BASIS SUPERPOSITION PRECISION MATRIX MODELS FOR LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

K.C. Sim and M.J.F. Gales

Engineering Department, Cambridge University, Trumpington St., Cambridge, CB2 1PZ U.K.
Email: {kcs23,mjfg}@eng.cam.ac.uk

ABSTRACT

An important aspect using Gaussian mixture models in a HMM-based speech recognition systems is the form of the covariance matrix. One succesfull approach has been to model the inverse covariance, precision, matrix by superimposing multiple basis. This paper presents a general framework of basis superposition. Models are described in terms of parameter tying of the basis coefficients and restrictions in the number of basis. Two forms of parameter tying are described which provide compact model structure. The first constrains the basis co-efficients over multple basis vectors (or matrices). This is related to the subspace for precision and mean (SPAM) model. The second constrains the basis coefficients over multiple components, yielding as one example heteroscedastic LDA (HLDA). Both maximum likelihood and minimum phone error training of these models is discussed. The performance of various configurations is examined on a conversational telephone speech task, SwitchBoard.

1. INTRODUCTION

Gaussian Mixture Models are commonly used as the state probability density function for HMM-based LVCSR. They may be expressed as

$$p(\boldsymbol{o}_t|s) = \sum_{m=1}^{M} c^{(m)} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)}) \tag{1}$$

where $\boldsymbol{o}_t$ is a d-dimensional observation vector, $c^{(m)}$ is the mixture weight for component m, $\boldsymbol{\mu}^{(m)}$ and $\boldsymbol{\Sigma}^{(m)}$ are d-dimensional mean vector and $d \times d$ symmetric covariance matrix respectively and s denotes the HMM state. For a system with large number of Gaussian components and high dimensionality, the number of parameters would be domainated by those associated with the covariance matrix, $\mathcal{O}(d^2)$. This has lead to the use of diagonal covariance matrices in most large vocabulary systsms. However, structured covariance matrix approximations offer an alternative compact and efficient approach. Multiple forms of approximation have been examined. State-space models, for example the factor-analysed HMM [1], provide one option for compact covariance matrix modelling. Recently there has been interest in modelling the inverse covariance matrux, *precision matrix*, as this can be highly efficient during decoding. Schemes in this category include semi-tied covariances STC [2] (or maximum likelihood linear transforms MLLT [3]), extended MLLT [4] and subspace for precision and mean (SPAM) models [5]. It is also possible to describe heteroscedastic LDA [6] within this class.

This work was supported by DARPA grant MDA972-02-1-0013. The paper does not necessarily reflect the position or the policy of the US Government and no official endorsement should be inferred.

In this paper, a general framework for basis superposition for precision matrix modelling is described. The precision matrix is modelled as a linear interpolation of a set of globally shared symmetric matrices (known as *basis matrices* or the associated *basis vectors*). The interpolation weights (known as *basis coefficients*) are usually Gaussian specific (STC, EMLLT and SPAM). The nature of the model is determined by the form of the tying of the basis co-efficients. They may be tied over multple basis matrices, resulting in a SPAM-like model, or over multiple components, one form of which is the HLDA precision matrix. This paper discuss the general form of models models and how to train them using both maximum likelihood and minimum phone error (MPE) criteria.

The rest of this paper is organised as follows. In Section 2, the concept of basis superposition is introduced as the generic framework for precision matrix modelling. Section 3 and 4 then describe the ML and MPE estimation process respectively for the model parameters. Experimental results on a conversational telephone speech task are given in Section 5.

2. PRECISION MATRIX MODELLING

When using Gaussian distributions it is convenient to express the log likelihood function in terms of the precision matrix

$$\begin{aligned} \log(p(\boldsymbol{o}_t|\boldsymbol{\theta})) \quad &= \quad \frac{1}{2}\log(|\boldsymbol{P}^{(m)}|) - \frac{d}{2}\log(2\pi) \\ &\quad -\frac{1}{2}(\boldsymbol{o}_t - \boldsymbol{\mu}^{(m)})'\boldsymbol{P}^{(m)}(\boldsymbol{o}_t - \boldsymbol{\mu}^{(m)}) \end{aligned} \tag{2}$$

