was observed by Burget et al. [18]. The performance of the cross-lingual systems is shown in Figure 1 with the globally shared parameters obtained from each of the source language systems, as well as the tied multilingual system. The results indicate that the system with multilingually
Table 7: WER of GMM and SGMM baseline systems with 5 hour training data.

<table>
<thead>
<tr>
<th>System</th>
<th>WER(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM baseline</td>
<td>34.3</td>
</tr>
<tr>
<td>SGMM baseline, $S = 20$</td>
<td>31.1</td>
</tr>
<tr>
<td>SGMM baseline, $S = 40$</td>
<td>32.0</td>
</tr>
</tbody>
</table>

Figure 4: 5 Hour Training Data Case: WER of cross-lingual systems. In these experiments, the dimension of state vectors is 20. The best performance is achieved by multilingual subspace system denoted as "w/Mul" and the WER is 28.6% which is considerably better than 31.1% by SGMM baseline.

Trained subspace parameters results in considerably lower WERs compared with the other cross-lingual systems derived from a single source language, as well as compared with the SGMM baseline. We may also observe that the cross-lingual system with Spanish subspace (denoted as “w/SP”) results in a higher WER compared with the other cross-lingual systems. In addition to factors such as linguistic differences and corpus mismatch, this difference may also be due to the larger model subspace in the Spanish system (Table 5) which may make it harder for the data to saturate the model.

While training a model with 40-dimensional sub-state vectors (i.e. $S = 40$) we encountered numerical instabilities. This is shown in Figure 2, where the condition number of the covariance matrix of the sub-state vectors start increasing rapidly and the estimation failed, leading to a decrease in the log-likelihood. Other measures like the determinant and trace of the covariance of $v_{jm}$ as well as the maximum and minimum elements of the vectors show similar trends. The precise reason for this instability is currently under investigation.

However, using a regularized estimation of the 40-dimensional sub-state vectors, with a
Figure 5: 5 Hour Training Data Case: WER of cross-lingual systems with and without the original $\ell_1$-norm regularization and the modified regularization (M).

Figure 6: Sparsity achieved by $\ell_1$-norm regularization for the “w/Mul + regularization” system in Figure 5. With larger number of sub-states, hundreds of sub-state vectors are set to the zero-vector.
relatively strong penalty on their $\ell_1$-norm, we could not only train the systems but also observe significantly lower WERs for all cross-lingual systems (Figure 3). Applying $\ell_1$-norm regularization to the 20-dimension cross-lingual SGMM systems results in slightly higher WERs, even with a relatively weak regularization penalty. This is probably due to the relative simplicity of the 20-dimensional subspace model.

Results for the modified regularization (equation 43) are also shown in Figure 3. The modification is based on the assumption that the first column of $\mathbf{M}_i$ corresponds to global means, which should be learned by fixing the first coefficient of $\mathbf{v}_{jm}$ to 1. However, the source language systems were not trained with this constraint, leading to a potential mismatch. Yet we found that modified regularization led to modest improvements in WER for subspaces trained using Swedish and multilingual data, but not for Spanish and Portuguese. Table 6 shows that this was due to serendipity, as the first element of the state vectors had a mean value of nearly 1 with low variance for the Swedish and multilingual systems, but that was not the case for the other two systems in which modified regularization was ineffective.

We increased the amount of target language acoustic training data to 5 hours and trained the baseline monolingual GMM and SGMM systems. The GMM system had 1561 tied triphone states and each state is modelled by an 8-component GMM. As before, all the following SGMM systems share a context-dependent phonetic clustering with GMM system. WERs of these baseline monolingual systems are shown in Table 7.

The results of the cross-lingual systems are shown in Figure 4 where the dimension of sub-state vectors are still $S = 20$. Again, the cross-lingual system with multilingual subspace parameters denoted as (“w/Mul”) results in the lowest WER. The results of the regularized cross-lingual SGMM systems are shown (for the multilingual subspace) in Figure 5, and the results are broadly consistent with the 1 hour case. We did not observe any reductions in WER when regularizing the baseline system ($S = 20$), but we were again able to observe significant reductions in WER when regularizing a cross-lingual SGMM with dimension $S = 40$. A small improvement in WER was observed using the modified regularization approach, from 26.8% (original) to 26.6% (modified).

