HIDDEN MARKOV MODELS WITH DIVERGENCE BASED VECTOR QUANTIZED VARIANCES

Jae Kim, Raziel Haimi-Cohen*, and Frank Soong***

*Philips Consumer Communications
330 S. Randolphiville Rd.
Piscataway, NJ 08854

**Lucent Technologies
600 Mountain Ave.
Murray Hill, NJ 07974 USA

ABSTRACT

This paper describes a method to significantly reduce the complexity of continuous density HMM with only a small degradation in performance. The proposed method is noise-robust and may perform even better than the standard algorithm if training and testing noise conditions are not matched. The method is based on approximating the variance vectors of the Gaussian kernels by a vector quantization (VQ) codebook of a small size. The quantization of the variance vectors is done using an information theoretic distortion measure. Closed form expressions are given for the computation of the VQ codebook and the superiority of the proposed distortion measure over the Euclidean distance is demonstrated. The effectiveness of the proposed method is shown using the connected TI digits database and a noisy version of it. For the connected TI digit database, the proposed method shows that by quantizing the variance to 16 levels we can maintain recognition performance within 1% degradation of the original VR system. In comparison, with Euclidean distortion, a size 256 codebook is needed for a similar error rate.

1. INTRODUCTION

The accuracy of hidden Markov models (HMM) based speech recognizer depends largely on the effectiveness of the representation of the observation likelihoods in the model. Continuous Density HMM [1] has shown to be very successful in this respect: The probability density function (PDF) of the observations is modeled by a mixture of Gaussian kernels which can closely approximate the features distribution. Also, Laplacian and other kernels have also been used successfully for this purpose[2]. In this paper we consider only Gaussian kernels; however, the method can be extended to Laplacian kernels as well. The flexibility of Gaussian mixtures comes at a price: Evaluating a large number of Gaussian kernels is computationally expensive; also, the kernel parameters require a large storage space and, since all these parameters need to be access by the processor every frame, fast access memory devices are required. In comparison, in a discrete HMM [3], the discrete probability can be retrieved from a table look-up and no computation is needed. Several attempts have been made to alleviate the problems of a continuous HMM. One approach was to organize the kernels in clusters and evaluate the likelihoods only for those clusters which are not too "far" from the given feature vector [4]. This method significantly reduced computations and memory access, but did not influence memory size. A widely known approach which aims at reducing the number of kernels is the semi-continuous HMM [5] which makes a more efficient use of a limited number of kernels by sharing them among all state mixtures. However, in order to maintain a high spectral resolution, semi-continuous HMM needs to use larger number of mixture densities, which require more storage for mixture weights and much more computations and memory access to obtain the mixtures likelihoods [6]. Recently, an alternative approach was proposed which applied vector quantization to the means and variances of the kernels in order to reduce storage and computation [7]. As it turned out, in order to get an acceptably small quantization error, the kernels parameters vector had to be split into sub-vector and each sub-vector was quantized separately. As a result even though the number of arithmetic operations has been reduced, the computation requires a large number of table look-up operations which may be quite costly, particularly if the recognition is performed on a digital signal processor (DSP) which is not optimized for this type of calculations.

It is also known that the Gaussian kernels are very sensitive to any perturbation in the value of their means. On the other hand, the values of the variances seem to have a much lesser impact on performance. In fact it has been shown that even with one "grand" variance vector for all kernels yields a moderate degradation in recognition accuracy when the training and testing conditions are matched and improves performance under non-matched conditions [8]. The grand variance method is equivalent to vector quantizing the variance vectors of the Gaussian kernels using a codebook of size one. It therefore seems plausible that if a codebook of higher order is used, the degradation effect will diminish.

This paper describes a method of applying vector-quantization to the variances of the kernels of continuous density HMMs. Sec. 2 describes the quantization algorithm, which uses a non-Euclidean error metric in order to account for the non-linear effect that the variances have on the Gaussian kernel likelihood. Sec 3 shows the computational gain achieved by using quantized variances. Sec. 4 gives some performance results using the method and Sec. 5 provides some conclusions.

