A FAST, ACCURATE APPROXIMATION TO LOG LIKELIHOOD OF GAUSSIAN MIXTURE MODELS

Pierre L. Dognin, Vaibhava Goel, John R. Hershey and Peder A. Olsen

IBM T.J. Watson Research Center
Yorktown Heights, NY 10598, USA
{pdognin,vgoel,jrhershe,pederao}@us.ibm.com

ABSTRACT

It has been a common practice in speech recognition and elsewhere to approximate the log likelihood of a Gaussian mixture model (GMM) with the maximum component log likelihood. While often a computational necessity, the max approximation comes at a price of inferior modeling when the Gaussian components significantly overlap. This paper shows how the approximation error can be reduced by changing component priors. In our experiments the loss in word error rate due to max approximation, albeit small, is reduced by 50-100% at no cost in computational efficiency. Furthermore, we expect acoustic models will become larger with time and increase component overlap and word error rate loss. This makes reducing the approximation error more relevant. The techniques considered do not use the original data and can easily be applied as a post-processing step to any GMM.

Index Terms— Gaussian mixture model, acoustic model, maximum approximation, exponential distribution

1. INTRODUCTION

Modern speech recognition systems have acoustic models with thousands of context dependent hidden Markov model states, each modeled with a Gaussian mixture model (GMM). The total number of component Gaussians easily exceed 100,000 and exact log likelihood evaluation becomes prohibitively expensive. Clever use of hierarchies of Gaussian clusters, [1, 2, 3, 4], efficiently locates top Gaussians while only of the order of 1000 Gaussians are evaluated. In such systems, exact evaluation is impossible and improvements in the max approximations are useful.

A GMM is a distribution whose marginal density of $\mathbf{x} \in \mathbb{R}^d$ is

$$f(\mathbf{x}) = \sum_{i=1}^{n} \pi_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \tag{1}$$

We will use the shorthand $f_i(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ for the Gaussian component of f. A common approximation to (1) uses $\log(a+b) \approx \max(a, b)$, which holds for positive numbers $a \gg b > 0$. The resulting approximation

$$f(\mathbf{x}) \approx \tilde{g}(\mathbf{x}) = \max_i \pi_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \tag{2}$$

satisfies the bound $\log \tilde{g}(\mathbf{x}) < \log f(\mathbf{x}) \leq \log n + \log \tilde{g}(\mathbf{x})$ and is within $\log n$ of the exact value. In general, the approximation will be better for smaller n and for well separated components. The upper bound will be attained only in the case when all the values $\pi_i f_i(\mathbf{x})$ are equal. The related special case $f_i = f_1$, $i = 1, \ldots, n$ is an important special case that we shall refer to as *extreme* overlap.

We shall consider the more general approximation

$$g(\mathbf{x}) = \max_i \omega_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

where ω_i may be chosen such that $\sum_i \omega_i \neq 1$. For extreme overlap, $\omega_i = 1$ gives the exact value of f. The challenge is to do better for the cases of moderate overlap too. The approximation $g(\mathbf{x})$ is more general than $\tilde{g}(\mathbf{x})$ and requires the same amount of computation. Thus we believe there exist ω_i that approximate f better than $\tilde{g}$. We explore how to choose $\boldsymbol{\omega}$ in this paper.

The rest of this paper is organized as follows. Section 2 introduces two expected value strategies for obtaining ω, Section 3 discusses how to scale ω to make g a distribution, and Section 4 shows how to estimate ω to minimize the Kullback-Leibler divergence between g and f. Section 5 shows experimental results for each technique.