### 7.2.4 Sparsity Analysis

The $\ell_1$-norm regularization used in this paper is able to penalize the model complexity of cross-lingual systems in order to achieve model robustness. In addition, it also has the effect of driving some coefficients to zero, thus leading to a kind of variable selection, where the most relevant bases from $\mathbf{M}_i$ and $\mathbf{w}_i$ get used. In Figure 6 we can see the proportion of parameters set to zero by the $\ell_1$-regularization. Not surprisingly, with sub-state splitting, the sub-state vectors are driven to be increasingly sparse as the amount of acoustic frames aligned to each sub-state decrease accordingly.

\[^1\text{Once again, it was not possible to train a system with } S = 40 \text{ without regularization.}\]
7.3 Experiments of MAP Adaptation

We have carried out experiments using a cross-lingual acoustic model trained using the GlobalPhone corpus [23]. We chose German to be the target language, and Spanish, Portuguese and Swedish as source languages. In low-resource cross-lingual experiments, we selected two random subsets of transcribed audio in the target language, of 1 hour and 5 hours duration, containing speech from 8 and 40 speakers respectively\(^2\). We estimated the globally shared parameters in a multilingual fashion by tying \(M_i\), \(w_i\), and \(\Sigma_i\) across the three source language SGMM systems. The number of Gaussians \(I\) was 400 (cf. equation (1)). The models were evaluated on a development data set which containing about 2 hours of speech. For decoding, we used a trigram language model with a 17,000 word lexicon that was provided with the corpus. The language model had a perplexity of 442 on the development set, with an out of vocabulary rate of 5.2%. Further details of this cross-lingual system can be found in [19].

7.3.1 Baseline results

The results of monolingual and cross-lingual German systems, with different amounts of training data and sizes of phonetic subspace, are given in Table 8. In the monolingual systems, all the parameters are estimated from the 1 or 5 hours of available training data. In cross-lingual SGMM systems, the globally shared parameters are taken from a multilingual system trained on Spanish, Portuguese and Swedish, and only sub-state vectors \(v_{jm}\) and weights \(c_{jm}\) (equation 1, 2) are updated during model training. The GMM and SGMM systems for the same amount of training data use the same phonetic decision tree. Hence, the performance differences are purely owing to better parameter estimation. For SGMM systems with \(S = 40\), regularized state vector estimation by \(\ell_1\)-norm penalty [5] is applied to improve numerical stability; we have also observed that such regularisation brings gains in accuracy [19]. For comparison, the monolingual GMM and SGMM systems with the entire 14.8 hours of target language training data available in GlobalPhone achieve 25.7% and 24.0% WER.

7.3.2 MAP adaptation results

Our MAP experiments started from the cross-lingual SGMM systems in Table 8, with the MAP update of \(M_i\) performed for several iterations until convergence, while \(w_i\) and \(\Sigma_i\) were kept fixed. We compare different configurations of the row and column covariances for the priors as shown in Table 9, where the results are obtained by tuning the smoothing parameter \(\tau\) to be optimal on the development set. As mentioned before, setting \(\tau = 0\) is equivalent to an ML update of \(M_i\). When \(S = 20\), ML update provided considerable improvements with both the 1 hour and 5 hour data, since the number of

\(^2\)Although German should not be considered a low resource language, the GlobalPhone corpus provides a controlled and standardised experimental environment for experiments of this nature.
Table 8: WER (%) of baseline monolingual and cross-lingual baseline systems with 1 hour and 5 hour training data, \( S \) denotes the dimension of phonetic subspace.

