Korse, Srikanth; Fuchs, Guillaume; Bäckström, Tom

GMM-Based Iterative Entropy Coding for Spectral Envelopes of Speech and Audio

Published in:
Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

DOI:
10.1109/ICASSP.2018.8461527

Published: 01/01/2018

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Korse, S., Fuchs, G., & Bäckström, T. (2018). GMM-Based Iterative Entropy Coding for Spectral Envelopes of Speech and Audio. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5689-5693). Article 8461527 (Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing). IEEE. https://doi.org/10.1109/ICASSP.2018.8461527

GMM-BASED ITERATIVE ENTROPY CODING FOR SPECTRAL ENVELOPES OF SPEECH AND AUDIO

Srikanth Korse[1], *Guillaume Fuchs*[1,2], *Tom Bäckström*[3]

[1]Fraunhofer IIS, Erlangen, Germany, [3]Aalto University, Helsinki, Finland
[2]International Audio Laboratories, Friedrich-Alexander University (FAU), Erlangen, Germany
srikanth.korse@iis.fraunhofer.de

ABSTRACT

Spectral envelope modelling is a central part of speech and audio codecs and is traditionally based on either vector quantization or scalar quantization followed by entropy coding. To bridge the coding performance of vector quantization with the low complexity of the scalar case, we propose an iterative approach for entropy coding the spectral envelope parameters. For each parameter, a univariate probability distribution is derived from a Gaussian mixture model of the joint distribution and the previously quantized parameters used as a-priori information. Parameters are then iteratively and individually scalar quantized and entropy coded. Unlike vector quantization, the complexity of proposed method does not increase exponentially with dimension and bitrate. Moreover, the coding resolution and dimension can be adaptively modified without retraining the model. Experimental results show that these important advantages do not impair coding efficiency compared to a state-of-art vector quantization scheme.

Index Terms— Entropy Coding, Gaussian mixture models, Envelope Modelling, Speech Coding, Audio Coding

1. INTRODUCTION

Spectral envelope models form an integral part of speech and audio codecs. While speech codecs models the short-time spectral envelope with the help of linear predictive coding (LPC) parameters, audio codecs models the same usually with the help of scale factor bands [1, 2, 3]. A further parametrization of the envelope shape known as distribution quantization (DQ) was recently introduced in [4]. For transmission, all of these parameters need to be quantized and coded. DQ parameters as well inter-band differential spectral factors are considered mainly decorrelated. Therefore, they are typically scalar quantized and indices are further coded by a memoryless entropy coder. On the other hand, LPC parameters are usually converted to line spectral frequencies (LSF) before being vector quantized [5].

Typically, envelope parameters exhibit some degree of correlation, whereby vector quantizers (VQ) can be used instead of scalar quantizer to reduce the bitrate [6, 7, 8]. A vector quantizer is essentially a codebook which covers all possible input vectors, whereby the codebook size is a function of the number of dimension and bitrate [8, 5]. It increases exponentially with both increasing bitrate (with fixed dimension) or dimension (with fixed bitrate). Codebook size, in turn, is directly proportional to the computational complexity. Specifically, if we use B bits to transmit the codebook vector, then we have 2^B codebook vectors. If the input vector is of length N, then we need $\mathcal{O}(N2^B)$ operations to find the optimal codebook vector and an equal amount of storage. To reduce the complexity and memory requirements, several approaches have been proposed. One of the most popular ones is multi-stage vector quantization [9, 10], which splits the task into multiple stages, where each stage has only a small codebook. For a codebook with M stages, we thus have a computational complexity of $\mathcal{O}(NM\sum_k 2^{B_k})$.

Another approach for encoding the signal is to use parametric models of the distribution, such as Gaussian mixture models (GMM) [11, 12, 13]. Specifically, by assigning the input signal to a specific Gaussian, we can use the Karhunen-Lóeve transform to decorrelate samples, whereby the signal can be quantized and coded with conventional methods such as a lattice quantizer [12, 13]. The computational complexity of the method is then essentially independent of bit-rate. However, the approach does not provide optimal coding efficiency, since Gaussian components of the mixture will exhibit some degree of overlap. It follows that any input vector could (in theory) be assigned to any Gaussian component, whereby the representation has inherent redundancy.

Instead of decorrelating the signal, we propose an iterative process, where previously encoded samples of the signal are used as prior information for the following samples. By updating the parameters of each Gaussian component based on the prior samples, we can use a scalar entropy coder for the current sample. The advantages of our proposal are three-fold; 1. The complexity is essentially independent of bit-rate and dimension, 2. we do not require a component classifier

International Audio Laboratories is a joint institution between Fraunhofer IIS, Erlangen and Friedrich-Alexander University (FAU), Erlangen. This project was supported by the Academy of Finland project 284671.

and overlaps between Gaussian components are taken optimally into account, and 3. quantization is performed in the original domain and not in a transformed domain, for high computational efficiency, and direct control of the quantization accuracy.