2. EXPECTED VALUE OF THE PRIORS

Let B_i be the regions where component i dominates

$$B_i \stackrel{\text{def}}{=} \{\mathbf{x} : f_i(\mathbf{x}) \geq f_j(\mathbf{x}) \text{ for all } i \neq j\}$$

If ω_i is allowed to vary with $\mathbf{x} \in B_i$ then $g(\mathbf{x})$ is *exactly* equal to $f(\mathbf{x})$ for the choice

$$\omega_i(\mathbf{x}) = \frac{f(\mathbf{x})}{f_i(\mathbf{x})} = \frac{\sum_{j=1}^{n} \pi_j f_j(\mathbf{x})}{f_i(\mathbf{x})}.$$

The expected value of $\omega_i(\mathbf{x})$ is independent of $\mathbf{x}$ and can be used for ω_i. We have

$$\mathrm{E}_f[\omega_i(\mathbf{x}) | \mathbf{x} \in B_i] = \sum_{j=1}^{n} \pi_j \mathrm{E}_f[f_j(\mathbf{x})/f_i(\mathbf{x}) | \mathbf{x} \in B_i].$$

Using Jensen's inequality we have the approximate expression

$$\begin{aligned} \mathrm{E}_f[f_j(\mathbf{x})/f_i(\mathbf{x}) | \mathbf{x} \in B_i] &\geq e^{-\mathrm{E}_{f_i}[\log(f_i(\mathbf{x})/f_j(\mathbf{x})) | \mathbf{x} \in B_i]} \\ &\approx e^{-\mathrm{E}_{f_i}[\log(f_i(\mathbf{x})/f_j(\mathbf{x}))]} \\ &= e^{-D(f_i \| f_j)} \end{aligned}$$

where going from the first to the second line we are assuming the quantity is dominant inside of $\mathcal{B}_i$. Consequently we get

$$\mathrm{E}_f[\omega_i(\mathbf{x}) | \mathbf{x} \in B_i] \approx \sum_{j=1}^{n} \pi_j e^{-D(f_i \| f_j)}. \tag{3}$$

This can be computed analytically and very efficiently.

A more computationally intensive approach would be to estimate the expected value directly

$$\mathrm{E}_f[\omega_i(\mathbf{x})|\mathbf{x}\in B_i] = \frac{\int_{B_i}\frac{(f(\mathbf{x}))^2}{f_i(\mathbf{x})}\mathrm{d}\mathbf{x}}{\int_{B_i} f(\mathbf{x})\mathrm{d}\mathbf{x}}.$$

If we draw samples $\{\mathbf{x}_k\}_{k=1}^N$ from $f(\mathbf{x})$ the Monte Carlo estimate of the expected value is

$$\mathrm{E}_f[\omega_i(\mathbf{x})|\mathbf{x}\in B_i] \approx \frac{\sum_{k=1}^N \frac{f(\mathbf{x}_k)}{f_i(\mathbf{x}_k)} 1_{\{x\in\mathcal{B}_i\}}(\mathbf{x}_k)}{\sum_{k=1}^N 1_{\{x\in\mathcal{B}_i\}}(\mathbf{x}_k)}. \quad (4)$$

Introducing the index sets, $\mathcal{B}_i = \{k : \mathbf{x}_k \in B_i\}$, the last expression can be more compactly written

$$\mathrm{E}_f[\omega_i(\mathbf{x})|\mathbf{x}\in B_i] \approx \frac{\sum_{k\in\mathcal{B}_i}\frac{f(\mathbf{x}_k)}{f_i(\mathbf{x}_k)}}{\sum_{k\in\mathcal{B}_i} 1}.$$

3. THE MAX DISTRIBUTION

As noted in the introduction the max approximation $\tilde{g}(\mathbf{x})$ is bounded from above by $f(\mathbf{x})$. Therefore

$$\int \tilde{g}(\mathbf{x})\mathrm{d}\mathbf{x} \le \int f(\mathbf{x})\mathrm{d}\mathbf{x} = 1$$

and $\tilde{g}(\mathbf{x})$ is in general not a probability distribution function. We can choose

$$\omega_i = \pi_i/\alpha \quad (5)$$

in such a way that $g(\mathbf{x})$ becomes a distribution. This gives $\alpha = \int \tilde{g}(\mathbf{x})\mathrm{d}\mathbf{x}$. We estimate the integral as before by drawing samples $\{\mathbf{x}_k\}_{k=1}^N$ from f, giving the Monte Carlo estimate

$$\alpha \approx \frac{1}{N}\sum_{k=1}^N \frac{\tilde{g}(\mathbf{x}_k)}{f(\mathbf{x}_k)}. \quad (6)$$