<table>
<thead>
<tr>
<th>Training Data</th>
<th>System</th>
<th>WER</th>
<th>#states</th>
<th># sub-states</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 hour</td>
<td>Mono-GMM</td>
<td>41.2</td>
<td>620</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Mono-SGMM ( S = 20 )</td>
<td>38.0</td>
<td>620</td>
<td>2k</td>
</tr>
<tr>
<td></td>
<td>Cross-SGMM ( S = 20 )</td>
<td>35.0</td>
<td>620</td>
<td>12.8k</td>
</tr>
<tr>
<td></td>
<td>Cross-SGMM ( S = 40 )</td>
<td>32.7</td>
<td>620</td>
<td>4.4k</td>
</tr>
<tr>
<td>5 hour</td>
<td>Mono-GMM</td>
<td>34.3</td>
<td>1561</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Mono-SGMM ( S = 20 )</td>
<td>31.1</td>
<td>1561</td>
<td>6.7k</td>
</tr>
<tr>
<td></td>
<td>Cross-SGMM ( S = 20 )</td>
<td>28.6</td>
<td>1561</td>
<td>12k</td>
</tr>
<tr>
<td></td>
<td>Cross-SGMM ( S = 40 )</td>
<td>26.8</td>
<td>1561</td>
<td>12k</td>
</tr>
</tbody>
</table>

Table 9: WER (%) of MAP adapted systems.

<table>
<thead>
<tr>
<th>System</th>
<th>1 hour</th>
<th>5 hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-lingual baseline ((S = 20))</td>
<td>35.0</td>
<td>28.6</td>
</tr>
<tr>
<td>with ML update ((\tau = 0))</td>
<td>33.5</td>
<td>27.7</td>
</tr>
<tr>
<td>with MAP update ((I \otimes I))</td>
<td>32.1</td>
<td>26.7</td>
</tr>
<tr>
<td>with MAP update ((I \otimes \Omega_c))</td>
<td>32.2</td>
<td>26.8</td>
</tr>
<tr>
<td>with MAP update ((\Omega_r \otimes I))</td>
<td>32.2</td>
<td>26.8</td>
</tr>
<tr>
<td>with MAP update ((\Omega_r \otimes \Omega_c))</td>
<td>32.2</td>
<td>26.9</td>
</tr>
<tr>
<td>Cross-lingual baseline ((S = 40))</td>
<td>32.7</td>
<td>26.8</td>
</tr>
<tr>
<td>with ML update ((\tau = 0))</td>
<td>33.3</td>
<td>27.8</td>
</tr>
<tr>
<td>with MAP update ((I \otimes I))</td>
<td>31.1</td>
<td>25.6</td>
</tr>
<tr>
<td>with MAP update ((I \otimes \Omega_r))</td>
<td>31.3</td>
<td>25.9</td>
</tr>
<tr>
<td>with MAP update ((\Omega_r \otimes I))</td>
<td>31.1</td>
<td>25.5</td>
</tr>
<tr>
<td>with MAP update ((\Omega_r \otimes \Omega_c))</td>
<td>31.4</td>
<td>25.8</td>
</tr>
</tbody>
</table>

parameters to be updated is relatively small. But for systems with \( S = 40 \), which have a much larger number of parameters, we observed an increase in WER. MAP update, on the other hand, provided consistent reductions in WER.

For systems with \( S = 20 \) MAP update gave an additional 1% absolute WER reduction compared with ML update, in both training conditions. For their counterparts with \( S = 40 \), MAP update resulted in 1% absolute reduction in WER compared with the baseline, whereas the ML update increased the WER. This is consistent with our expectation that MAP can overcome the model overfitting encountered by ML. Again we used \( \ell_1 \)-norm regularized (sub-)state vector estimation [5] to improve numerical stability. However, we do not observe any improvement in WER by using full row and column covariance matrices compared to the identity matrices used in the priors. Setting just one of two covariance matrices to be the identity, did not result in a significant difference.

To obtain a better understanding of these results, we have plotted the eigenvalues of \( \Omega_r \) and \( \Omega_c \) for systems with \( S = 20 \) and \( S = 40 \) (Figure 7). This shows that the eigenvalues of column covariances decrease rapidly, and that the first few eigenvectors corresponding
Figure 7: Eigenvalues of row and column covariance matrices $\Omega_r, \Omega_c$ for both 20 and 40 dimensional subspace.