where $\boldsymbol{P}^{(m)} = \boldsymbol{\Sigma}^{(m)-1}$ is the precision matrix and $\boldsymbol{\theta}$ denotes the vector of model parameters. A general form of precision matrix can be expressed as a linear superposition of rank-1 basis matrices,

$$\boldsymbol{P}^{(m)} = \boldsymbol{A}'\boldsymbol{\Lambda}^{(m)}\boldsymbol{A} = \sum_{i=1}^{n} \lambda_{ii}^{(m)} \boldsymbol{a}_i' \boldsymbol{a}_i \tag{3}$$

where $\boldsymbol{a}_i$ (basis vectors) denotes the i^{th} row of a $n \times d$ matrix $\boldsymbol{A}$ and $\boldsymbol{a}_i'\boldsymbol{a}_i$ forms a rank-1 symmetric matrix. If the basis coefficients are component specific, $\lambda_{ii}^{(m)}$, the precision matrix in equation 3 becomes a STC model when $n = d$ and an EMLLT model when $d < n \leq \frac{d}{2}(d+1)$. The use of EMLLT results in an increase in the number of model parameters. Two forms of parameter tying are discussed that allow a significant compression in the number of model parameters without losing much of its modelling capability.

2.1. Tying of basis coefficients over basis vectors

The basis coefficients $\lambda_{ii}^{(m)}$ can be shared by a set of basis vectors $\boldsymbol{a}_{ir}$, where R_i denotes the number of basis vectors sharing the same coefficient, $\lambda_{ii}^{(m)}$. Equation 3 may then be expressed as

$$\boldsymbol{P}^{(m)} = \sum_{i=1}^{n} \lambda_{ii}^{(m)} \sum_{r=1}^{R_i} \boldsymbol{a}_{ir}' \boldsymbol{a}_{ir} = \sum_{i=1}^{n} \lambda_{ii}^{(m)} \boldsymbol{S}_i \tag{4}$$

where $\boldsymbol{S}_i = \sum_{r=1}^{R_i} \boldsymbol{a}_{ir}' \boldsymbol{a}_{ir}$ is a symmetric matrix of rank R_i. This form of precision matrix model is an example of SPAM model with unconstrained mean [5].

2.2. Tying of basis coefficients over Gaussian components

The basis coefficients can also be tied over several Gaussian components. This generalises the precision matrix expression to[1]

$$\boldsymbol{P}^{(m)} = \sum_{i=1}^{k} \lambda_{ii}^{(m)} \boldsymbol{a}_i' \boldsymbol{a}_i + \sum_{i=k+1}^{n} \lambda_{ii}^{(g_i(m))} \boldsymbol{a}_i' \boldsymbol{a}_i \tag{5}$$

where $g_i(m)$ is the shared basis coefficient for component m and basis vector i. If all the Gaussian components are gathered into one global group and $n = d$, this becomes the standard HLDA scheme

$$\boldsymbol{P}^{(m)} = \sum_{i=1}^{k} \lambda_{ii}^{(m)} \boldsymbol{a}_i' \boldsymbol{a}_i + \sum_{i=k+1}^{n} \lambda_{ii} \boldsymbol{a}_i' \boldsymbol{a}_i \tag{6}$$

where $\lambda_{ii}^{(m)} = 1/\sigma_{ii}^{(m)2}$ and $\lambda_{ii} = 1/\sigma_{ii}^2$. $\sigma_{ii}^{(m)2}$ and σ_{ii}^2 are the variances of the HLDA useful and nuisance dimensions respectively. However, the mean vectors of the conventional HLDA models are confined within the HLDA subspace. Where the mean is not constrained to lie in the useful sub-space this will be referred to as an HLDA precision matrix model (HLDA-PMM).

A simplified version of this basis coefficient tying may be implemented with the HTK parameter tying scheme [7]. If all the basis coefficients of multiple components are tied in the same fashion, then it may be viewed as a *tied-covariance* scheme. Though this restricts the possible tying approaches, the estimation of the model parameters is simplified, the standard EMLLT updates may be used. Furthermore it is only necessary to accumulate the tied covariance matrix statistics to estimate the basis vectors, which can reduce the memory requirements during training.

