Regularized Subspace Gaussian Mixture Models for Speech Recognition

Liang Lu*, Arnab Ghoshal, and Steve Renals

***Abstract*—Subspace Gaussian mixture models (SGMMs) provide a compact representation of the Gaussian parameters in an acoustic model, but may still suffer from over-fitting with insufficient training data. In this paper, the SGMM state parameters are estimated using a penalized maximum-likelihood objective, based on ℓ_1 and ℓ_2 regularization, as well as their combination, referred to as the elastic net, for robust model estimation. Experiments on the 5,000 word Wall Street Journal transcription task show word error rate reduction and improved model robustness with regularization.**

***Index Terms*—Subspace Gaussian mixture models, Regularization, ℓ_1/ℓ_2-norm penalty, sparsity, elastic net**

I. Introduction

ACOUSTIC modeling for large vocabulary speech recognition often needs to address the problem of robust model estimation from limited acoustic data. As such, there has recently been a renewed interest in regularization approaches to address the problem of data sparsity and model complexity. For instance, Sivaram et al. [1] introduced an approach to obtain sparse features from an auto-associative network using an ℓ_1-norm regularization function, and Sainath et al. [2], [3] combined ℓ_1 and ℓ_2 regularization to obtain a sparse exemplar-based representation for phoneme recognition. Regularised maximum likelihood linear regression (MLLR) for speaker adaptation is proposed in [4].

Conventional automatic speech recognition (ASR) systems use hidden Markov models (HMMs) whose emission densities are modeled by mixtures of Gaussians. The subspace Gaussian mixture model (SGMM) [5] is a recently proposed acoustic modeling approach for ASR, which has been demonstrated to outperform conventional systems while providing a more compact model representation [5], [6]. Similar to the joint factor analysis (JFA) model for speaker recognition [7], the SGMM uses a globally shared model subspace to capture the major model variations such that the Gaussian parameters in each HMM state are inferred subject to this subspace constraint, as opposed to the conventional approach of direct estimation. This leads to a significant decrease in the total number of parameters. Povey and coworkers [6] used maximum likelihood estimation (MLE) to train the state-specific and the global subspace parameters. While this approach works well in practice, it may still lead to overfitting with insufficient training data, despite the inherent compactness of the model.

In this paper, we investigate the regularized estimation of state-specific parameters in the SGMM acoustic model, by penalising the original maximum likelihood (ML) objective function using a regularization term. We investigate ℓ_1- and ℓ_2-norm regularization [8] in this context, as well as a combination of ℓ_1 and ℓ_2, sometimes referred to as the elastic net [9]. After giving a brief overview of the SGMM acoustic model in section II, we describe the use of regularization for SGMM state vector estimation in section III, and the optimization of such regularised objective functions in section IV. Finally, in section V, we present experiments on the 5,000 word Wall Street Journal (WSJ-5k) speech transcription task.

II. Overview of SGMM acoustic model

The basic form of the SGMM acoustic model can be expressed using the following equations [5]:

$$p(\mathbf{o}_t|j) = \sum_{c=1}^{C} w_{jc} \mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_{jc}, \boldsymbol{\Sigma}_c) \quad (1)$$

$$\boldsymbol{\mu}_{jc} = \mathbf{M}_c \mathbf{v}_j \quad (2)$$

$$w_{jc} = \frac{\exp \mathbf{w}_c^T \mathbf{v}_j}{\sum_{c'=1}^{C} \exp \mathbf{w}_{c'}^T \mathbf{v}_j} \quad (3)$$

where $\mathbf{o}_t \in \mathbb{R}^F$ denotes the t-th F-dimensional acoustic frame, $p(\mathbf{o}_t|j)$ is the emission density function for j-th HMM state modelled by a GMM with C Gaussians. $\mathbf{v}_j \in \mathbb{R}^S$ is referred to as the state vector, where S denotes the subspace dimension. The matrix $\mathbf{M} = (\mathbf{M}_1^T, \ldots, \mathbf{M}_C^T)^T$ of size $CF \times S$ (typically $S \ll CF$) spans the model subspace for Gaussian means and the $CF \times 1$ vector $\mathbf{w} = (\mathbf{w}_1^T, \ldots, \mathbf{w}_C^T)^T$ denotes the weight projection vector from which the mixture weights are derived. Together with $\mathbf{M}$ and $\mathbf{w}$, the covariance matrices $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_C\}$ are globally shared between all the HMM states.