Figure 8: Effect of smoothing parameter $\tau$ in MAP adapted system on the log-likelihood in the training stage and WER in the testing stage.
to the top eigenvalues account for most of the variance. This was unexpected, as priors
with such a covariance structure will constrain the model to model subspace of lower
effective dimension, and limit its ability to learn from the data. In addition, the ad
hoc approach we have employed to approximate the covariances of the prior may not be
optimal. We cannot guarantee that the global covariance from the multilingual subspace
will work well for the target system. In future work, we shall investigate the estimation
of $\Omega_r$, $\Omega_c$ using the Bayesian approach of equation (50).

Finally, figure 8 shows the effect of the smoothing parameter $\tau$ for both training and
testing for a system with $S = 40$ and 5 hours of training data. Here, we only show the
MAP systems with row and column covariance matrices in the priors to be both identity
or full, denoted as “(I, I)” and “(R, C)”, respectively. When $\tau$ is small, the log-likelihood
is close to that of ML system, and as $\tau$ increases, the log-likelihood decreases accordingly,
but it is bounded by the baseline system which corresponds to $\tau \to \infty$. On the other
hand, by tuning the value of $\tau$, the WER of a MAP adapted system can be smaller than
both baseline and ML system. Other MAP adapted systems in Table 9 show a similar
trend. Note that the absolute value of optimal $\tau$ depends on the prior distribution and
also the amount of training data which means its range varies for different systems.

7.4 Experiments of Noise Adapation

We carried out the experiments on Aurora 4 dataset. The clean model was trained on
the WSJ 5k corpus, which has about 15 hours data. The testing set is categorised into
4 sub-sets, where A denote the clean set which comes from the same microphone with
training data; B is the noisy speech where various kinds of noise was added to A; C
denotes the clean speech that comes from a different microphone with the training data,
and D is various noise added on C. We used bigram language model for decoding.

7.4.1 Results of noise compensation on GMM based systems

Table 10 shows the results of VTS and JUD noise compensation on GMM based systems.
The clean speech model and multi-condition trained (MTR) model have about 3.1k
triphone states, and each speech state model is modelled by 16 Gaussians while the
silence state model uses 32 Gaussians. As expected, the clean speech model perform very
poor for noise corrupted testing data, while MTR model can alleviate the mismatch and
leads to significant improvement. We then carried out the noise compensation with both
VTS and JUD for the clean speech model. For JUD based experiments, we used 112
total regression classes, 48 for the silence models and 64 for speech models. The noise
model was initialised by first and last 20 frames for each utterance, which was used in
system denoted as “VTS-init” and “JUD-init” in the table. As we can see from Table 10,
the initialised noise model can lead to significant WER reduction for both VTS and JUD
based system. The hypothesis generated by the initial decoding were used to update
Table 10: Results of noise compensation by VTS and JUD on GMM systems.

<table>
<thead>
<tr>
<th>Methods</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean model</td>
<td>7.7</td>
<td>56.6</td>
<td>46.7</td>
<td>72.8</td>
<td>59.3</td>
</tr>
<tr>
<td>MTR model</td>
<td>12.7</td>
<td>18.6</td>
<td>31.7</td>
<td>36.8</td>
<td>26.9</td>
</tr>
<tr>
<td>VTS-init</td>
<td>8.7</td>
<td>22.4</td>
<td>43.0</td>
<td>48.0</td>
<td>33.9</td>
</tr>
<tr>
<td>+ 1st EM</td>
<td>7.1</td>
<td>15.8</td>
<td>17.3</td>
<td>28.6</td>
<td>20.8</td>
</tr>
<tr>
<td>+ 2nd EM</td>
<td>7.3</td>
<td>14.8</td>
<td>12.1</td>
<td>24.8</td>
<td>18.3</td>
</tr>
<tr>
<td>JUD-init</td>
<td>8.4</td>
<td>23.8</td>
<td>42.6</td>
<td>47.1</td>
<td>34.0</td>
</tr>
<tr>
<td>+ 1st EM</td>
<td>7.2</td>
<td>17.3</td>
<td>24.1</td>
<td>31.8</td>
<td>23.3</td>
</tr>
<tr>
<td>+ 2nd EM</td>
<td>7.0</td>
<td>16.6</td>
<td>16.3</td>
<td>28.7</td>
<td>21.1</td>
</tr>
</tbody>
</table>

Table 11: Results of noise compensation by VTS and JUD on SGMM systems with $\alpha = 0$.