3. MAXIMUM LIKELIHOOD TRAINING

The ML estimation of the model parameters makes use of the expectation maximisation (EM) algorithm, in the same fashion as standard HMM parameter training. The auxiliary function for these models may be expressed as

$$\begin{aligned} \mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) \;\; = \;\; & K + \frac{1}{2} \sum_{m=1}^{M} \beta^{(m)} \Big\{ \log \left(|\boldsymbol{P}^{(m)}| \right) \\ & - \sum_{i=1}^{n} \lambda_{ii}^{(m)} \sum_{r=1}^{R_i} \boldsymbol{a}_i \boldsymbol{W}^{(m)} \boldsymbol{a}_i' \Big\} \end{aligned} \tag{7}$$

[1] A more general expression is possible where the basis coefficient tying over multiple basis vectors is incorporated into the scheme.

where K is a constant subsuming the terms independent of $\boldsymbol{\theta}$ and

$$\boldsymbol{W}^{(m)} = \frac{\sum_{t=1}^{T} \gamma_m(t)(\boldsymbol{o}_t - \boldsymbol{\mu}^{(m)})(\boldsymbol{o}_t - \boldsymbol{\mu}^{(m)})'}{\sum_{t=1}^{T} \gamma_m(t)} \tag{8}$$

$$\beta^{(m)} = \sum_{t=1}^{T} \gamma_m(t) \tag{9}$$

are the required statistics. $\gamma_m(t)$ is the probability of component m at time t using the current parameters $\hat{\boldsymbol{\theta}}$. In this general basis superposition framework, the mean vectors are not constrained, though in the same same fashion as the SPAM or HLDA models they mey be tied. The precision matrix parameters are updated in an iterative fashion, alternating between the updates of basis vectors and coefficients.

There is no closed form solution for the update of the basis vectors. A row by row second order gradient optimisation method is employed using the gradient vector and the Hessian matrix evaluated at the current estimate. This requires full variance statistics, $\boldsymbol{W}^{(m)}$, to be accumulated, making the basis vector update both computational expensive and memory intensive. Furthermore, it is also highly sensitive to the initial values of the basis vectors. Thus, a good starting point is important to ensure fast convergence.

Component specific coefficients, which may be shared over multiple basis vectors, can be updated using an iterative closed form solution for each $\lambda_{ii}^{(m)}$ as follows

$$\hat{\lambda}_{ii}^{(m)} = \lambda_{ii}^{(m)} + \Delta_{ii}^{(m)} \tag{10}$$

$$\Delta_{ii}^{(m)} = \sum_{r=1}^{R_i} \left(\boldsymbol{a}_{ir} \boldsymbol{W}^{(m)} \boldsymbol{a}_{ir}' \right)^{-1} - \sum_{r=1}^{R_i} \left(\boldsymbol{a}_{ir} \boldsymbol{\Sigma}^{(m)} \boldsymbol{a}_{ir}' \right)^{-1} \tag{11}$$

where R_i again specifies the basis vectors sharing the same coefficient. When $R_i = 1$, Eqn (11) simplifies to the EMLLT additive[2] update given by [4]. However, if the basis coefficients are tied over different Gaussian components, no closed form solution exists. A gradient optimisation scheme is employed. This yields the update formula of

$$\Delta_{ii}^{(m)} = -\eta \left\{ \frac{\sum_{m \in M} \beta^{(m)} \left\{ \boldsymbol{a}_i \boldsymbol{\Sigma}^{(m)} \boldsymbol{a}_i' - \boldsymbol{a}_i \boldsymbol{W}^{(m)} \boldsymbol{a}_i' \right\}}{\sum_{m \in \mathcal{M}} \beta^{(m)} \left(\boldsymbol{a}_i \boldsymbol{\Sigma}^{(m)} \boldsymbol{a}_i' \right)^2} \right\} \tag{12}$$

where $\mathcal{M}$ denotes the set of Gaussian components sharing the same basis coefficient and η is the gradient optimisation step size. As it is possible to compute the auxiliary function value given the sufficient statistics, it is possible to reduce the value of η until the auxiliary function for the component increases.