2. MULTIVARIATE GAUSSIAN DISTRIBUTIONS

Our objective is to quantize envelope parameters $x \in \mathbb{R}^{N\times 1}$ and transmit the quantized parameters $\hat{x}$ using an entropy coder. We will assume that x can be modelled using a Gaussian mixture model (GMM) with M components such that the probability distribution function is

$$f(x) = \sum_{k=1}^{M} \lambda_k f_k(x), \tag{1}$$

where $\sum_{k=1}^{M} \lambda_k = 1$,

$$f_k(x) = |2\pi\Sigma_k|^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right) \tag{2}$$

and $|\Sigma|$ denotes the determinant of the covariance matrix Σ and μ_k and Σ_k are, respectively, the mean and covariance of the kth Gaussian.

We encode the elements of $x = [\xi_0,\, \xi_1,\, \dots,\, \xi_{N-1}]$ one by one, and use the previously encoded elements as prior information for subsequent elements. In other words, we will use the conditional probability $f(\xi_h \,|\, \xi_0,\, \dots,\, \xi_{h-1})$ to encode the element ξ_h, or more specifically, the corresponding cumulative probability function.

Let us begin with deriving the conditional probability distribution function for a single multivariate Gaussian,

$$f(x) = |2\pi\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right). \tag{3}$$

We can split x into two parts, x_0 and x_1, corresponding respectively to the coefficients which have been already coded x_0 and those still to be coded x_1 as

$$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}, \qquad \text{where} \begin{cases} x_0 = & [\xi_0,\, \dots,\, \xi_{h-1}]^T \\ x_1 = & [\xi_h,\, \dots,\, \xi_{N-1}]^T. \end{cases} \tag{4}$$

The mean and covariance must be split accordingly as

$$\mu = \begin{bmatrix} \mu_0 \\ \mu_1 \end{bmatrix} \qquad \text{and} \qquad \Sigma^{-1} = \begin{bmatrix} A_0 & A_{01} \\ A_{01}^T & A_1 \end{bmatrix}. \tag{5}$$

Here we use the notation A_k for the partition matrices, to emphasise the fact that these are partitions-of-the-*inverse* of the covariance, which is in general different from the inverses-of-the-partitions of the covariance.

With these definitions we obtain the equality

$$(x-\mu)^T \Sigma^{-1}(x-\mu) = (x_1-\hat{\mu}_1)^T A_1 (x_1-\hat{\mu}_1) - c, \tag{6}$$

where

$$\begin{cases} \hat{\mu}_1 = & \mu_1 - A_1^{-1}A_{01}^T(x_0-\mu_0), \\ c = & (x_0-\mu_0)^T \left[A_{01}A_1^{-1}A_{01}^T - A_0\right](x_0-\mu_0). \end{cases} \tag{7}$$

Substituting into Eq. 3 yields

$$\begin{aligned} f(x) &= \frac{e^c}{|2\pi\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x_1-\hat{\mu}_1)^T A_1(x_1-\hat{\mu}_1)\right) \\ &= \frac{e^c|A_1|^{\frac{1}{2}}}{|\Sigma|^{\frac{1}{2}}} \frac{\exp\left(-\frac{1}{2}(x_1-\hat{\mu})^T A_1(x_1-\hat{\mu})\right)}{|2\pi A_1|^{\frac{1}{2}}}. \end{aligned} \tag{8}$$

In other words, when x_0 is known, then x_1 follows the normal distribution with covariance A_1^{-1} and mean $\hat{\mu}$, but such that the probability distribution is scaled with $\alpha = \frac{e^c|A_1|^{\frac{1}{2}}}{|\Sigma|^{\frac{1}{2}}}$.

When the signal is a mixture of multiple Gaussians, as in Eq. 1, then it can be written as

$$f(x) = \sum_{k=1}^{M} \lambda_k \alpha_k f_k(x_1; \hat{\mu}_k, A_{k,1}^{-1}), \tag{9}$$

where $f(x;\mu,C)$ is the multivariate Gaussian of x with mean μ and covariance C, and α_k, $\hat{\mu}_k$ and $A_{k,1}$ are calculated for each component as in Eqs. 4–7. The probability distribution of x_1 can then be obtained by scaling the above expression such that $\sum_{k=1}^{M} \lambda_k\alpha_k = 1$.

