INTERSPEECH 2009 BRIGHTON

Acoustic Modeling Using Exponential Families

Vaibhava Goel, Peder Olsen

T.J. Watson Research Center, IBM

{vgoel,pederao}@us.ibm.com

Abstract

We present a framework to utilize general exponential families for acoustic modeling. Maximum Likelihood (ML) parameter estimation is carried out using sampling based estimates of the partition function and expected feature vector. Markov Chain Monte Carlo procedures are used to draw samples from general exponential densities. We apply our ML estimation framework to two new exponential families to demonstrate the modeling flexibility afforded by this framework.

Index Terms: Acoustic Modeling, Exponential Families, Markov Chain Monte Carlo Sampling, Metropolis, Hybrid Monte Carlo

1. Introduction

Although diagonal covariance gaussian mixture models (GMMs) suffuse acoustic modeling, there have been several alternative models that surpass diagonal covariance models in performance. Examples are other GMMs that take covariance into acount, such as semi-tied covariance [1], subspace constrained GMMs [2, 3]. Also, there are non-gaussian distributions such as the Richter distribution and power exponential distributions [4].

In this paper we present a framework to utilize general exponential families for acoustic modeling. This allows for a very rich set of distributions, including Gamma, Weibull (with known shape parameter), Chi-square, and of course the diagonal and full covariance Gaussians.

Exponential families and their properties are well understood. They have a number of properties that make parameter estimation simple and feasible. They have previously been used extensively in statistics [5] and for Language Modeling [6]. However, their use in acoustic modeling has been limited.

2. Exponential Families

We define an exponential family as any family of distributions on $\mathbb{R}^d$, parameterized by $\boldsymbol{\theta}$, that can be written

$$P(\mathbf{x}|\boldsymbol{\theta}) = \frac{e^{\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x})}}{Z(\boldsymbol{\theta})}, \tag{1}$$

where $\mathbf{x}$ are d-dimensional *base* observations. The function $\boldsymbol{\phi} : \mathbb{R}^d \longrightarrow \mathbb{R}^D$ gives the *derived* features and characterizes the exponential family. $Z(\boldsymbol{\theta})$ is the normalizer, also known as the partition function

$$Z(\boldsymbol{\theta}) = \int e^{\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x})}\, \mathrm{d}\mathbf{x}. \tag{2}$$

2.1. Normal distributions as an Exponential Family

A gaussian $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ can be represented as an exponential family. The features and parameters are

$$\boldsymbol{\phi}(\mathbf{x}) = \begin{bmatrix} \mathbf{x} \\ -\frac{1}{2}\mathrm{vec}(\mathbf{x}\mathbf{x}^T) \end{bmatrix} \quad \boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\ \mathrm{vec}(\boldsymbol{\Sigma}^{-1}) \end{bmatrix}. \tag{3}$$

Here for any $d \times d$ symmetric matrix $\mathbf{S}$, $\mathrm{vec}(\mathbf{S})$ is a column vector whose entries are the $d(d+1)/2$ upper triangular elements of $\mathbf{S}$ written in some fixed order, with the off diagonal elements multiplied by $\sqrt{2}$. The normalizer is given by

$$2 \log Z(\boldsymbol{\theta}) = \log \det(2\pi\boldsymbol{\Sigma}) + \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}. \tag{4}$$

2.2. Exponential Mixture Models

To model HMM state emission densities, we use mixtures of exponential densities

$$P(\mathbf{x}|s) = \sum_{e \in \mathcal{E}(s)} \pi_e P(\mathbf{x}|\boldsymbol{\theta}_e) = \sum_{e \in \mathcal{E}(s)} \pi_e \frac{e^{\boldsymbol{\theta}_e^T \boldsymbol{\phi}(\mathbf{x})}}{Z(\boldsymbol{\theta}_e)}. \tag{5}$$

When (3) is used, the mixture model above is the GMM.

3. Maximum Likelihood Estimation

Maximum likelihood (ML) estimation for exponential families is well studied; it was recently described in detail by Axelrod et. al. [2]. The exponential families considered in this paper had known, closed form partition functions. We now study exponential families with unknown partition functions. However, the ML estimation theory discussed in [2] is applicable.