2. DIVERGENCE BASED VQ

In variance quantization we create a small codebook of *variance centroid vectors* and replace the variance vector in each Gaussian

kernel by a centroid vector from the codebook. Our goal is to minimize the resulting distortion in the PDFs of the mixtures. We use the divergence probabilistic distance as measure for this distortion [9]:

$$d(f,\hat{f}) = \int \left[f(\mathbf{x}) - \hat{f}(\mathbf{x}) \right] \log \left[f(\mathbf{x}) / \hat{f}(\mathbf{x}) \right] d\mathbf{x} \quad (1)$$

where $f(\mathbf{x})$ and $\hat{f}(\mathbf{x})$ are the PDFs of the original and the quantized mixtures, respectively. The divergence distance measure was selected because it measures the distance between log-likelihoods, and the cumulative HMM Viterbi score is the sum of the log likelihoods over the optimal path. However, if $f(\mathbf{x})$ is a mixture of more than one kernel, then the integral in Eq. (1) becomes intractable and one cannot get a closed form solution for the distortion minimization problem. Therefore, instead of minimizing the distortion of the mixtures, we will minimize the distortion of their component Gaussians, again with respect to the divergence distortion measure.

It can be easily shown that if $f(\mathbf{x})$ and $\hat{f}(\mathbf{x})$ are Gaussians with the same mean vectors and diagonal variance vectors $\mathrm{diag}(\sigma_1,\ldots,\sigma_M)$ and $\mathrm{diag}(\hat{\sigma}_1,\ldots,\hat{\sigma}_M)$ respectively, then the divergence between the two Gaussians is given by [9]:

$$d(f,\hat{f}) = \sum_{i=1}^{M} \frac{1}{2} \left[\frac{\sigma_i}{\hat{\sigma}_i} - \frac{\hat{\sigma}_i}{\sigma_i} \right]^2 \quad (2)$$

since the right hand side depends only on the variance vectors, we may consider it as a distance measure between two *M*-dimensional variance vectors, $\underline{\sigma}$ *and* $\underline{\hat{\sigma}}$:

$$d(\underline{\sigma},\underline{\hat{\sigma}}) = \sum_{i=1}^{M} \frac{1}{2} \left[\frac{\sigma_i}{\hat{\sigma}_i} - \frac{\hat{\sigma}_i}{\sigma_i} \right]^2 \quad (3)$$

Given a set of *N* Gaussians and a corresponding set of variance vectors $A = \left\{ \underline{\sigma}_k | k = 1,\ldots,K \right\}$, we compute a codebook $B = \left\{ \underline{\hat{\sigma}}_k | k = 1,\ldots,K \right\}$ of centroid vectors and for each variance vector $\underline{\sigma}_j$, $j = 1,\ldots,N$ we assign a centroid $\underline{\hat{\sigma}}_{k(j)} \in B$. These computations are done iteratively using the Linde-Buzo-Gray (LBG) algorithm [10]. Each iteration consists of two steps:

Step 1: For each Gaussian $\sigma_j \in A$, select the nearest centroid $\underline{\hat{\sigma}}_{k(j)} \in B$ such that

$$k(j) = \arg\min_{k=1,\ldots,K} d(\underline{\sigma}_j, \underline{\hat{\sigma}}_k) \quad (4)$$

Step 2: For each *k=1,…,K* let

$$C_k = \left\{ j \middle| 1 \le j \le N, k = \arg\min_{s=1,\ldots,K} d(\underline{\sigma}_j, \underline{\hat{\sigma}}_s) \right\} \quad (5)$$

be the cluster of all variance vector indices assigned to the *k*-th centroid. For each cluster *k* let the total distortion be:

$$J_k(\underline{\tilde{\sigma}}_k) = \sum_{j \in C_k} d(\underline{\sigma}_j, \underline{\tilde{\sigma}}_k) \quad (6)$$

We compute a new centroid $\underline{\tilde{\sigma}}_k$ which minimizes $J_k(\underline{\tilde{\sigma}}_k)$ by equating the gradient of the right hand side of Eq. (6) to 0. With some manipulations one can get a closed form formula for $\tilde{\sigma}_k$:

$$\tilde{\sigma}_{ki}^2 = \sqrt{\frac{\sum_{j \in C_k} \sigma_{ji}^2}{\sum_{j \in C_k} \sigma_{ji}^{-2}}} \quad \text{for } i=1,\ldots,M \quad (7)$$

After the new centroids are computed, they are copied into the old ones and we go back to step one. This iterative process is guaranteed to converge to a local minimum of the overall distortion.