<table>
<thead>
<tr>
<th>Methods</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean model</td>
<td>5.2</td>
<td>58.2</td>
<td>50.7</td>
<td>72.1</td>
<td>59.9</td>
</tr>
<tr>
<td>MTR model</td>
<td>6.8</td>
<td>15.2</td>
<td>18.6</td>
<td>32.3</td>
<td>22.2</td>
</tr>
<tr>
<td>VTS-init</td>
<td>6.3</td>
<td>40.3</td>
<td>38.9</td>
<td>61.7</td>
<td>46.9</td>
</tr>
<tr>
<td>+ 1st EM</td>
<td>6.4</td>
<td>32.7</td>
<td>26.3</td>
<td>47.8</td>
<td>36.8</td>
</tr>
<tr>
<td>+ 2nd EM</td>
<td>6.4</td>
<td>32.9</td>
<td>25.9</td>
<td>45.7</td>
<td>36.0</td>
</tr>
<tr>
<td>JUD-init</td>
<td>5.2</td>
<td>43.1</td>
<td>47.8</td>
<td>61.2</td>
<td>48.5</td>
</tr>
<tr>
<td>+ 1st EM</td>
<td>5.4</td>
<td>42.6</td>
<td>30.7</td>
<td>57.1</td>
<td>45.3</td>
</tr>
<tr>
<td>+ 2nd EM</td>
<td>5.6</td>
<td>42.5</td>
<td>30.0</td>
<td>56.8</td>
<td>45.1</td>
</tr>
</tbody>
</table>

From Table 10, updating the noise model leads to considerable performance gains for both VTS and JUD systems, and, in addition, VTS based systems consistently outperform their JUD counterparts just as our expectation, as JUD is an approximation of VTS. However, the computation cost for JUD is much lower than that of VTS.

7.4.2 Results of noise compensation on SGMM based systems

We then carried out the experiments to apply the VTS and JUD on SGMM based system. Firstly, the baseline results of clean and multi-condition trained model are given in Table 11. We observe that the SGMM system significantly outperforms the GMM one on clean test condition, however, it’s not the case for noisy testing data. It may demonstrate that though the model capacity of SGMM acoustic model is much larger, which achieve performance improvement in matched testing condition, it’s not able to handle mismatch well. On the other hand, the multi-condition trained SGMM system give much better results, and significantly outperform the GMM counterpart.

We then apply the VTS and JUD to the clean trained SGMM acoustic model, with
Table 12: Results of noise compensation by VTS and JUD on SGMM systems with $\alpha = 2.5$. 

<table>
<thead>
<tr>
<th>Methods</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTS-init</td>
<td>7.8</td>
<td>21.8</td>
<td>40.9</td>
<td>46.2</td>
<td>32.6</td>
</tr>
<tr>
<td>+ 1st EM</td>
<td>6.2</td>
<td>17.8</td>
<td>16.9</td>
<td>30.3</td>
<td>22.3</td>
</tr>
<tr>
<td>+ 2nd EM</td>
<td>6.3</td>
<td>18.0</td>
<td>12.7</td>
<td>29.4</td>
<td>21.7</td>
</tr>
<tr>
<td>JUD-init</td>
<td>10.7</td>
<td>15.2</td>
<td>40.9</td>
<td>42.5</td>
<td>29.9</td>
</tr>
<tr>
<td>+1st EM</td>
<td>5.7</td>
<td>23.0</td>
<td>20.7</td>
<td>38.0</td>
<td>28.0</td>
</tr>
<tr>
<td>+2nd EM</td>
<td>5.5</td>
<td>23.9</td>
<td>18.9</td>
<td>38.3</td>
<td>28.4</td>
</tr>
</tbody>
</table>