The previous section did not enforce the normalization constraint, $\int g(\mathbf{x})\mathrm{d}\mathbf{x} = 1$. We could use the technique of this section to similarly scale the priors $\boldsymbol{\omega}$ of equations (3) and (4) but we have not done that in this paper.

4. MINIMIZING THE KULLBACK LEIBLER DIVERGENCE

The Kullback Leibler divergence between the max-approximation g and the GMM f is given by

$$D(f\|g) = \int f(\mathbf{x})\log(f(\mathbf{x})/g(\mathbf{x}))\,\mathrm{d}\mathbf{x},$$

which can be estimated by drawing samples $\{\mathbf{x}_k\}_{k=1}^N$ from f. The Monte Carlo approximation is given by

$$\begin{aligned} D(f\|g) &\approx \frac{1}{N}\sum_{k=1}^N \log\left(\frac{f(\mathbf{x}_k)}{g(\mathbf{x}_k)}\right) \\ &= \frac{1}{N}\sum_b\sum_{k\in\mathcal{B}_b}\log\left(\frac{f(\mathbf{x}_k)}{\omega_b f_b(\mathbf{x}_k)}\right) \\ &= \frac{1}{N}\sum_b\sum_{k\in\mathcal{B}_b}\log\left(\frac{f(\mathbf{x}_k)}{f_b(\mathbf{x}_k)}\right) - \log\omega_b \\ &= -\frac{1}{N}\sum_b |\mathcal{B}_b|\log\omega_b + C, \end{aligned}$$

where C is independent of ω_b, save through the sets $\mathcal{B}_b$. For the constraints we have in the same way,

$$\begin{aligned} \int g(\mathbf{x})\,\mathrm{d}\mathbf{x} &\approx \frac{1}{N}\sum_{k=1}^N \frac{g(\mathbf{x}_k)}{f(\mathbf{x}_k)} \\ &= \sum_b \omega_b \frac{1}{N}\sum_{k\in\mathcal{B}_b}\frac{f_b(\mathbf{x}_k)}{f(\mathbf{x}_k)} \\ &= \sum_b \omega_b \frac{F_b}{N}. \end{aligned}$$

F_b are the fractional counts $\sum_{k\in\mathcal{B}_b}\frac{f_b(\mathbf{x}_k)}{f(\mathbf{x}_k)}$. If we fix $\mathcal{B}_b$ corresponding to the present estimate of ω_b then the rest of the function can be minimized analytically satisfying the constraint. The Lagrangian to be optimized is

$$L(\omega,\lambda) = -\frac{1}{N}\sum_b |\mathcal{B}_b|\log\omega_b + \lambda(\frac{1}{N}\sum_b \omega_b F_b - 1)$$

Differentiating and equating to zero we get

$$\hat{\omega}_b = \frac{|\mathcal{B}_b|}{F_b} \quad (7)$$

and the corresponding value of $D(f\|g)$ is

$$-\frac{1}{N}\sum_b |\mathcal{B}_b|\log\frac{|\mathcal{B}_b|}{F_b} + C.$$

After we update ω with $\hat{\omega}$ according to this equation the sets $\mathcal{B}_b$ will no longer be consistent with $\hat{\omega}$ – if they were we have found the minimal value for ω. Thus we need to recompute the sets $\mathcal{B}_b(\hat{\omega})$, and iterate as follows.