3. COMPUTATIONAL GAINS

It is obvious that quantizing the variance vectors will reduce the required storage. The following shows how the quantization may be used to reduce computations. Let $N(\underline{x};\underline{\mu},\mathbf{C})$ be a Gaussian kernel with a mean vector $\underline{\mu}$ and a covariance matrix $\mathbf{C} = \mathrm{diag}(\sigma_1,\ldots,\sigma_M)$ and let $\mathbf{D} = \mathbf{C}^{-1/2}$ .Then the log likelihood of this kernel for a given feature vector $\underline{x}$ is given by:

$$\log N(\underline{x};\underline{\mu},\mathbf{C}) = G + \left(\underline{x} - \underline{\mu}\right)^T \mathbf{C}^{-1} \left(\underline{x} - \underline{\mu}\right) = G + \left\| \mathbf{D}\underline{x} - \mathbf{D}\underline{\mu} \right\|^2 \quad (8)$$

Where *G* is a constant. $\mathbf{D}\underline{\mu}$ is independent of $\underline{x}$ hence it may be pre-computed and stored. $\mathbf{D}\underline{x}$ is a vector multiply operation which needs to be computed once for each variance vector. Therefore, reducing the number of variance vectors will correspondingly reduce the number of vector multiplies required.

4. EXPERIMENT RESULTS

The variance quantization method was tested on two databases: The Texas Instruments database of American English digit strings (TIDIGITS) and a noisy version of that database, which was obtained by adding car noise at various SNR-s to TIDIGITS [11]. The feature vector in all experiments included 25 features: 12 cepstrum coefficients, 12 delta-cepstrum coefficients and the delta-energy coefficient. Continuous density HMM models were generated from the training part of both clean and the noisy databases. All 11 word models had 8 states with 8 mixtures and the silence model has 1 state with 5 mixtures. Altogether, there are 709 Gaussian kernels for the 12 models. The variances of the models were quantized in the method described in sec. 2 (except where otherwise specified). Recognition experiments were conducted with an unknown string length assumption (except where otherwise indicated).

4.1 Divergence vs. Euclidean Distortion

The performance variance quantization based on the divergence distance was compared to the performance of variance quantization using Euclidean distance between variance vectors.

Models were trained on clean TIDIGTS and testing was also on clean TIDIGITS. The results are shown in Fig. 1 as a function of the codebook size. The total number of connected string used in this experiment is 8,700 utterances (28,583 digits), average digit length per string is 3.29 and the baseline accuracy (without quantization) was 95.6%.

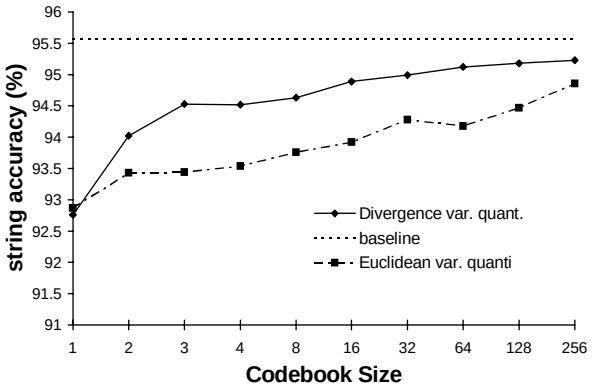

Figure 1: Comparison of two quantization methods

The performance of divergence based variance quantization (VQD) system is better than that of the Euclidean distance based variance quantization (VQE) system for all codebook sizes except codebook size 1 where VQE is negligibly better. In both methods the accuracy increases with the size of the codebook. However, with VQE the performance improves very quickly and very little is gained beyond a codebook size of 8 or 16. On the other hand, in VQD, the improvement is rather slow and we need to go to a codebook of size 256 in order to get performance similar to that of VQE at codebook size 16.