The training of SGMM acoustic model can be decoupled into two main stages: the estimation of the globally-shared parameter set $\boldsymbol{\theta}_1 = \{\mathbf{M}, \mathbf{w}, \boldsymbol{\Sigma}\}$ and the estimation of the state-specific parameters $\boldsymbol{\theta}_2 = \{\mathbf{v}_1, \ldots, \mathbf{v}_J\}$ where J is the total number of states. Povey et al. [6] presented an EM algorithm to estimate these two interdependent parameter sets using an ML criterion.

Extensions to this basic model are also presented in [5], [6] which include sub-state splitting and speaker-specific subspaces. In this paper, we use sub-state splitting, where each

Liang Lu and Steve Renals are with University of Edinburgh, UK; email: {liang.lu,s.renals}@ed.ac.uk

Arnab Ghoshal is with Saarland University, Germany; email: aghoshal@lsv.uni-saarland.de

The research leading to these results was supported by the European Community's Seventh Framework Programme under grant agreement number 213850 (SCALE).

state j is represented by a mixture of state vectors $\mathbf{v}_{jm}$, such that

$$p(\mathbf{o}_t|j) = \sum_{m=1}^{M_j} a_{jmc} \sum_{c=1}^{C} w_{jmc} \mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_{jmc}, \boldsymbol{\Sigma}_c). \tag{4}$$

Although such a formulation significantly reduces the total number of parameters [6], 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 log-likelihood function of a particular (sub-)state vector $\mathbf{v}$ is approximated by a quadratic funciton which comes from the EM auxiliary function of state vectors as [6]:

$$\log p(\mathbf{O}|\mathbf{v}, \boldsymbol{\theta}_1) \simeq -\frac{1}{2}\mathbf{v}^T\mathbf{H}\mathbf{v} + \mathbf{b}^T\mathbf{v} + const, \tag{5}$$

where $\mathbf{O}$ denotes the set of all acoustic observations; $\mathbf{b}$ is a S-dimensional vector and $\mathbf{H}$ is a $S \times S$ matrix, representing the first- and second-order statistics respectively.[1] Although the state vectors are normally low-dimensional, the amount of data for computing the statistics $\mathbf{H}$ and $\mathbf{b}$ may still be insufficient. Some heuristic approaches may be applied, for instance $\mathbf{H}$ and $\mathbf{b}$ may be smoothed by the global statistics:

$$\hat{\mathbf{H}} = \mathbf{H} + \tau\mathbf{H}^{sm}, \qquad \hat{\mathbf{b}} = \mathbf{b} + \tau\mathbf{b}^{sm} \tag{6}$$

where $\mathbf{H}^{sm}$ and $\mathbf{b}^{sm}$ denotes the smoothing term calculated based on all the HMM states (see [10] for details), and $\tau \in \mathbb{R}$ is the tuning parameter. Povey et al. [6] also discuss some numeric controls to tackle the poor condition of $\mathbf{H}$. In this paper we address the problem using an explicit regularization function.

III. Regularized state vector estimation

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

$$\hat{\mathbf{v}} = \arg\max_{\mathbf{v}} \log p(\mathbf{O}|\mathbf{v}, \boldsymbol{\theta}_1) - J_{\boldsymbol{\lambda}}(\mathbf{v}). \tag{7}$$

$J_{\boldsymbol{\lambda}}(\mathbf{v})$ denotes the regularization function for $\mathbf{v}$ parametrised by $\boldsymbol{\lambda}$. We may interpret $-J_{\boldsymbol{\lambda}}(\mathbf{v})$ as a log-prior for the state vector, in which case we can interpret (7) as a MAP estimate. However, in this paper, 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 [11].

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

$$J_{\boldsymbol{\lambda}}(\mathbf{v}) = \lambda \sum_i |v_i|^q \quad s.t. \quad \lambda \geq 0. \tag{8}$$

The case $q = 1$ corresponds to ℓ_1-norm regularization, sometimes referred to as the lasso [12], and the case $q = 2$ corresponds to ℓ_2-norm regularization, which is referred to as ridge regression [8] or weight decay.