3.1. Maximum Likelihood (ML) Estimation

Given acoustic training data and transcriptions, $\{(X_t, W_t)\}_{t=1}^T$, the maximum likelihood (ML) estimation aims to maximize

$$F(\Theta) = \prod_{t=1}^{T} P(X_t|W_t, \Theta) \tag{6}$$

where $\Theta = \{(\pi_e, \boldsymbol{\theta}_e)\}$ denotes all the observation model parameters in the HMM.

Using the expectation-maximization (EM) procedure, the auxiliary function to be maximized is

$$\begin{aligned} L(\boldsymbol{\theta}) &= \boldsymbol{\theta}^T \sum_t \gamma(t,e) \boldsymbol{\phi}(\mathbf{x}_t) - \log Z(\boldsymbol{\theta}) \sum_t \gamma(t,e), \\ &= \boldsymbol{\theta}^T \mathbf{s}(e) - n(e) \log Z(\boldsymbol{\theta}), \end{aligned} \tag{7}$$

where $\gamma(t, e)$ is the posterior probability of observing component e at time t, and the sufficient statistics are

$$n(e) = \sum_t \gamma(t, e) \quad (8)$$

$$\mathbf{s}(e) = \sum_t \gamma(t, e)\boldsymbol{\phi}(\mathbf{x}_t) \quad (9)$$

3.2. Convexity of $L(\boldsymbol{\theta})$

Since $\boldsymbol{\theta}^T\mathbf{s}$ is linear in $\boldsymbol{\theta}$, to show that $L(\boldsymbol{\theta})$ is convex, it is enough to show that the log partition function, $\log Z(\boldsymbol{\theta})$, is concave.
The first and second derivatives are

$$\frac{\partial \log Z}{\partial \boldsymbol{\theta}} = \mathrm{E}_{\boldsymbol{\theta}}[\boldsymbol{\phi}(\mathbf{x})] \stackrel{\text{def}}{=} G(\boldsymbol{\theta}) \quad (10)$$

$$\frac{\partial Z}{\partial \boldsymbol{\theta}\partial \boldsymbol{\theta}^T} = \mathrm{E}[\boldsymbol{\phi}(\mathbf{x})\boldsymbol{\phi}(\mathbf{x})^T] - \Big(\mathrm{E}[\boldsymbol{\phi}(\mathbf{x})]\mathrm{E}[\boldsymbol{\phi}^T(\mathbf{x})]\Big)^2 = \mathrm{Var}[\boldsymbol{\phi}(\mathbf{x})]. \quad (11)$$

Clearly, the co-variance is positive definite, so the Hessian is positive definite as well, which means the log partition function is concave as claimed.

4. Numerical Integration and Sampling

For general exponential models, there is no analytic solution for maximizing $L(\boldsymbol{\theta})$ and we use gradient based numerical optimization methods. This requires us to compute $L(\boldsymbol{\theta})$ and its gradient $s(e) - n(e)E_{\boldsymbol{\theta}}[\boldsymbol{\phi}(\mathbf{x})]$. For $L(\boldsymbol{\theta})$, the tricky part is to compute the partition function $Z(\boldsymbol{\theta})$.
As discussed in Neal [7], the partition function is often computed using a reference parameter value $\boldsymbol{\theta}^r$ which is sufficiently close to $\boldsymbol{\theta}$ and $Z(\boldsymbol{\theta}_r)$ is known. $Z(\boldsymbol{\theta})$ is computed as

$$\Psi(\boldsymbol{\theta}, \boldsymbol{\theta}_r) \stackrel{\text{def}}{=} \frac{Z(\boldsymbol{\theta}_r)}{Z(\boldsymbol{\theta})} = \int \frac{e^{\boldsymbol{\theta}_r^T\boldsymbol{\phi}(\mathbf{x})}}{e^{\boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x})}} \frac{e^{\boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x})}}{Z(\boldsymbol{\theta})}\, \mathrm{d}\mathbf{x} \quad (12)$$

We define sufficient closeness of $\boldsymbol{\theta}_r$ and $\boldsymbol{\theta}$ as

$$\sup_{\mathbf{x}}\{e^{\boldsymbol{\theta}_r^T\boldsymbol{\phi}(\mathbf{x})} - e^{\boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x})}\} \le th. \quad (13)$$

