Title: Description of a Data Compression Algorithm for Fast Quantization Estimation

by

Author: Xavier Menéndez-Pidal

Technical Report ASEL/Speech/01b-96

asel

Applied Science & Engineering Laboratories
University of Delaware/A.I. duPont Institute
1600 Rockland Rd., P.O. Box 269
Wilmington, DE 19803
Phone: (302)651-6830 FAX: (302)651-6895

INDEX

Description of a Data Compression Algorithm for Fast Quantization Estimation

Xavier Menéndez-Pidal

I. Abstract

The purpose of this internal report is to explain a compression algorithm developed to reduce the computational cost of a Mahalanobis Vector Quantizer. The algorithm has been implemented to reduce the time of the quantization process in around 50% and to compress the spectral coefficients in a controlled and automatic way. The mission of the algorithm is double, it tries to get some benefits of the cuadratic distances, like the Mahalanobis distance to:

-1. Eliminate the difference in the variances of the coefficients in the pre-processing stage to reduce the cost of quantization process.

-2. Compress the original float spectral coefficients in a short integer maximizing the use of the short size so the lost of accuracy due to the compression process is minimized. The only restriction we have imposed to the system is that the Mahalanobis distance must be estimated in integer arithmetic without overflow to optimize the quantization process too.

I.1. Introduction

A vector quantizer is usually composed by a collection of mean vectors also called centroids which try to identify the different acoustic properties existing in the input space to analyze. In a vector quantizer a distance measure is used to estimate what is the nearest centroid of an input vector. The most common distance measure used in speech recognition is the Mahalanobis distance. The Mahalanobis distance is an adequate distance measure to fit an input space when the different components are independents. This last condition is more or less reached with the ceptral coefficients. Using the mahalanobis distance a vector quantizer can be represented by:

1) The n-dimensional input space vector Zj:

$$Zj = (Zj(0), \ldots, Zj(n)) \qquad (1)$$

2) The centroid Ci of the VQ. The k-component of the centroid Ci can be calculated as the mean of the k-component of all the Zj input vectors pertaining to Ci:

$$Ci(k) = \frac{\sum_{j \in i}^{Ni} Zj(k)}{Ni} \qquad (2)$$

3) The vector of variances v(k)= δ^2 (k) of all the input space.

4) The distance measure between the input vector Zj and the centroid Ci:

$$Mahalanobis(Ci, Zj) = D(Ci, Zj) = \sum_{k}^{n} \frac{(Ci(k) - Zj(k))^2}{v(k)} = \sum_{k}^{n} \left(\frac{Ci(k) - Zj(k)}{\delta(k)} \right)^2 \qquad (3)$$

During the compression of the database a transformation is applied to the input space to optimize the cost of the quantization process and to reduce the size of the data. But, the transformation applied must not alter the acoustic characteristics of the input space and must not distort the distance measure. The requirements established for the compression of the input space can be satisfied, in an automatic way taking advantage of some mathematical & statistical properties:

1) The components of the Mahalanobis distance are affected by a proportional factor when an linear transformation of the input space is performed. This property is used to transform the float input space in an short output space equivalent where the original Mahalanobis distance is reduced to a simple integer Euclidean distance which is faster to compute, and to compress the size of the float coefficients.

2) The centroids of a training set are always inside the bounds of the training set. The maximum distance reached by any couple of points (X,Y) pertaining to the input space can be estimated using the limits of input space. This property is used to established some conditions which guarantee an overflow in the distance estimation is never produced.

Experimentally it is known that at least 10 to 11 bits are required to compress linearly the speech coefficients without lost of accuracy. For example if we are working with log energy bands with values expressed in decibels the typical range value for the coefficients vary between 20~120 dbs. For this coefficients an estimation with a precision of 0.1 db (~10 bits) is enough to have a correct representation of the energy bands without any lost of accuracy. For the ceptral coefficients a resolution of 10 to 11 bits is also enough as the ceptral coefficients can be estimated as a linear combination of log energy bands. The time derivative coefficients also can be compressed in 11 bits as they are also estimated as a linear combination of ceptral coefficients. These considerations made the short integer of 2 bytes (16 bits) a variable with enough capacity to store speech coefficients without lost of accuracy if the compression algorithm adopted uses linearly at least 11 bits (+- 1000 values). In the algorithm proposed the speech coefficients are linearly compressed in an integer using 13 to 14 effective bits (+- 8000), values which clearly satisfied there are not lost of precision during the compression stage.

I.2. The Compression algorithm

The compression algorithm adopted requires the estimation of three parameters per coefficient and it transforms the k-component of an input vector Zj in following way:

$$Zjsh(k) = \frac{h(Zjfl(k) - a(k))}{m(k)} \qquad (4)$$

where:

Zjfl(k) is the k-component of the float input vector Zj

Zjsh(k) is the k-component of the short output vector Zj

m(k) is a multiplicative constant which depend on the k-component

a(k) is an additive constant which depend on the k-component

h is a multiplicative compression factor and it is the same for all the k-components

The three parameters introduced (**m(k)**, **a(k)** & **h**) affect the Mahalanobis distance in a different way. The final purpose is to simplify and to convert the float Mahalanobis distance into an equivalent distance simpler to compute, and to compress the input space.