In other words, the distribution of x_1 remains a GMM, but the weights and means of each Gaussian component are updated depending on the previously encoded samples. The covariances $A_{k,1}^{-1}$ are dependent only on Σ_k, whereby they can be calculated off-line.

The overall algorithm can then be stated as:

1. Encode the first component ξ_0 using conventional, scalar-valued arithmetic coding [14, 5], where the mean and variance of each Gaussian are the first components of μ_k and Σ_k.

2. For $h = 1$ to $N-1$

   (a) Define covariances and means for each mixture component according to Eqs. 4–7.

   (b) Encode component ξ_h using conventional, scalar-valued arithmetic coding, where the means and variances of each Gaussian are obtained from above.

On each iteration, we need to calculate only the mean, variance and weights of each Gaussian for the current sample, which are essentially vector multiplications of algorithmic complexity $\mathcal{O}(N-h)$, whereby the overall complexity is $\mathcal{O}(N^2)$. However, arithmetic coding requires evaluation of the cumulative distribution of M Gaussian components, at an overall complexity of $\mathcal{O}(MN)$. Though this would typically

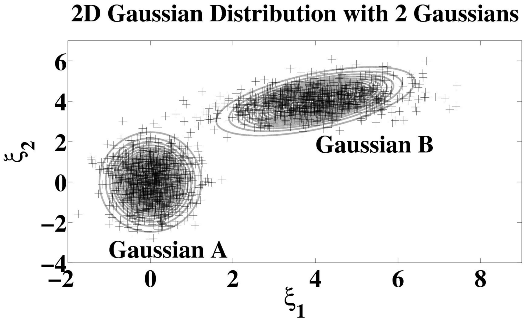

Fig. 1. Illustration of 2D Gaussian mixture model with 2 Gaussians.

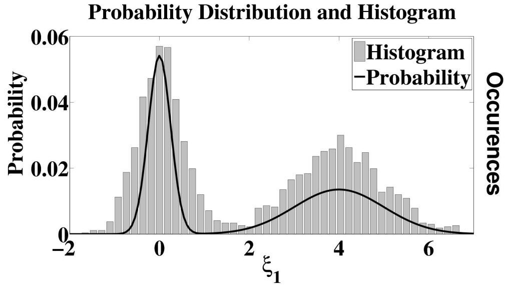

Fig. 2. Histogram and probability distribution model of first sample ξ_1.

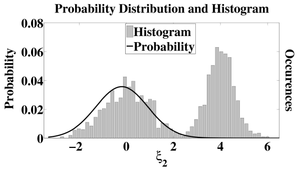

Fig. 3. Histogram and probability distribution model of ξ_2 w.r.t Gaussian A.

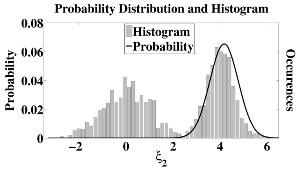

Fig. 4. Histogram and probability distribution model of ξ_2 w.r.t Gaussian B.

be a relatively small number, it has a large constant coefficient, since calculation of the cumulative distribution involves evaluation of the error function, which is a non-elementary special function. To reduce the complexity of evaluating the cumulative distribution, we can approximate it with a distribution of similar shape, like, for example, the logistic distribution, whose cumulative distribution has a trivial form [15].

3. EXPERIMENTS

As an illustration of the algorithm explained in Sec. 2, we consider a two dimensional Gaussian distribution with two components as shown in Fig. 1. Our objective is to encode an observation using the Gaussian mixtures. The first sample of the observation can be directly encoded as explained in the first step of the algorithm, since there are no priors. Fig. 2 shows the 1D histogram for the first parameter along with the marginal distribution of the 2D Gaussian mixture model. With the quantized first observation as a-priori, one can determine the distribution in $N-1$ dimensional space which in our case is 1 dimensional space as explained in the second step of the algorithm. If the first observation lies near the center of Gaussian A, Eq. 9 yields distribution as shown in Fig. 3. The weight assigned to Gaussian B is so small that, for all practical reason, it can be ignored. Similarly if the first observation lies near the center of Gaussian B, the output of Eq. 9 yields distribution as shown in Fig. 4. The weight assigned to Gaussian A is so small that, for all practical reason, it can be ignored.