If $\boldsymbol{\theta}$ is not sufficiently close to $\boldsymbol{\theta}_r$, $Z(\boldsymbol{\theta})$ could be computed by taking small steps from $\boldsymbol{\theta}_r$ to $\boldsymbol{\theta}$

$$\frac{Z(\boldsymbol{\theta}_r)}{Z(\boldsymbol{\theta})} = \frac{Z(\boldsymbol{\theta}_r)}{Z(\boldsymbol{\theta}_1)} \cdot \frac{Z(\boldsymbol{\theta}_1)}{Z(\boldsymbol{\theta}_2)} \cdots \frac{Z(\boldsymbol{\theta}_N)}{Z(\boldsymbol{\theta})} \quad (14)$$

so that $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_{k+1}$ are sufficiently close.
For low dimensional problems we can evaluate the integral in (12) using numerical integration, whereas for high dimensional problems ($d > 3$) we use sampling. If we draw samples $\{\mathbf{x}_i\}_{i=1}^N$ from $P(\mathbf{x}|\boldsymbol{\theta}) = \frac{e^{\boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x})}}{Z(\boldsymbol{\theta})}$,

$$\Psi(\boldsymbol{\theta}, \boldsymbol{\theta}_r) \approx \widehat{\Psi}_N(\boldsymbol{\theta}, \boldsymbol{\theta}_r) = \frac{1}{N}\sum_{i=1}^N \frac{e^{\boldsymbol{\theta}_r^T\boldsymbol{\phi}(\mathbf{x}_i)}}{e^{\boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x}_i)}} \quad (15)$$

Similarly, the expected value of $\boldsymbol{\phi}(\mathbf{x})$, $G(\boldsymbol{\theta})$, can be approximated as

$$G(\boldsymbol{\theta}) \approx \widehat{G}_N(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \boldsymbol{\phi}(\mathbf{x}_i) \quad (16)$$

The sampling approximations given in (15) and (16) are not the only possible approximations. An alternative for (15) would utilize samples from $\boldsymbol{\theta}_r$, and yet another could utilize samples from both $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_r$.
To draw samples from $\frac{e^{\boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x})}}{Z(\boldsymbol{\theta})}$, we compared two variants of the Markov Chain Monte Carlo sampling procedure: a basic Metropolis algorithm and a Hybrid Monte Carlo algorithm. For Hybrid Monte Carlo, we also tried a special case known as Langevin Monte Carlo. These methods have several desirable properties, including applicability to fairly arbitrary distributions, ease of implementation, and ability to generate samples from un-normalized distributions. We describe our implementations of these procedures briefly in the following.

4.1. Basic Metropolis Sampling Algorithm

The Metropolis procedure is

1. Choose a starting sample, $\mathbf{x}_0$.
2. At iteration k, perturb $\mathbf{x}_k = \mathbf{x}_{k-1} + N(0, \Sigma)$
3. Accept $\mathbf{x}_k$ as the new sample with probability $\min(1, q)$ where
$$q = \frac{e^{\boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x}_k)}}{e^{\boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x}_{k-1})}}$$
4. If $\mathbf{x}_k$ is accepted, it is the next sample. If it is rejected, use $\mathbf{x}_{k-1}$ as the next sample. Go to Step 2.

The perturbation covariance Σ, used in Step 2, could be a function of the exponential model parameters $\boldsymbol{\theta}$. However, in this paper Σ was kept at a fixed diagonal covariance.

4.2. Hybrid Monte Carlo Sampling Algorithm

Hybrid Monte Carlo is a very effective technique for sampling from distributions where the gradient of the density function with respect to the base observations can be efficiently computed. Our implementation, based largely on the description by Neal [7], is

Algorithm:

1. Choose a starting sample, $\mathbf{x}_0$, and starting momentum $\mathbf{p}_0 \sim N(0, I)$
2. At iteration k, starting from ($\mathbf{x}_{k-1}$, $\mathbf{p}_{k-1}$), take N leapfrog steps to compute ($\mathbf{x}_k$, $\mathbf{p}_k$), as described in the following.
3. Accept $\mathbf{x}_k$ with probability $\min(1, q)$ where
$$q = \frac{e^{\boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x}_k) - 0.5\mathbf{p}_k^T\mathbf{p}_k}}{e^{\boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x}_{k-1}) - 0.5\mathbf{p}_{k-1}^T\mathbf{p}_{k-1}}}$$
4. If $\mathbf{x}_k$ is accepted, it is the next sample. If it is rejected, use $\mathbf{x}_{k-1}$ as the next sample. Go to Step 2.