Both ℓ_1- and ℓ_2-norm penalties perform an element-wise shrinkage of $\mathbf{v}$ towards zero in the absence of an opposing data-driven force [8], which enables more robust estimation. The ℓ_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 [1], [2]. It is possible to seek a compromise between the ℓ_1 and ℓ_2 penalties by simply setting $1 < q < 2$ which is sometimes referred to as a bridge penalty. However, the non-linearity of the bridge penalty brings increased computational complexity. Alternatively, the ℓ_1- and ℓ_2-norm penalties can both be applied, as in elastic net regularization [9]:

$$\begin{aligned} J_{\boldsymbol{\lambda}}(\mathbf{v}) &= \lambda_1 \sum_i |v_i| + \lambda_2 \sum_i |v_i|^2, \\ s.t. &\quad \lambda_1, \lambda_2 \geq 0. \end{aligned} \tag{9}$$

This is much less computationally demanding than the bridge penalty. In this paper, we investigate the ℓ_1-norm, ℓ_2-norm and elastic net regularization for the estimation of SGMM state vectors.

IV. Optimization

Given the regularized objective function for state vector estimation (7), a closed form solution is readily available for the ℓ_2-norm penalty:

$$\begin{aligned} \hat{\mathbf{v}} &= \arg\max_{\mathbf{v}} -\frac{1}{2}\mathbf{v}^T\mathbf{H}\mathbf{v} + \mathbf{b}^T\mathbf{v} - \lambda\|\mathbf{v}\|_{\ell_2} \\ &= (\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{b} \end{aligned}$$

However, there is no such closed form solutions for the ℓ_1-norm and elastic net penalties as their objective functions are not differentiable. In both the optimization and signal processing fields, there have been numerous approaches proposed to solve the ℓ_1-norm penalty problem and here we adopt the gradient projection algorithm of Figueiredo et al. [13]. The same approach may be applied to the elastic net penalty as it can be formulated in terms of the ℓ_1 penalty:

$$\hat{\mathbf{v}} = \arg\max_{\mathbf{v}} -\frac{1}{2}\mathbf{v}^T(\mathbf{H} + \lambda_2\mathbf{I})\mathbf{v} + \mathbf{b}^T\mathbf{v} - \lambda_1\|\mathbf{v}\|_{\ell_1}, \tag{10}$$

given the regularization parameters λ_1 and λ_2. A proper scaling factor should applied to the result of (10) to get the exact elastic net solution, but we did not do it in this work which corresponds to the naive elastic net [9].

Expressing (7) with the ℓ_1 penalty results in the following objective function:

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}} \frac{1}{2}\mathbf{v}^T\mathbf{H}\mathbf{v} - \mathbf{b}^T\mathbf{v} + \lambda\|\mathbf{v}\|_{\ell_1}, \quad \lambda > 0. \tag{11}$$

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

$$\mathbf{v} = \mathbf{x} - \mathbf{y}, \quad \mathbf{x} \geq 0, \mathbf{y} \geq 0, \tag{12}$$

where, $\mathbf{x} = [\mathbf{v}]_+$ which takes the positive entries of $\mathbf{v}$ while keeping the rest as 0, i.e. $x_i = \max\{0, v_i\}$ for all $i = 1, \ldots, S$.

[1] If the state is split, (5) should be the objective function of sub-state vectors—regularization is employed at the sub-state level in this work.

Similarly, $\mathbf{y} = [-\mathbf{v}]_+$. In this case, equation (11) can be rewritten as

$$\begin{aligned}(\hat{\mathbf{x}},\hat{\mathbf{y}}) = \arg\min_{\mathbf{x},\mathbf{y}} & \frac{1}{2}(\mathbf{x}-\mathbf{y})^T\mathbf{H}(\mathbf{x}-\mathbf{y}) \\ & - \mathbf{b}^T(\mathbf{x}-\mathbf{y}) + \lambda\mathbf{1}_S^T\mathbf{x} + \lambda\mathbf{1}_S^T\mathbf{y} \\ s.t. \quad & \mathbf{x} \geq 0, \mathbf{y} \geq 0\end{aligned} \tag{13}$$

where $\mathbf{1}_S$ denotes an S-dimensional vector whose elements are all 1. We can reformulate (13) further as a more standard bound-constraint quadratic program

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} \frac{1}{2}\mathbf{z}^T\mathbf{B}\mathbf{z} + \mathbf{c}^T\mathbf{z} \quad s.t. \quad \mathbf{z} \geq 0 \tag{14}$$

where we have set

$$\mathbf{z} = \left[\begin{array}{c}\mathbf{x}\\ \mathbf{y}\end{array}\right], \mathbf{c} = \lambda\mathbf{1}_{2S} + \left[\begin{array}{c}-\mathbf{b}\\ \mathbf{b}\end{array}\right], \text{and } \mathbf{B} = \left[\begin{array}{cc}\mathbf{H} & -\mathbf{H}\\ -\mathbf{H} & \mathbf{H}\end{array}\right].$$

The objective function (14) does not suffer the nonlinearity problem of the original objective function (11), and its gradient is readily available as

$$\mathcal{G}(\mathbf{z}) = \mathbf{B}\mathbf{z} + \mathbf{c}, \tag{15}$$

which forms the basis of the gradient projection algorithm (see [13] for details).