4.1. An iterative algorithm to minimize the Kullback Leibler divergence

Putting together all the observations of the previous section we propose the following iterative algorithm

1. Precompute $f_b(\mathbf{x}_k)$ for all $k = 1,\ldots,N$ and $b = 1,\ldots,n$.
2. Compute $\mathcal{B}_b$ and F_b based on current value of ω_b.
3. Compute $\hat{\omega}_b = |\mathcal{B}_b|/F_b$.
4. Compute $\hat{\mathcal{B}}_b$ and $\hat{F}_b$ based on $\hat{\omega}_b$.
5. Compute $\alpha = \frac{1}{N}\sum_b \hat{\omega}_b\hat{F}_b$ and normalize $\hat{\hat{\omega}}_b = \hat{\omega}_b/\alpha$.
6. Let $\omega_b = \hat{\hat{\omega}}_b$, $\mathcal{B}_b = \hat{\mathcal{B}}_b$ and repeat from step 3 until convergence.

5. EXPERIMENTAL RESULTS

The merits of our proposed techniques were assessed on an IBM internal Chinese (Mandarin) test set. This data set was collected in automobiles under a variety of noise conditions. It has altogether 184,693 words from 31,067 sentences.

Two acoustic models, named 122K and 149K, were built as follows. For the 122K model, the acoustic feature vectors were obtained by first computing 13 Mel-cepstral coefficients (including energy) for each time slice under a 25 msec. window with a 15 msec. shift. Spectral subtraction [5] was applied during cepstrum computation. Nine such vectors were concatenated and projected to a 40 dimensional space using linear discriminant analysis (LDA). The acoustic

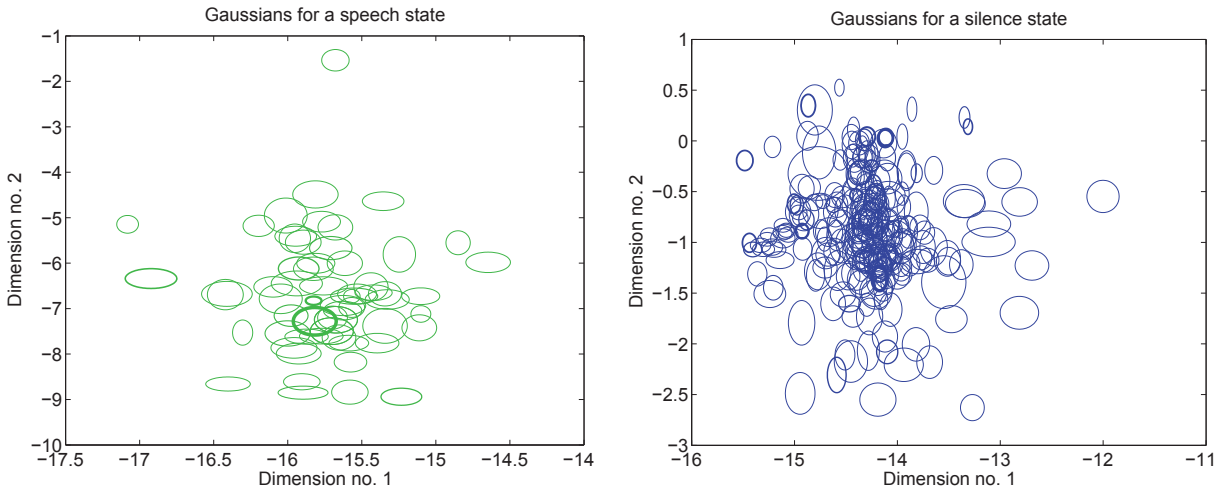

Fig. 1. The two figures are a graphical representation of two GMMs for respectively a speech and silence state of the 122K model. The silence GMM comprises 259 Gaussians and the speech GMM has 64 Gaussians. The first two dimensions of the 40 dimensional feature space are shown. Each Gaussian is represented as an ellipsoid centered at the mean and with radius equal to 0.2 standard deviations. Thicker lines corresponds to Gaussians with larger priors. Note that the silence state has considerably larger overlap between Gaussians than for the speech state. A picture in 2 dimensions could be deceptive as a visualization for a 40 dimensional Gaussian, but in this case pairwise KL divergences verifies the intuition. The number of Gaussians is probably the primary reason for the overlap.