Leapfrog steps:

1. Choose a direction $s = +1$ or -1, and a stepsize $\epsilon = s\,\gamma$
2. Starting from ($\mathbf{x}$, $\mathbf{p}$) first take a half step of $\mathbf{p}$
$$\mathbf{p} = \mathbf{p} - \frac{\epsilon}{2}\nabla_{\mathbf{x}}(\boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x}))$$
3. Next, take $N - 1$ steps of $\mathbf{x}$ and $\mathbf{p}$
$$\mathbf{x} = \mathbf{x} + \epsilon\,\mathbf{p}$$
$$\mathbf{p} = \mathbf{p} - \epsilon\,\nabla_{\mathbf{x}}(\boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x}))$$
4. Finally, take a full step of $\mathbf{x}$, as above, and a half step of $\mathbf{p}$, as the first half step

We also experimented with the special case with one leap frog step $N = 1$ per sample, known as Langevin Monte Carlo sampling.

5. Exponential Families In This Paper

Besides the multivariate diagonal Gaussians, we experimented with the following two exponential families.

5.1. Power Exponential Like Distribution

For this model, we use the following features

$$\phi(\mathbf{x}) = \begin{bmatrix} \mathbf{x} \\ |\mathbf{x}|^{\alpha} \end{bmatrix} \quad (17)$$

$\alpha = 2$ corresponds to the diagonal Gaussian model. We experimented with $\alpha = 1.5$.

5.2. Gaussians Augmented With Quadrant Indicator Features

In this model, we added feature functions to indicate quadrant of first four dimensions of the feature vector, with the intent of capturing a primitive form of correlation between first four feature dimensions. The indicator features were of the form

$$\begin{aligned} I_1 &= I_{(\mathbf{x}[0]>0,\mathbf{x}[1]>0,\mathbf{x}[2]>0,\mathbf{x}[3]>0)}(\mathbf{x}) \\ I_2 &= I_{(\mathbf{x}[0]>0,\mathbf{x}[1]>0,\mathbf{x}[2]>0,\mathbf{x}[3]<0)}(\mathbf{x}) \\ &\vdots \end{aligned}$$

There were 16 such functions in total. These features were added to standard diagonal Gaussian features.

6. Experimental Setup

The experiments presented in this paper were carried out on a Mandarin data set. The training set consisted of about 500hrs of audio. The LDA feature vectors were obtained by first computing 13 Mel-cepstral coefficients (including energy) for each time slice under a 25.0 msec. window with a 15 msec. shift. Nine such vectors were concatenated and projected to a 40 dimensional space using LDA. These LDA features were used as the base observations for exponential models.

The phoneme inventory of the acoustic model consisted of 182 phonemes. Each of these was modeled using a 3 state left-to-right HMM. These HMM states were modeled using a total of 1180 context dependent states and 25K Gaussians.

The test set consisted of 43 tasks, each having its own finite state grammar to decode with. The tasks included booleans, names, currency, dates, times, and stock quotes. The test set had 185K words from 31K sentences.

7. Experimental Results

7.1. Comparison of Metropolis and Hybrid Monte Carlo

To compare the sampling procedures, we draw samples from three 1-dimensional Gaussian distributions, all with mean 1.0 and variance 0.01, 1.0, and 100.0, respectively. For each of these Gaussians, a reference Gaussian is chosen with the same variance and mean perturbed by 0.2 times the standard deviation. Since the sampling and reference distributions are Gaussians with known parameters, we can compute the true partition function ratio $\Psi(\boldsymbol{\theta}, \boldsymbol{\theta}_r)$ and true expected feature vector $G(\boldsymbol{\theta})$. As a reference sampling method, we used the well known Box-Muller [8] method to generate samples from a Gaussian distribution.

Figure 1 shows $||G(\boldsymbol{\theta}) - \widehat{G}_N(\boldsymbol{\theta})||_\infty$, i.e. the max absolute error in any dimension of expected feature vector estimation, as a function of N, for Gaussian with variance 1.0. From this figure, we note that, as expected, samples drawn using Box-Muller are clearly better than any other method. Hybrid Monte Carlo is clearly the next best method.