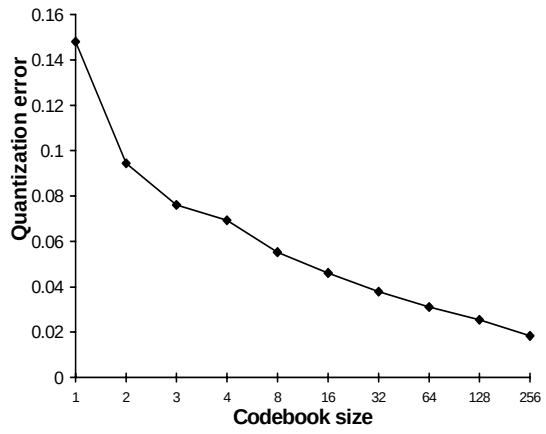

Figure 2: Quantization error vs. codebook size

4.2 Quantization Error and Accuracy

The divergence quantization error (the average distortion of variance vectors) with respect to codebook size has been plotted in Figure 2. As we expected, its value decreases as the codebook size increases. Comparing fig. 1 to fig. 2 we can see that an improvement in performance as the codebook size is increased by one step is highly correlated with a corresponding reduction of quantization error.

4.3 Performance vs. Digit String Length

Figure 3 represents the performance with respect to string length on the clean training/clean testing of the TIDIGTS. The performance of codebook sizes of 1, 8, and 256 are compared with that of the unquantized system. As we can see the performance of codebook sizes 8 and 256 is much better than that of a single variance and quite close to the unquantized performance.

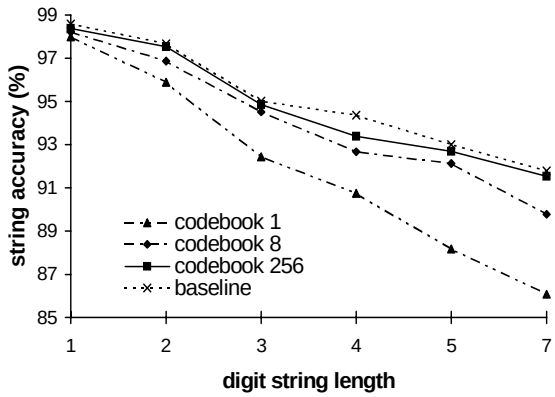

Figure 3: Performance of variance quantization w.r.t string size on the clean/clean connected TIDIGITS

4.4 Performance on Noisy Testing Data

We ran two experiments on noisy TIDIGITS test data. In the first experiment training was performed on the training parts of the noisy TIDIGITS database. In this case the test was run for both known string length and unknown string length. The results are shown in Figure 4. The performance of the system with quantization approaches the baseline system (unquantized system) with increasing codebook size, however, the convergence rate is slower than in the clean train/clean test case and we need a codebook of order 16-32 in order to reach the flat part of the curve. The performance on the known length case is better, as expected, but its behavior as a function of codebook size is similar to that of the unknown length.

The second experiment tested performance in conditions of mismatch between testing and training. The training was performed on the clean database while the testing was performed on the noisy one. The results are shown in Figure 5. It appears that under mismatch conditions the relative degradation caused by variance quantization is very small and for codebook sizes of 16 and above the quantization actually improves performance, reaching the peak codebook sizes of 64 to 128.

The reason that the quantized version performs better than the unquantized is probably the same reason that the grand variance improved noise robustness: “over-training” of the variances in the clean database, created some strange “outlier” variances

which were ineffective in the noisy environment. The quantization process probably mapped those "outliers" to better behaved centroids, thus actually improving performance in noise.