After this introductory illustration, we can proceed to the evaluate the model with real data. For the experiment, we chose 3 variants of LPC, 2 variants of DQ and 3 variants of scale factor band. The line spectral frequencies (LSFs) is a representation often used for encoding linear predictive models [1, 2, 3]; D-LSF (delta-LSF) are the intra-frame LSF difference and LSF-IS (LSF-Inverse Sigmoid) are computed by normalizing the LSF and then computing the inverse sigmoid. The distribution quantization represents spectral envelopes in terms of energy ratios (DQ-ER) [4] whereas log difference (DQ-LD) of the same segments is a similar measure. Scale Factor Bands (SFBs) are piece-wise constant spectral envelope, and their logarithms (SFB-LD) and inverse sigmoids (SFB-IS) represent alternative parameterizations. The log domain and inverse sigmoid maps the normalized input range [0,1] to input range [0,∞] and [-∞ ∞] respectively.

As a reference for the GMM-based algorithm, we used tree searched multi-stage vector quantizer as described in [10]. To simplify evaluation, we did not to use inter-frame dependencies which is usually done in the design of the vector quantizers [10].

We trained both the VQ and the GMM using the training set of the TIMIT database [16]. The training was based on 689466 vectors. For narrow band (NB), the VQ was designed at 24 and 33 bits. The GMM was trained with 3, 5 and 10 Gaussians. For wideband (WB), the VQ was designed at 36 and 43 bits. The GMM was trained with 5, 10 and 15 Gaussians.

Methods were tested over the test set of the TIMIT database [16] with 25493 test vectors. A rate loop was used such that bit consumption of each frame of our GMM based system (variable bit rate) is comparable to the VQ based reference system (constant bit rate). We used mean log spectral distance (LSD) between original and quantized envelopes as our primary evaluation parameter.

4. RESULTS

Fig. 5 compares the mean log spectral distance (LSD) of all the parameters for VQ and GMM10 at NB at 33 bits and for VQ and GMM15 at WB at 43 bits. From the comparison, it can be concluded that among the LSF variants, LSF per-

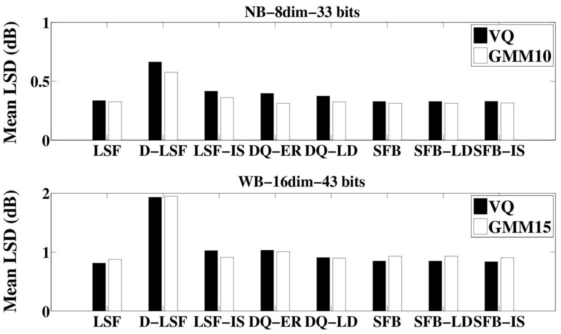

Fig. 5. Mean Log Spectral distance (LSD) (dB) vs all parameters at 33 bits (NB) and 43 bits (WB).

forms better than D-LSF and LSF-IS for VQ, GMM10 and GMM15. DQ-LD also performs slightly better than DQ-ER among the DQ variants. The difference among the SFB variants are minor. Hence, for the further analysis, we chose LSF, DQ-LD and SFB parameters among the 8 available types of model parameters.

Fig. 6 compares the mean LSD for the chosen parameters for all the configurations (VQ and GMM variants) at both NB (24 and 33 bits) and at WB (36 and 43 bits) respectively. The performance is relatively constant at both high and low bitrates for NB and WB respectively. At NB, VQ performs slightly better than the GMM versions for both LSF and SFB parameters but GMM versions perform better than VQ for DQ-LD. At WB, VQ perform slightly better than GMM versions for all the three parameters (LSF, DQ-LD, SFB). The difference between VQ and GMM versions are slightly higher at 36 bits than 43 bits at WB. The difference could be explained by the fact that at low bitrates, the GMM method predicts the probability distribution of future samples from quantized previous samples. The feedback of quantization noise might thus reduce coding efficiency when encoding later samples. Among the Gaussian versions, GMM10 seems to slightly better than GMM3 and GMM5 at NB and GMM15 seems to be slightly better than GMM5 and GMM10 at WB.

Table 1 compares the proportion of outliers for the LSF parameter for VQ and GMM10 at NB at 24 bits and for VQ and GMM15 at 43 bits at WB. The number of outliers and the mean LSD of GMM10 and GMM15 are comparable to VQ at NB and WB respectively. The values of mean LSD are less than 1dB at both NB and WB for the compared configurations. By making use of inter-frame dependencies, one can expect the reduction in the proportion of outliers and mean LSD.

5. CONCLUSION

Modelling and coding the spectral envelopes is an integral part of both audio and speech codecs. In this paper, we have proposed an iterative GMM based entropy coder to encode the spectral envelope parameters. The performance of this proposed system is on par with VQ at both NB and WB conditions with several advantages. The advantages of our iterative GMM based system compared to VQ and previously proposed GMM based systems are: 1. The complexity and the training procedure is independent of both bitrate and vector length. 2. Unlike previous GMM-based coding schemes, the new method does not require decorrelation of the parameters using an extra transformation, which simplifies the design of the system. 3. There is no need for a component classifier.