The regularization parameters in (8) or (9) 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 $\mathbf{v}_j$ and subspace parameters $\mathbf{M}, \mathbf{w}$ are interdependent. In principle, the shrinkage of state vectors by regurlarization may be undone by a corresponding scaling of $\mathbf{M}$ or $\mathbf{w}$. This can be addressed by the renormalizing the phonetic subspaces, as described in [10] [Appendix K], such that the state vectors $\mathbf{v}_j$ always have unit variance after each iteration.

V. Experiments

We use the WSJ-5k data for our speech transcription experiments. We follow the setup described in [14]. 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+$\Delta + \Delta\Delta$ 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).

A. Baseline System

We first train a conventional HMM-GMM baseline recognizer using the HTK speech recognition toolkit [15]. The baseline system has 3093 tied cross-word triphone states, each with a 16-component GMM with diagonal covariance. Our baseline result of of 10.3% WER on the test set is comparable to the 10.48% WER reported in [14] 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 [6], 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.

TABLE I
Word error rates of SGMM acoustic model with ad-hoc smoothing or renormalization, $S = 40$

GMM baseline: 10.3					
SGMM with ad-hoc smoothing or renormalization					
#Substates	$R(\mathbf{v})$	$\tau = 0$	$\tau = 5$	$\tau = 10$	$\tau = 20$
3k	9.7	9.8	9.9	10.0	10.1
4.5k	9.7	9.6	9.7	9.7	9.8
6k	9.7	9.4	9.4	9.5	9.6
9k	9.2	9.1	9.2	9.2	9.2
12k	9.0	8.8	8.9	9.1	9.1
16k	8.8	**8.6**	8.8	**8.9**	**8.6**
20k	8.8	8.7	8.7	9.3	8.9
24k	**8.3**	8.8	8.6	9.1	8.8
28k	8.5	8.7	8.7	9.1	8.8
32k	8.7	9.0	**8.5**	9.4	9.7

B. SGMM results with smoothing and renormalization

We first compare the performance of ad-hoc smoothing shown in equation (6). The results are given in Table I for different values of the smoothing parameter τ. We also present the results by renormalization denoted as $R(\mathbf{v})$ in Table I. 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.

C. 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 [6].

Table II shows the results of regularization with ℓ_1, ℓ_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 ℓ_1, ℓ_2 and elastic net systems (i.e. $\lambda_1 = \lambda_2 = 10$ in equation 9). 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

TABLE II
COMPARISON OF SGMM ACOUSTIC MODEL WITH REGULARIZED (SUB-)STATE VECTOR ESTIMATION, $S = 40$

#Sub	without renormalization				with renormalization			
-states	-	ℓ_1	ℓ_2	$eNet$	-	ℓ_1	ℓ_2	$eNet$
3k	9.8	9.7	9.9	9.9	9.7	10.2	9.7	9.9
4.5k	9.6	9.4	9.7	9.6	9.7	9.8	9.7	9.9
6k	9.4	9.4	9.4	9.4	9.7	9.7	9.4	9.6
9k	9.1	9.1	9.1	9.3	9.2	9.2	9.2	9.5
12k	8.8	9.0	8.8	8.9	9.0	8.8	9.1	9.5
16k	**8.6**	8.8	**8.4**	8.7	8.8	8.9	8.9	9.1
20k	8.7	**8.3**	8.8	8.6	8.8	8.7	**8.4**	9.2
24k	8.8	8.4	8.7	**8.5**	**8.3**	8.5	8.6	**9.0**
28k	8.7	8.4	8.5	8.5	8.5	8.4	8.7	9.2
32k	9.0	8.5	8.5	8.8	8.7	**8.3**	9.0	9.2

TABLE III
RESULTS OF SGMM SYSTEM WITH ℓ_1-NORM REGULARIZATION, $S = 60$

#Sub-states	3k	4.5k	6k	9k	12k	16k	20k
Baseline	9.6	9.5	**9.1**	9.3	9.2	9.2	9.3
ℓ_1-norm	9.6	9.2	9.0	9.0	9.0	8.9	**8.9**

argument of regularization in this paper. We do not observe significant difference between ℓ_1 and ℓ_2-norm penalty in terms of performance, and elastic net do not give further gains. In our experiments, ℓ_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 II. 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 $\mathbf{M}$ and $\mathbf{w}$ can ameliorate over-penalization of the state-vectors to an extant.

