An efficient codebook design in SDCHMM for mobile communication environments

Gue Jun Jung, Su-Hyun Kim and Yung-Hwan Oh

Division of Computer Science
Department of Electrical Engineering and Computer Science, KAIST
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea
{sylph, ksh, yhoh}@speech.kaist.ac.kr

Abstract

Speech recognition systems require too much memory to run and are too slow for mass application. In order to overcome these constraints and make speech recognition systems suitable for mobile devices, we propose efficient codebook construction method for subspace distribution clustering hidden markov modeling (SDCHMM). The output probability of mixture Gaussians is more sensitive to quantization error of mean vectors than that of variance vectors. Therefore we propose a new subspace definition which minimizes quantization error of mean vectors first. Next, we split mixture Gaussians into mean and variance vectors and construct separate codebooks using modified Bhattacharyya distance measure. In experiments using RM database, proposed method decreases 24.5% relative word error rate compared with general SDCHMM without use of extra memory.

1. Introduction

Currently the most widely used statistic probability model for Automatic Speech Recognition (ASR) is continuous hidden Markov model (CHMM). This provides very high recognition accuracy but requires too much training data and memory. To use this statistical model in mobile devices, it is necessary to reduce the memory for model representation and the number of parameters in acoustic models [1]. The most common approach to reduce parameters is tying. The technique of parameter tying has been applied successfully at various granularities [2]-[4].

In various techniques, subspace distribution clustering hidden Markov modeling (SDCHMM) achieved great data compression and maintained good resolution for acoustic models [4]. In SDCHMM, distributions of CHMM are projected onto orthogonal subspaces and similar subspace distributions are tied into a small number of distribution prototypes over all states in CHMM [4]. In this technique, how to project distributions is an important issue. In previous works, it was defined by a simple and coherent definitions [4][5]. However for more efficient model representation, it is necessary to consider relation of vector quantization errors and recognition performance.

In this paper, we define a new subspace definition for SDCHMM, which consider vector quantization error and error sensitivity of the output probability in mixture Gaussians and split codebook of general SDCHMM. We also devise distance measures and codeword update equation for split codebook vector clustering. The outline of the paper is as follows: Section 2 deals with concept of SDCHMM and its subspace definitions. Section 3 addresses our codebook construction method for improving SDCHMM and proposed method is evaluated in Section 4 on the RM task. Finally, we draw our conclusions in Section 5.

2. SDCHMM

In SDCHMM, mixture Gaussians with diagonal covariance are first projected into low-dimensional (usually one to three dimensional) subspaces and subsequent subspace Gaussians are tied [4]. Because the efficiency is achieved by clustering in low dimensions, SDCHMM attains a high degree of tying without degradation in performance when compared to its original CDHMM. The state output probability with $M$ Gaussians can be rewritten as follows:

$$P(O) = \sum_{m=1}^{M} c_m N(O; \mu_m, \sigma_m^2)$$

$$= \sum_{m=1}^{M} c_m \left( \prod_{k=1}^{K} N(O_k; \mu_{mk}, \sigma_{mk}^2) \right)$$

$$\approx \sum_{m=1}^{M} c_m \left( \prod_{k=1}^{K} N^{tied}(O_k; \mu_{mk}, \sigma_{mk}^2) \right)$$

(1)

where $K$ is the number of subspaces which is determined by its definition and tied means clustering the subspace Gaussians into a small number of Gaussian prototypes in each subspace.
2.1. Subspaces Definition

For deriving SDCHMM with K subspaces, there are two kinds of heuristic approaches to obtain a reasonable partition [5].
1) Common Subspace : This definition puts conceptually similar features into the same subspace like commonly used in discrete HMM and semi-continuous HMM.
2) Correlated-Feature Subspace : In this definition each subspace is composed of the most correlated features.

The criterion to measure for multiple correlations among features is given by,

\[
R = 1 - \left( \begin{array}{cccc}
1 & \rho_{12} & \rho_{13} & \cdots & \rho_{1k} \\
\rho_{21} & 1 & \rho_{23} & \cdots & \rho_{2k} \\
\rho_{31} & \rho_{32} & 1 & \cdots & \rho_{3k} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\rho_{k1} & \rho_{k2} & \rho_{k3} & \cdots & 1 \\
\end{array} \right) \tag{2}
\]

where \( \rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \) is Pearson’s moment product correlation coefficient.

2.2. Subspace Gaussian Clustering

Subspace Gaussians are clustered by a bottom-up agglomerative clustering scheme. Since the entities to be clustered are Gaussians, Bhattacharyya distance in equation (3) is more suitable than Euclidian distance [4].

\[
D(N_1, N_2) = \frac{1}{8} (\mu_2 - \mu_1)^T \left( \frac{\Sigma_1 + \Sigma_2}{2} \right)^{-1} (\mu_2 - \mu_1) + \frac{1}{2} \ln \left( \frac{\sqrt{\det(\Sigma_1)} \sqrt{\det(\Sigma_2)}}{\sqrt{\det(\Sigma_1 + \Sigma_2)}} \right) \tag{3}
\]

where \( N_1 = N(\mu_1, \Sigma_1) \) and \( N_2 = N(\mu_2, \Sigma_2) \) are Gaussian distributions to be compared.