The performance of iterative GMM based entropy coder was on par with VQ in terms of mean LSD values at both NB and WB for all the bitrates tested. Iterative GMM based entropy coder has lower complexity and is more flexible in comparison to VQ system. This demonstrates that the proposed method provides similar accuracy as VQ based methods, but with higher flexibility and lower complexity, and that the method can be used to replace VQ in any speech and audio coding systems.

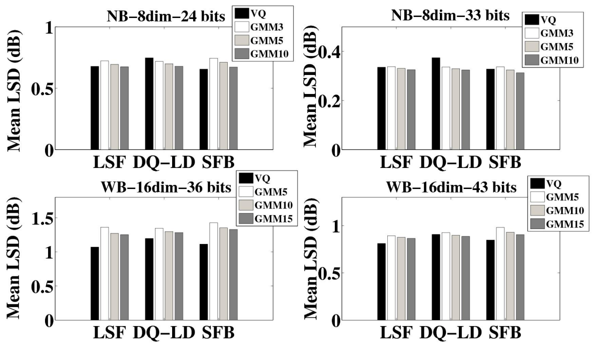

Fig. 6. Mean Log Spectral distance (LSD) (dB) vs all configurations for LSF, DQ-LD and SFB at NB (24 and 33 bits) and WB (36 and 43 bits).

Condition	2-4dB	> 4dB	Bits	SD
VQ-NB-24	3.87 %	0.59 %	24.00	0.68 dB
GMM10-NB-24	3.91 %	0.66 %	24.32	0.67 dB
VQ-WB-43	5.85 %	0.94 %	43.00	0.81 dB
GMM15-WB-43	6.87 %	1.31 %	42.40	0.87 dB

Table 1. Outlier comparison for LSF parameter at 24 bits (NB) and 43 bits (WB).

6. REFERENCES

[1] 3GPP, *TS 26.445, EVS Codec Detailed Algorithmic Description; 3GPP Technical Specification (Release 12)*, 2014.

[2] ISO/IEC 23003–3:2012, "MPEG-D (MPEG audio technologies), Part 3: Unified speech and audio coding," 2012.

[3] 3GPP, *3GPP TS 26.290, Audio codec processing functions; Extended Adaptive Multi-Rate – Wideband (AMR-WB+) codec*, 2007.

[4] S Korse, T Jähnel, and T Bäckström, "Entropy coding of spectral envelopes for speech and audio coding using distribution quantization," in *Proc. Interspeech*, 2016.

[5] Tom Bäckström, *Speech Coding with Code-Excited Linear Prediction*, *in press,* Springer, 2017.

[6] Kuldip K Paliwal and Bishnu S Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," *IEEE Trans. Speech Audio Process.*, vol. 1, no. 1, pp. 3–14, 1993.

[7] A Gersho and R M Gray, *Vector quantization and signal compression*, Springer, 1992.

[8] R Gray, "Vector quantization," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 1, no. 2, pp. 4–29, 1984.

[9] B H Juang and A H Gray, "Multiple stage vector quantization for speech coding," in *Proc. ICASSP*. IEEE, 1982, vol. 1, pp. 597–600.

[10] W P LeBlanc, B Bhattacharya, S A Mahmoud, and Cuperman. V, "Efficient search and design procedures for robust multi-stage VQ of LPC parameters for 4 kb/s speech coding," *IEEE Trans. Speech Audio Process.*, vol. 1, no. 4, pp. 373–385, 1993.

[11] P Hedelin and J Skoglund, "Vector quantization based on gaussian mixture models," in *Proc. ICASSP*. IEEE, 2000, vol. 4, pp. 385–401.

[12] D Y Zhao, J Samuelsson, and M Nilsson, "GMM-based entropy-constrained vector quantization," in *Proc. ICASSP*. IEEE, 2007, vol. 4, pp. IV–1097.

[13] D Y Zhao, J Samuelsson, and M Nilsson, "On entropy-constrained vector quantization using Gaussian mixture models," *IEEE Trans. Commun.*, vol. 56, no. 12, pp. 2094–2104, 2008.

[14] J Rissanen and G G Langdon, "Arithmetic coding," *IBM Journal of research and development*, vol. 23, no. 2, pp. 149–162, 1979.

[15] C Walck, *Handbook on statistical distributions for experimentalists*, University of Stockholm Internal Report SUF-PFY/96-01, 2007.

[16] J S Garofolo, Linguistic Data Consortium, et al., *TIMIT: acoustic-phonetic continuous speech corpus*, Linguistic Data Consortium, 1993.