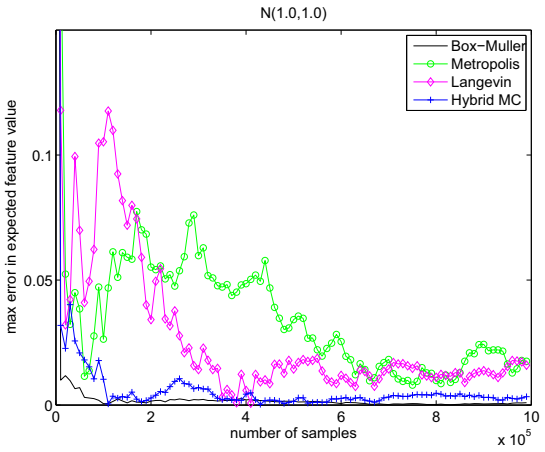

Figure 1: Maximum error in expected features as a function of number of samples.

To compactly present convergence results for various Gaussians under various sampling method, we compute

$$N_\epsilon^G = \min\left\{ N \text{ s.t. } ||\frac{G(\boldsymbol{\theta}) - \widehat{G}_k(\boldsymbol{\theta})}{G(\boldsymbol{\theta})}||_\infty < \epsilon \ \forall k > N \right\}$$

$$N_\epsilon^\Psi = \min\left\{ N \text{ s.t. } |1.0 - \frac{\widehat{\Psi}_k(\boldsymbol{\theta}, \boldsymbol{\theta}_r)}{\Psi(\boldsymbol{\theta}, \boldsymbol{\theta}_r)}| < \epsilon \ \forall k > N \right\}$$

Table 1 (a) shows N_ϵ^G for $\epsilon = 0.05$, and 1 (b) shows N_ϵ^Ψ for $\epsilon = 0.05$. Again, Hybrid Monte Carlo seems to be best among the Markov Chain sampling methods.

	1-dim Gaussian variance:			40-dim Gaussian	
	0.01	1.0	100.0		
(a) N_ϵ^G					
Box-Muller	25	1040	11015	1375	1455
Metropolis	135	4.4e5	1e6	5.0e5	4.7e5
Langevin MC	40	2.2e5	1e6	9.3e5	1.6e5
Hybrid MC	55	5830	8.9e5	39750	11920
(b) N_ϵ^Ψ					
Box-Muller	30	30	30	25	5
Metropolis	800	7245	5.3e5	3490	11055
Langevin MC	180	3655	8.0e5	36080	32560
Hybrid MC	300	775	9735	170	1140

Table 1: Number of samples needed to reach within 5% of true value

Next, we compared the sampling procedures for multivariate Gaussians. We randomly chose two Gaussians from our ML trained acoustic model. As with 1-d Gaussians, for each of these 40-dim Gaussians, a reference 40-dim Gaussian was chosen with the same variance and mean perturbed by 0.05.

The last two columns of Table 1 show the number of samples needed to reach within 5% of true values for these multivariate Gaussians. As with 1-dim Gaussians, Hybrid Monte Carlo

seems to perform best among the Markov Chain methods we tried.

7.2. Using Exponential Model Framework for Diagonal Gaussians

Our next step was to evaluate the gradient descent based ML estimation for diagonal Gaussians treated as exponential models. Starting with a seed GMM of 25124 Gaussians corresponding to 1180 context-dependent state, we carried out 15 iterations of expectation-maximization (EM) training. In each EM iteration, sufficient statistics were gathered and Gaussians updated using a standard gradient descent procedure. For each Gaussian, a maximum of 250 iterations of gradient descent procedure were run in each EM iteration. All sampling used the Hybrid Monte Carlo algorithm, with $N = 5$ leapfrog iterations per sample and a step-size $\gamma = 0.1$.

Figure 2 shows the error rate on test set as a function of EM iterations. This figure also shows the baseline performance obtained using the standard Gaussian training procedure where once the sufficient statistics are gathered, the Gaussians are updated using the closed form update equations.