models were built on these features with a phoneme set containing 182 phonemes. Each phoneme was modeled with a three state left to right HMM. Using a phonetic context of two phonemes to the left and right within the word and one phoneme to the left across the word, these phoneme states were clustered into 1,450 context dependent (CD) states. The CD states were then modeled with on average 84 Gaussians per state, resulting in a total of 122,366 Gaussians in the acoustic model.

The 149K model was built almost identically to the 122K model, except its feature space did not have spectral subtraction. This model had 2,143 context dependent states, an average of 70 Gaussians per state for a total of 148,942 Gaussians. We include results for two acoustic models simply to validate the effectiveness of our methods.

The training data set contains about 568 hours of audio, collected in automobiles under a variety of noise conditions. This is for the most part an IBM internal corpus with about 32 hours of data from the SPEECON [6] database.

The 122K model, when evaluated on the test set, had 4,752 words and 3,151 sentences in error, resulting in a word error rate of 2.57% and a sentence error rate of 10.14%. The 149K model had 5,377 word and 3,454 sentence errors, resulting in a word error rate of 2.91% and a sentence error rate of 11.12%. The worse error rate of the 149K model is due to the fact that its feature computation does not include spectral subtraction.

5.1. Visualizing overlap between Gaussians of GMM

To gain better intuition into the difference between GMM log-likelihood and its max approximation, we first tried to visualize the overlap between Gaussians of GMM. This is shown in Figure 1 for two GMMs, one modeling a state of silence and another modeling a state of speech.

From Figure 1 we note that in some regions of the feature space, especially so for silence GMM, there appears to be a considerable overlap between Gaussians. In those regions we would expect the max value to be a poor approximation to the GMM log-likelihood.

		silence	speech	overall
122K	baseline	0.033	0.035	0.035
	e^{-D}	0.022	0.034	0.034
	$\mathrm{E}[\omega]$	0.024	0.033	0.033
	norm ω	0.155	0.036	0.037
	min KL	0.020	0.032	0.032
149K	baseline	0.032	0.035	0.035
	e^{-D}	0.021	0.034	0.034
	$\mathrm{E}[\omega]$	0.039	0.032	0.033
	norm ω	0.285	0.035	0.037
	min KL	0.020	0.032	0.032

Table 1. Average absolute difference between sum and max log-likelihoods on the test data. e^{-D} and $\mathrm{E}[\omega]$ are with the prior updates of (3) and (4), respectively. norm ω numbers are with the prior normalization (5) and (6) of Section 3, and min KL numbers are with the prior update (7) of Section 4.

5.2. Difference between sum and max log-likelihoods & recognition performance of various techniques

To check the quality of our approximations to GMM log-likelihood, we computed the average difference between sum and max log-likelihoods on the entire test set for the techniques discussed in this paper. These values, for the 122K and 149K models, are presented in Table 1. The table also shows the average difference for silence and non-silence leaves separately for each of these techniques. The average GMM log-likelihood on the test data for the 122K model was 10.21 and for the 149K model was 10.54.