similar protocol as the GMM system. As stated before, we used the UBM model as the regression classes for the state models, and the noise model was updated based on the UBM. This may lead to a mismatch between the estimation and application of noise model, i.e., the objective function we used to update the noise model is to increase the likelihood of noise compensated UBM model given the testing utterance, and then the noise model is applied to the acoustic model. We can not guarantee that improving the likelihood of UBM model will lead to the improvement of likelihood of acoustic model. Yet, in the experiments, we do observe the likelihood of acoustic model increase while update the noise model. In Table 11, we show the results of such kind of system configuration. Here, the phase term in the mismatch function (58) is dropped, i.e. $\alpha = 0$. We observe that thought noise compensation lead to considerable improvement, the gains are much less than that in GMM based system. In addition, the first round of noise model update is effective but not the second round. This may show that objective function, i.e. estimating noise model based on UBM, is not a good option. The results also show that JUD based system is much worse than that of VTS. We observe that the value of JUD transforms $\{A^r, b^r, \Sigma_b^r\}$ is very large, as $J^r$ is very small.

We then introduce the non-zero phase term. Following Li et.al [37], we set all the coefficients of $\alpha$ be to 2.5, while the other system settings are the same with previous experiments. The results are given in Table 12. We can see that much better results are achieved for both VTS and JUD based systems. In [37], significant improvement is also observed by the phase term, and the authors attribute the improvement to that the phase term can alleviate the systematical bias, and also works as some kind of domain combination. Our observation is the non-zero $\alpha$ can smooth the coefficients of matrix $F$ in (63), results in much smoothed compensation parameters for both VTS and JUD. However, similar to the results in Table 11, JUD systems still considerable under-perform VTS counterparts, and the second round of noise model update does not lead to notable performance gains. This results confirms that clustering and noise model estimation based on the UBM model is not very efficient. Hence, in future work, we will investigate better approach for clustering, and shift the noise model estimation from UBM to SGMM acoustic model.
8 Plan for Future Work

SGMM acoustic modelling has shown considerable improvements compared to conventional one in several speech recognition tasks. As a not well explored research field, there are many interesting topics for further study. For our own work, we specially interest in the following topics:

- **Empirical Bayesian MAP adaptation.**
  In section 5, we have shown that MAP adaptation of model subspace with heuristic configuration of priors can lead to considerable performance gains for cross-lingual setting. In future, we would like to investigate estimating the priors by empirical Bayesian approach as equation (50) shows. This work is expected to be finished within the next 3 months.

- **Comparison of noise model estimation for noise robust SGMM, and**

- **Comparison of compensation techniques for noise robust SGMM.**
  Following our preliminary work in section 6, we will explore more effective approaches for noise model estimation and compensation. We have shown in the experiments that based on UBM model for clustering and noise model update, thought improvement was obtained, however, the results indicated that this is not an effective configuration. In future, we will investigate the noise model estimation based the SGMM acoustic model, and in addition, other clustering approaches such as by regression tree will also be explored. We expect to finished the work within the next 6-9 months.

After this work has been finished, we plan to summarise all the work and results into the Ph.D thesis. The writing of the thesis is expected to be started at the summer of 2012. If everything goes well and there are still time left, we are interested to explore one or two of the following topics:

- **Noise adaptive training of SGMM acoustic model.**

- **Joint speaker and noise adaptive training of SGMM acoustic model.**

- **Discriminative training with noise and speaker factors of SGMM acoustic model.**

9 Summary

In this report, we summarise our previous work in the field of Subspace Gaussian Mixture Model (SGMM) based acoustic modelling, which covers regularized model estimation for robustness, model training from out-domain data to improve estimation accuracy. We have shown by experimental results that these techniques can be successfully applied to cross-lingual speech recognition task. We also present the maximum a posteriori (MAP) adaptation of model subspace to alleviate the mismatch between out-domain trained
parameters and testing environment. In addition, we also discuss our preliminary work on noise robust SGMM acoustic model. We have shown how to apply conventional vector Taylor series (VTS) and joint uncertainty decoding (JUD) model based noise compensation techniques to SGMM acoustic model, with empirical setting of clustering. Finally, we point out the direction for our future work.

References