From Figure 2 we note that the numerical optimization comes very close to the analytic estimation. The fact that over several iterations the two curves remain close suggests that our estimation framework, despite all the sampling involved, is stable.

At the moment our implementation is significantly slower than analytic update.

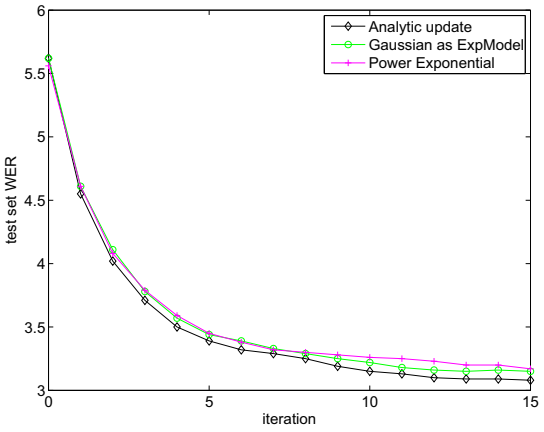

Figure 2: Error rate variations according to number of EM iterations.

7.3. Power Exponential Like Distribution

These exponential models were initialized with the GMM that was the starting point for experiments reported in Section 7.2. For $\alpha = 1.5$ we computed the partition function, mean, and variance for each dimension using numerical integration. The initial parameters were chosen so as to match the mean and variance of the diagonal Gaussian. 15 EM iterations were carried out. In each EM iteration, at most 100 gradient descent iterations were allowed for each exponential model.

Figure 2 shows the test set word error rate as a function of number of iterations. We note that our new exponential model is slightly worse in error rate as compared to GMM. However, over several iterations it is able to track the GMM performance closely.

7.4. Gaussians Augmented With Quadrant Indicator Features

As discussed in Section 5.2, 16 features were added to standard diagonal Gaussians with 80 exponential model parameters. A well converged GMM using 30 EM iterations was chosen as the starting point. The parameters for the 16 new features were initialized to 0.0 and 7 EM iterations on the exponential model were run to learn all 96 parameters.

Baseline GMM WER = 3.04%							
0	1	2	3	4	5	6	7
3.04	3.29	3.11	3.09	3.08	3.07	3.07	3.06

Table 2: WER as a function of iteration number for Gaussians augmented with quadrant indicator features

Table 2 shows the WER as a function of these iterations. From these results, we note that at the first iteration there's a loss of about 10% relative. With further iterations, the WER almost comes back to the baseline.

8. Conclusions

We have demonstrated feasibility of acoustic modeling using general exponential families with unknown partition functions. Our key result is that Markov Chain sampling seems to suffice in estimating the EM auxiliary function and its gradient, and the exponential model parameters and partition function could be reliably estimated over several EM iterations. However, the two exponential families we evaluated did not give a gain in recognition accuracy over Gaussian mixture models.

9. References

[1] M. J. F. Gales, "Semi-tied covariance matrices for hidden markov models," *IEEE Transactions on Speech and Audio Processing*, 1999.

[2] S. Axelrod, V. Goel, R. A. Gopinath, P. A. Olsen, and K. Visweswariah, "Subspace constrained gaussian mixture models for speech recognition," *Transactions in Speech and Audio Processing*, vol. 13, no. 6, pp. 1144–1160, November 2005.

[3] V. Vanhoucke and A. Sankar, "Mixtures of inverse covariances," in *Proceedings of ICASSP 2003*, 2003.

[4] M. J. F. Gales and P. Olsen, "Tail distribution modelling using the Richter and power exponential distributions," in *Eurospeech '99 - 6th European Conference on Speech Communication and Technology*, no. 4, Budapest, Hungary, September 1999, pp. 1507–1510.

[5] L. D. Brown, *Fundamentals of Statistical Exponential Families with Applications in Statistical Decision Theory*, ser. Monograph series. California: Hayward, 1986, vol. 9, institute of Mathematical Statistics.

[6] A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[7] R. M. Neal, "Probabilistic inference using Markov Chain Monte Carlo methods," Dept. of Computer Science, University of Toronto, Tech. Rep. CRG-TR-93-1, 1993.

[8] G. E. P. Box and M. E. Muller, "A note on the generation of random normal deviates," *Ann. Math. Statist.*, vol. 29, no. 2, pp. 610–611, 1958.