D. Extensions

In this paper, we focused on the regularized estimation of the state-dependent parameters. However, this approach can be extended to the estimation of the global shared parameters, i.e. $\mathbf{M}$, $\mathbf{w}$ and $\mathbf{\Sigma}$, which we will explore in future works. As in our experiments, we observe that except for the state vectors, these state independent parameters may also suffer from the data sparsity problem which limits the model capacity, especially for higher dimensional subspaces.

Table III shows the results of SGMM model with ℓ_1-norm regularization (without renormalization), in which the dimension of state vector is increased to 60. Compared to the 40-dimensional subspace SGMM systems in Table II, we do not achieve any improvement but notable degradation for both baseline and ℓ_1 regularized systems, which is partly due to the poor estimation of the globally shared parameters. Based on the approach presented in this paper, extending the regularized estimation to the state independent parameters is not difficult, as we can reformulate the objective functions of these parameters into their quadratic forms, by which the code used for state vector regularization can be shared.

VI. CONCLUSION

In this paper, we have investigated regularized state model estimation for the subspace GMM acoustic model. Given the original ML based objective function, we added regularization penalties based on the ℓ_1-norm and the ℓ_2-norm, as well as their combined form, the elastic net. From our experimental results on WSJ-5k speech transcription task, we have observed reductions in word error rate and improved model robustness by all the three types of regularization. While the performance gains are found to be mostly due to improved numerical stability of the updates, which can also be achieved by renormalizing the phonetic subspaces, regularization is shown to prevent overfitting with larger models. This may prove helpful in training acoustic models with lesser resources. In future, we plan to study the effect of regularization on the global subspace parameters, as well as in a low resource setting.

REFERENCES

[1] G. Sivaram, S. Nemala, M. Elhilali, T. Tran, and H. Hermansky, “Sparse coding for speech recognition,” in *Proc. ICASSP*, 2010, pp. 4346–4349.

[2] T. Sainath, A. Carmi, D. Kanevsky, and B. Ramabhadran, “Bayesian compressive sensing for phonetic classification,” in *Proc. ICASSP*, 2010, pp. 4370–4373.

[3] T. Sainath, B. Ramabhadran, D. Nahamoo, D. Kanevsky, and A. Sethy, “Sparse representation features for speech recognition,” in *Proc. INTERSPEECH*, 2010, pp. 2254–2257.

[4] M. K. Omar, “Regularized feature-based maximum likelihood linear regression for speech recognition,” in *Proc. INTERSPEECH*, 2007, pp. 1561–1564.

[5] D. Povey, L. Burget *et al.*, “Subspace Gaussian mixture models for speech recognition,” in *Proc. ICASSP*, 2010, pp. 4330–4333.

[6] ——, “Subspace Gaussian mixture models–A structured model for speech recognition,” *Computer Speech & Language*, vol. 25, no. 2, pp. 404–439, April 2011.

[7] P. Kenny, “Joint factor analysis of speaker and session variability: theory and algorithms,” CRIM-06/08-13, Tech. Rep., 2005.

[8] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*. Springer, 2005.

[9] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[10] D. Povey, “A tutorial-style introduction to subspace Gaussian mixture models for speech recognition,” MSR-TR-2009-111, Microsoft Research, Tech. Rep., 2009.

[11] X. Zhao, Y. Dong, J. Zhao, L. Lu, J. Liu, and H. Wang, “Variational Bayesian joint factor analysis for speaker verification,” in *Proc. ICASSP*, 2009, pp. 4049–4052.

[12] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[13] M. Figueiredo, R. Nowak, and S. Wright, “Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems,” *IEEE Journal on selected topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.

[14] P. Woodland, J. Odell, V. Valtchev, and S. Young, “Large vocabulary continuous speech recognition using HTK,” in *Proc. ICASSP*, 1994, pp. 125–128.

[15] S. Young *et al.*, *The HTK Book*. Cambridge University Engineering Department, 2002.