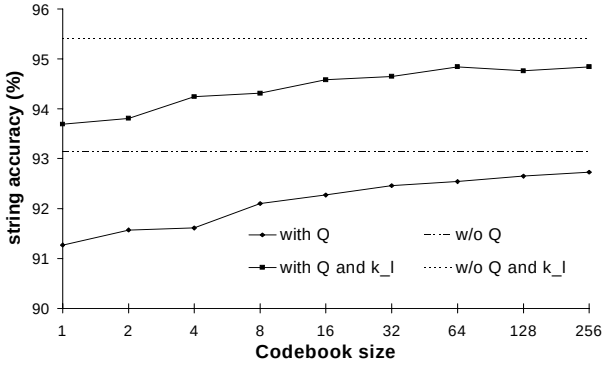

Figure 4: Performance of variance quantization with noisy train/noisy test. k_l =known-length, with Q and w/o Q = with and without variance quantization.

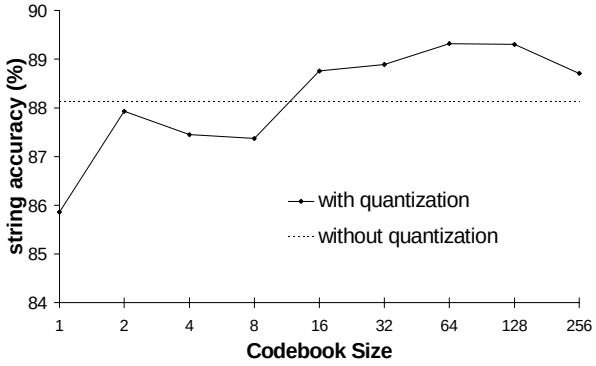

Figure 5: Performance of variance quantizaion in clean train/noisy test conditions.

5. CONCLUSIONS

This work shows that continuous density HMM is quite robust to divergence based quantization of the variance vectors. In some cases vector quantized variance may even improve recognition result. We have also shown that the proper choice of distortion measure gives the robustness in performance. This proposed divergence based vector quantization should be equally applicable to other methods that rely on a quantization of the mixture kernels.

Quantizing variances can lead to significant savings in storage, memory access and computations. In a system with limited resources, this option is highly desirable.

6. REFERENCES

[1] L. Rabiner, B. Juang, S. Levinson, and M Sondhi, "Recognition of Isolated Digits Using Hidden Markov Models With Continuous Mixture Densities," *AT&T Technical Journal*, Vol. 64, July-August 1985.

[2] S. Euler and D. wolf, "Continuous hidden Markov models in speakeer independent isolated word recognition" in J.L. Lacoume et al. (eds.) *Signal Processing: Theory and Applications*, vol. 3, pp. 1185-8, 1988.

[3] K. Lee, "Automatic Speech Recognition-The Development of the SPHINX System." *Kluwer Academic Publishers*, 1989.

[4] E. Bocchieri, "Vector Quantization For The Efficient Computation of Continuous Density Likelihoods," *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing, ICASSP-93*, Vol. 2, pp. 692-695, 1993.

[5] X.D. Huang and M.A. Jack, "Semi-continuous hidden Markov models for speech signals," *Computer Speech and Language*, vol. 3, pp. 239-251, 1989.

[6] S.K. Gupta, F.K. Soong and R. Haimi-Cohen, "Speaker-independent recognition for mobile applications using tied - mixture HMM," *Proc. IEEE Int. Conf. Spoken Language Understanding, ICSLP-96*, vol. 3, 1996.

[7] M. Ravishankar, R. Bisiani, and E. Thayer, "Sub-Vector Clustering To Improve Memory and Speed Performance of Acoustic Likelihood Computation," *Eurospeech*, Vol. 1, pp. 151-154, 1997.

[8] R.P. Lippmann, E.A. Martin and D.B. Paul, "Multi-style training for robust isolated-word speech recognition," *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing - ICASSP-87*, pp. 692-695, 1987.

[9] P.A. Devijver and J. Kittler, *Pattern Recognition, A Statistical Approach*, ch. 7, Prentice Hall, London 1982.

[10] Y. Linde, A. Buzo and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Comm.*, vol. COM-26, pp. 702-710, April 1980.

[11] S.K. Gupta, Frank Soong and Raziel Haimi-Cohen, "High accuracy connected digit recognition for mobile applications," *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing - ICASSP-96*, vol. 1, pp. 57-60, 1996.