As previously mentioned, the update of basis vectors is memory intensive. If a *good* set of initial basis vectors is found (usually by stacking several sets of STC basis vectors together), only the basis coefficients need to be updated. The required statistics reduces to $\boldsymbol{a}_i \boldsymbol{W}^{(m)} \boldsymbol{a}_i'$ for $1 \leq i \leq n$. This significantly reduces the amount of compute and memory usage.

In LVCSR systems, variance floor is commonly used to set a lower bound to the variance elements. Unlike the STC and HLDA models, conventional variance floor techniques are not directly applicable to basis superposition precision matrix models when the

[2] There is an alternative update rule presented in [4] called the multiplicative update formula. This update restricts the basis coefficients to take only positive values.

number of basis vectors exceeds the feature dimensionality. If the variance is applied to the resultant precision matrix, then the computationally efficient calculation of the log-likelihoods [4] is not possible. In this work the flooring is directly applied to the individual full variance statistics, $\boldsymbol{W}^{(m)}$. The flooring was applied to the elements on the leading diagonal as used in HTK [7].

4. MPE ESTIMATION OF EMLLT PARAMETERS

Discriminative training has been found to provide significant gain in performance over the conventional ML training for LVCSR [8]. Discriminative training has been successfully applied on SPAM models using the Maximum Mutual Information (MMI) training criterion [5]. In this section, discriminative training of EMLLT models will be presented using the MPE criterion. The approach used is applicable to all the tying schemes described.

The objective function for MPE training is given by [8]

$$\mathcal{F}_{\text{MPE}}(\boldsymbol{\theta}) = \sum_{r=1}^{R} \frac{\sum_{s_n=1}^{s_N} p_\theta(\mathcal{O}_r|s_n)^\kappa P(s_n)\text{RPA}(s_n, s_r)}{\sum_{s_d=1}^{s_D} p_\theta(\mathcal{O}_r|s_d)^\kappa P(s_d)} \quad (13)$$

where s_r is the correct transcription for the r^{th} speech data $\mathcal{O}_r$. $p_\theta(\mathcal{O}|s)$ denotes the likelihood probability of the speech data $\mathcal{O}$ given the transcription s and $\boldsymbol{\theta}$. $P(s)$ is the language model probability for sentence s. RPA(s_n, s_r) denotes the raw phone accuracy of the sentence s_n given the correct sentence s_r. κ is a scaling factor and can be adjusted to improve test-set performance. This objective function can be maximised using the following weak-sense auxiliary function [8]

$$\begin{aligned} \mathcal{Q}^{mpe}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) \quad &= \quad \mathcal{Q}^{num}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) - \mathcal{Q}^{den}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) + \\ & \qquad \mathcal{Q}^{sm}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) + \mathcal{Q}^{ml}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) \end{aligned} \quad (14)$$

The numerator and denominator auxiliary functions, $\mathcal{Q}^{num}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$ and $\mathcal{Q}^{den}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$, have the same form as equation 7. However, in each case the form of the "posterior", $\gamma_m(t)$ is altered in the same fashion as standard MPE training. For more details of this see [9]. The smoothing term, $\mathcal{Q}^{sm}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$, satisfies the constraint that it is a maximum at the current parameters value, $\hat{\boldsymbol{\theta}}$. This may be written in the same form as equation 7 with $\beta_{sm}^{(m)} = D_m$ and $\boldsymbol{W}_{sm}^{(m)} = \frac{\boldsymbol{\Sigma}^{(m)}}{D_m}$. The smoothing constant, D_m, is adjusted to ensure stability of the MPE estimation. The final term in equation 14 corresponds to a prior based around the ML statistics. This is I-smoothing [9]. Again the form of equation 7 is used with $\beta_{ml}^{(m)} = \tau$ and $\boldsymbol{W}_{ml}^{(m)}$ is the full variance ML statistics. The I-smoothing constant τ is determined empirically [9]. The overall statistics required by equation 14 can be expressed in terms of the individual statistics, so that

$$\begin{aligned} \boldsymbol{W}_{mpe}^{(m)} \quad &= \quad \frac{1}{\beta_{mpe}^{(m)}} \Big\{ \beta_{num}^{(m)} \boldsymbol{W}_{num}^{(m)} - \beta_{den}^{(m)} \boldsymbol{W}_{den}^{(m)} + \\ & \qquad \boldsymbol{\Sigma}^{(m)} + \tau \boldsymbol{W}_{ml}^{(m)} \Big\} \end{aligned} \quad (15)$$

$$\beta_{mpe}^{(m)} \quad = \quad \beta_{den}^{(m)} - \beta_{den}^{(m)} + D_m + \tau \quad (16)$$

Thus, the ML update formulae described in Section 3 can be used directly for MPE training by replacing the ML statistics with those in equation 16 and 15.