From Table 1, we first note that the average difference between sum and max log-likelihoods is about three orders of magnitudes less than

		max decode	relative change	sum decode	relative change
122K	baseline	4752		4678	-1.6%
	e^{-D}	4703	-1.0%	4691	-1.3%
	$\mathrm{E}[\omega]$	4740	-0.2%	4696	-1.2%
	norm ω	4706	-1.0%	4668	-1.8%
	min KL	4725	-0.6%	4694	-1.2%
149K	baseline	5377		5282	-1.8%
	e^{-D}	5327	-0.9%	5241	-2.5%
	$\mathrm{E}[\omega]$	5295	-1.5%	5224	-2.8%
	norm ω	5251	-2.3%	5177	-3.7%
	min KL	5353	-0.4%	5263	-2.1%

Table 2. Number of word errors and relative gain (or loss) resulting from various techniques. For each method, decoding with both max and sum log-likelihood scores is carried out, as shown in columns labeled max decode and sum decode, respectively. Baseline numbers use (1) for the sum and (2) for the max. e^{-D} and $\mathrm{E}[\omega]$ are with the prior updates of (3) and (4), respectively. norm ω numbers are with the prior normalization (5) and (6) of Section 3, and min KL numbers are with prior update (7) of Section 4.

the log-likelihood itself. This small difference still seems to be of significance for recognition errors, as seen from the gap between max and sum decoding with baseline models in Table 2.

The *sum decoding* for the rows other than the baseline uses the re-estimated priors in the computation of the GMM likelihood (1) without concern for whether the priors add up to 1. Interestingly, as seen from Table 2, the word errors from sum decoding also improve from their baseline value for the 149K model, and in one case for the 122K model. Our current hypothesis for this gain is as follows. The re-estimated priors that no longer sum to 1 effectively introduce state-dependent multipliers. States with more overlapping gaussians get larger multipliers. Whether these state-dependent multipliers are truly responsible for the observed gain in sum-decoding remains to be seen.

From the overall values in Table 1 it appears that on average all four techniques have a small impact on narrowing the gap between sum and max log-likelihoods. However, as seen from max decoding numbers in Table 2, they all have an appreciable positive impact on error rates.

Table 1 shows that min KL yields best approximation to GMM log-likelihood. However, it does not result in the best word error rate. This also is believed to be due to the secondary effect of the HMM state-dependent multipliers.

6. CONCLUSIONS AND FUTURE WORK

We note that while the loss in word error rate due to the max approximation to GMM log-likelihood is small, the tendency for acoustic models to become larger with time will increase component overlap and broaden the gap. The techniques considered in this paper have a positive impact on bridging this gap. Furthermore, the techniques considered have zero computational overhead and since they do not use the original data, they can easily be applied as a post-processing step to any GMM.

In the future, we plan to carry out direct EM training of the max distribution.

7. ACKNOWLEDGEMENT

The authors thank Ke Li and Guo Kang Fu for providing valuable data and references.

8. REFERENCES

[1] Raimo Bakis, David Nahamoo, Michael A. Picheny, and Jan Sedivy, "Hierarchical labeler in a speech recognition system," U.S. Patent 6023673. filed June 4, 1997, and issued February 8, 2000.

[2] A. Aiyer, M.J.F. Gales, and M. A. Picheny, "Rapid likelihood calculation of subspace clustered gaussian components," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, June 2000, vol. 3, pp. 1519–1522.

[3] Xiao-Bing Li, Frank K. Soong, Tor André Myrvoll, and Ren-Hua Wang, "Optimal Clustering and Non-Uniform Allocation of Gaussian Kernels in Scalar Dimension for HMM Compression," in *ICASSP*, 2005, vol. 1, pp. 669–672.

[4] E. Bocchieri and Brian K. Mak, "Subspace distribution clustering hidden Markov model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 9, no. 3, pp. 264–275, March 2001.

[5] D. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions of Acoustics, Speech and Signal Processing*, vol. 27, no. 2, pp. 113–120, April 1979.

[6] Dorota Iskra, Beate Grosskopf, Krzysztof Marasek, Henk van den Heuvel, Frank Diehl, and Andreas Kiessling, "SPEECON – speech databases for consumer devices: Database specification and validation," in *Proceedings of LREC*, 2002, pp. 329–333.