3. Efficient codebook design

3.1. Feature subspace with minimum quantization error

When subspace Gaussians are clustered, the distortion of HMM is occurred. To construct efficient codebook, quantization errors in Gaussian clustering should be minimized. Assume that set A is \( \{a_1, a_2, \cdots, a_k\} \) and set B is \( \{b_1, b_2, \cdots, b_k\} \). If these are distributed like Figure 1, vector quantization errors of set A is smaller than those of set B. That is to say, vector quantization errors are minimized in case that all the elements of each vector dimension are evenly distributed. In addition, the output probability of mixture Gaussians is more sensitive to error of mean than that of variance [6]. Therefore, in dividing subspaces, we consider to reduce quantization errors of mean vectors. For measuring distribution similarity of \( i \)th and \( j \)th dimensions in mean vectors, the following criterion is used:

\[
D_{\text{sim}}(i, j) = \log \frac{\tilde{\sigma}_i}{\tilde{\sigma}_j} \tag{4}
\]

where \( \tilde{\sigma}_i \) is variance of \( i \)th dimension in mean vectors of whole Gaussians and \( D \) is the number of dimensions in full feature vector.

There is some spread or variation in each dimension. The mathematical concept of variance is an attempt to give a formal measure of this spread. Therefore the similarity of two variances can be measured by ratio of two values. To preserve the constraints of the distance measure, we change measured value to log-scale and get its absolute value. The method to get newly devised subspaces is shown by following steps.

step1) calculate variance \( \sigma_d \) of each dimension in mean vectors. \((1 \leq d \leq D)\)

step2) make k clusters by agglomerative clustering scheme based on \( \log \tilde{\sigma}_1, \log \tilde{\sigma}_2, \cdots, \log \tilde{\sigma}_D \) using equation (4).

step3) define results of step 2) as k subspaces.

This method determines the dimension of each subspace dynamically and requires much less computation than correlated-feature subspace method.

3.2. Split codebook construction

In the aspect of vector quantization errors, it is more profitable to split Gaussians into mean and variance vectors and to construct separate codebooks [7]. However, this increases the amount of required memory for storing acoustic models. By using error sensitivity of output probability in Gaussian distribution, the codebook for variance can be shrunked and overhead memory can be reduced.

3.2.1. distance measures in split codebook

In clustering, proper distance measure constructs more efficient codebook.
1) Euclidian distance measure: It is most general distance measure. In this case split codebook construction is the same as general vector quantization. Distance measures for mean and variance are given by,

\[ D_{\text{mean}}(\mu_1, \mu_2) = (\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2) \]  
\[ D_{\text{var}}(\sigma_1, \sigma_2) = (\sigma_1 - \sigma_2)^T (\sigma_1 - \sigma_2) \]  

2) modified Bhattacharyya distance measure: Euclidian distance measure can’t capture the second order statistics. Therefore, statistical characteristic is not reflected in output codewords for variance. To overcome this problem, we modify Bhattacharyya distance measure as below,

\[ D_{\text{mean}} = \frac{1}{8} (\mu_2 - \mu_1)^T \Sigma^{-1} (\mu_2 - \mu_1) + \frac{1}{2} \ln \frac{1}{\sqrt{2 \pi e}} \]  
\[ D_{\text{var}} = \frac{1}{2} \ln \frac{|\Sigma_1 + \Sigma_2|}{\sqrt{|\Sigma_1||\Sigma_2|}} \]

In equations we assume that all variances are the same for the codebook of mean vectors and all means are the same for that of variance vectors.

3.2.2. codeword update using Gaussian weights

1) codeword update for mean vectors

Suppose that \( \mu_{ik} (1 \leq i \leq N) \) is the mean vector of \( k \)th subspace Gaussians which share \( l \)th codeword and \( \hat{\mu}_l \) is \( l \)th codeword of \( k \)th subspace. In K-means algorithm \( l \)th codeword is updated as follows:

\[ \hat{\mu}_l = \frac{\mu_{1l} + \mu_{2l} + \cdots + \mu_{Nl}}{N} \]

In CHMM, each Gaussian has its weight. If we assume that all training data are uniformly distributed in each state, the weight indicates the influence of each Gaussian. By applying this assumption, equation (9) can be modified as below:

\[ \hat{\mu}_l = \frac{\sum_{i=1}^{N} w_i \mu_{il}}{\sum_{i=1}^{N} w_i} \]  

where \( w_i \) is the weight of \( i \)th Gaussian.

2) codeword update for variance vectors

The same idea is applied in the codeword for the variance.

\[ \hat{\sigma}_{lk}^2 = \frac{\sum_{i=1}^{N} w_i \sigma_{ilk}^2}{\sum_{i=1}^{N} w_i} \]

where \( \sigma_{ilk}^2 \) is the variance vector of \( k \)th subspace Gaussians which share \( l \)th codeword and \( \hat{\sigma}_{lk}^2 \) is \( l \)th codeword of \( k \)th subspace.
Finally we compared integrated system performances in table1 and table2. In the aspect of word error rate, SDCHMM with proposed codebook gave good performances but required extra memory for storing models. To overcome this matter, we reduced the size of variance codebook to balance between the word error rates and the amount of memory usage.

**5. Conclusions**

In this paper we have proposed an efficient codebook design for SDCHMM, which minimize quantization error of Gaussians. Vector quantization errors have been occurred when subspace Gaussians were clustered. To reduce these errors efficiently, we designed a new subspace definition and split Gaussians into mean and covariance. From various experiments, proposed codebook achieved 24.5% improvement of recognition accuracy without extra memory usage. It shows that the proposed codebook design is efficient and well-matched in SDCHMM. In the future, constructed codebook should be adapted to improve the environmental robustness.

**6. Acknowledgements**

This research was supported by University IT Research Center Project.

**7. References**