The smoothing constant, D_m, is set in the same fashion as [8],

$$D_m = \arg\max_i \left\{ \max \left(E\beta_{den}^{(m)}, 2D_i \right) \right\} \quad (17)$$

where D_i is the value of smoothing constant that ensures positive variance in the i^{th} dimension and E is a configurable constant. However, to ensure stability of estimation and positive-definiteness of a non-diagonal covariance matrix, a larger value of D_m may be required. Unfortunately, this slows down the convergence of the mean estimation. Here, a modified approach is propiosed where different smoothing constant values are used for the updates of the mean vectors and parameters involving the precision matrix structure. For mean estimation, equation 17 is used to determine D_m. This value is then used as the initial value of D_m for calculating the required statistics given by equation 15 and 16. This value of D_m is then gradually increased until $\boldsymbol{W}_{mpe}^{(m)}$ becomes positive definite.

5. EXPERIMENTAL RESULTS

Systems were trained using the 296 hours `h5etrain03` Switchboard English acoustic training set. 12 PLP coefficients were extracted, including the `C0` term, and the first, second and third derivatives were appended to form a 52-dimensional feature vector. Side-based Cepstral Mean Normalisation (CMN), Cepstral Variance Normalisation (CVN) and Vocal Tract Length Normalisation (VTLN) were also used. Either HLDA, STC, or EMLLT was then applied to this feature vector. All models were gender independent, using decision tree state-clustered triphone models with 6192 distinct states. The experimental results presented in this paper are based on the 2.96 hours `dev01sub` evaluation test-set of the Switchboard English task using a 58k-word trigram language model. The baseline system was a 16-component HLDA diagonal covariance system.

Since both EMLLT and HLDA may be viewed as precision matrix modelling techniques, initial experiments investigated the interaction of these two approaches. Table 1 shows the perfor-

System	Matrix Dimensions	WER (%)
HLDA	39 x 52	33.5
HLDA+EMLLT	78 x 39 + 39 x 52	33.1
(STC)	52 x 52	33.3
EMLLT	78 x 52	32.6
	91 x 52	32.7

Table 1. Comparisons of 16-component precision matrix models on the `dev01sub` test set

mance of various forms HLDA, EMLLT and STC systems. The baseline HLDA system used a 13-dimensional nuisance space. The feature vector was thus projected fomr 52 to 39 dimensions. Building an EMLLT system (n=78) on the 39-dimensional projected space (HLDA+EMLLT), reduced the error rate by about 0.4% absolute. Rather than building systems in the projected space, they may be built in the original 52 dimensional. Note in this space the mean vector will also be 52 dimensional. Using a STC system the error rate was slightly better than the HLDA system, but worse than the HLDA+EMLLT system, despite having the extrended mean vectors. Two EMLLT systems were then built. One using $n = 78$, the other $n = 91$. Both systems showed significant gains the STC system and the HLDA+EMLLT system.

System	Update Parameters	Average Log Likelihood	WER (%)
HLDA-PMM	Basis	-54.51	33.2
EHLDA-PMM	Coefficients	-54.22	32.9
EHLDA-PMM	Basis Vectors & Coefficients	-53.97	32.7

Table 2. Comparisons of 16-component HLDA precision matrix models on the `dev01sub` test set

In the previous experiments HLDA was run as a projection scheme rather than as a tied precision matrix model. Table 2 shows the performance of the HLDA-PMM model. Initially the basis vectors were fixed and only the means and basis co-efficients updated. By using mean vectors in the full 52 dimensional space the error rate was reduced by 0.3% absolute over the HLDA system. Note the model for the 39 nuisance dimensions remained unaltered. An extended version of this model,was built, where the HLDA basis were appended by a set of 39 (the static, delta and delta-delta) identity vectors. This EHLDA-PMM system was then trained again only updating the basis co-efficients. This further decreased the error rate by 0.3%. For the EHLDA-PMM systems the basis vectors were then also updated. This gave a further 0.2% gain in performance. However this EHLDA+PMM gave about the same perfoamnce as the EMLLT system in table 1. For this configuration the additional HLDA basis and global basis co-efficients gave no gain in performance. It is hoped that alternative more flexible basis tying will yield greater gains.

The systems presented so far have been simpler than those typically trained on this data. MPE training is normally used and there are typically 28 component per state rather than 16. It was not practical to gather statistics for the full 28 component system. Hence for the 28 component EMLLT experiments a *tied-covariance* system was built. The standard HLDA system was used as the start point For the EMLLT system during the iterative mixture splitting for this model, only distinct means were generated, covariance matrices were tied. Thus the total number of distinct covariance per state was 16, though there were 28 Gaussian components[3]. An EMLLT system was then built on this model set.

System	Number of Components	WER (%) ML	WER (%) MPE
HLDA	16	33.5	30.8
EMLLT		32.6	30.1
HLDA	28	32.3	29.9
EMLLT	$28\boldsymbol{\mu}$ $16\boldsymbol{\Sigma}$	31.9	29.6

Table 3. Comparison of WER for MPE trained HLDA and EMLLT models on the `dev01sub` test set

Table 3 shows the results of MPE training of the 16 component HLDA and EMLLT (78 basis) systems. For both systems significant gains were obtained using MPE training. The EMLLT system was 0.7% absolute better than the HLDA system after MPE training. The HLDA system was then iterative split, either in the standard for the HLDA system or in tied-covariance fashion, to 28 components. The results for both MLE and MPE training are also shown in table 3. The gains of EMLLT system over the HLDA system was significantly smaller for this larger system that the 16 component gains. This gain was reduced to only 0.3% absolute after MPE training. Using a pair-wise significance test, this was not a significant difference. This reduction in performance is probably due to the increased number of parameters in the EMLLT system. Additional tying, either within the SPAM or the hierarchical fashion may yield improved performance.

[3]Compared to the standard 28 component system this was only 0.1% absolute worse in error rate.

6. CONCLUSIONS

This paper presented a generic framework of basis superposition for precision matrix modelling. Basis *vectors* are globally tied while basis *coefficients* are component specific parameters that can be tied to eliminate redundancies. If the basis coefficients are untied, this form of model subsumes the STC and EMLLT models. Tying these coefficients over different basis vectors gives the SPAM model. Alternatively, basis coefficients can be tied over multiple Gaussian components. This paper illustrated HLDA and tied-variance systems as two simple examples of such tying. These models were found to give similar performance, but fewer model parameters compared to those with untied coefficients, Finally, this paper discussed the theory and experimental results of MPE training on EMLLT models.

7. REFERENCES

[1] A-V.I. Rosti and M.J.F. Gales, "Factor analysed hidden Markov models for speech recognition," Tech. Rep. CUED/F-INFENG/TR453, Cambridge University, 2003, Available via anonymous ftp from: svr-www.eng.cam.ac.uk.

[2] M.J.F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[3] R.A. Gopinath, "Maximum likelihood modeling with Gaussian distributions for classification," in *Proc. ICASSP*, 1998, pp. II–661–II–664.

[4] P. Olsen and R.A. Gopinath, "Extended MLLT for Gaussian Mixture Models," in *IBM, T.J. Watson Research Center*.

[5] V. Goel, S. Axelrod, R. Gopinath, P Olsen, and K. Visweswariah, "Discriminative estimation of subspace precision and mean (SPAM) models," in *Proc. Eurospeech '03*, Geneva, Switzerland, 2003, pp. 2617–2620.

[6] N. Kumar, *Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition*, Ph.D. thesis, Johns Hopkins University, 1997.

[7] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book (for HTK version 3.0)*, Cambridge University, 1997.

[8] D. Povey and P.C. Woodland, "Large scale discriminative training of hidden Markov models in speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 25–48, Jan 2002.

[9] D. Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. thesis, Cambridge University, 2003.