1) The multiplicative factor **m(k)** is made equal to the standard deviation of each coefficient **δ(k)** to suppress the explicit use of the variances reducing the Mahalanobis distance in an Euclidean distance (see eq.5 &6):

$$Dint(Cish, Zjsh) = \sum_{k}^{n}(Cish(k)-Zjsh(k))^2 = h^2 Mahalanobis(Cifl, Zjfl) \qquad (5)$$

2) The additive factor **a(k)** has no effect on the distance measure, it shifts the input space but also it shifts the centroids and the final effect of a(k) is canceled in the distance measure (see eq. 6):

$$\sum_{k}^{n}(Cish(k)-Zjsh(k))^2 = \sum_{k}^{n}\left(\frac{1}{Ni}\sum_{t \in i}^{Ni} h\left(\frac{Ztfl(k)-a(k)}{\delta(k)}\right) - \frac{h(Zjfl(k)-a(k))}{\delta(k)}\right)^2 = h^2\sum_{k}^{n}\left(\frac{Cifl(k)-Zjfl(k)}{\delta(k)}\right)^2 \qquad (6)$$

The **a(k)** factor is used to center the input space in the origin of the short output space in order to maximize the use of the short area. Two centering methods can be adopted:

Geometric centering, this method has provided the better results and a(k) is made equal to the geometric center of each coefficient estimating **a(k)** as the mean of the maximum and minimum limits of the training database (eq. 7):

$$a(k) = \frac{\underset{t}{Min}\, Ztfl(k) + \underset{j}{MAX}\, Zjfl(k)}{2} \qquad (7)$$

Statistical centering, where the a(k) is made equal to the center of the input space estimating **a(k)** as the mean of the training database (eq. 8). This centering method can be adopted to perform a cepstral mean substraction to eliminate the linear distorsions of the transmission chanel existing in the training database.

$$a(k) = \frac{\sum_{j}^{N} Zjfl(k)}{N} \qquad (8)$$

3) The multiplicative compression factor h has two roles:

-The h factor is introduced to establish some conditions which guarantee an overflow in the distance measure is never produced.

-The h factor expands the coefficients in the short area and it reduces the lost of accuracy produced by the compression algorithm.

The first requirement implies the h factor is hardware dependent, and dependent on the variable used to store the distance measure. In the implementation adopted the distance measure is stored in an unsigned integer of 4 byte with the following a ranges:

$$0 \le Dint(Cish, Zjsh) \le MaxUint = (2Maxshort)^2 = 2^{32} \qquad (9)$$

We can use this relation to determine the maximum range of the short output space to avoid overflow in the distance measure. Two considerations are assumed to establish the maximum range of the short output space: The centroids are inside the output short space, and the limits of the short output space are determined if all the components of a vector exceed the limits of the output short space. With the above considerations the maximum range of the coefficients in the short space which avoid overflow in the distance estimation can be estimated using the following approximations:

$$Dint(Cish, Zjsh) \le \sum_{k}^{n} \left(\underset{i}{Max} |Cish(k)| + \underset{j}{Max} |Zjsh(k)| \right)^2 \le \sum_{k}^{n} \left(2 \underset{j}{Max} |Zjsh(k)| \right)^2 \le n(2Maxlimit)^2 \qquad (10)$$

Using eq. 9 and eq. 10 the values of the coefficients in the short output space must be limited by the following bounds to avoid overflow:

$$-Maxlimit \le Zjsh(k) \le Maxlimit = \frac{Maxshort}{sqrt(n)} \qquad (11)$$

This condition implies we can not use all the short area (16 bits) to compress the input coefficients. For example, if we are using 14 ceptral coefficients per VQ, the short coefficients must be compressed in approximately 14.1 bits (+- 8700~32768/sqrt(14)) to avoid overflow in the distance estimation.

Those conditions are used to calculate the h compression factor which linearly expands the coefficients in the above bounds using the following formula:

$$h = \frac{Maxlimit}{\underset{t}{MAX} \left(\frac{\underset{i}{MAX} |Zifl(t) - a(t)|}{\delta(t)} \right)} \qquad (12)$$

This formula is estimated in the training set and it guarantees the short coefficients never produce an overflow in the distance estimation for the training set, as the Zjsh k-components of the output short vector Zjsh are always in the short area desired (see eq.13).

$$-Maxlimit \le Zjsh(k) = \frac{Maxlimit}{\underset{t}{MAX}\left(\frac{\underset{i}{MAX}|Zifl(t)-a(t)|}{\delta(t)}\right)} \cdot \frac{Zjfl(k)-a(k)}{\delta(k)} \le Maxlimit \qquad (13)$$

To avoid problems in the testing set the real h factor is divided by 1.5 reducing some the effective range of all the coefficients in the training set. Also, all the possible coefficients with values higher to Maxlimit are limited to the Maxlimit value. With those last restrictions the algorithm compress on average the training data using 13.2 bits (+- 4700) and allows values in the range +-8700 (14.1 bits) for the coefficients estimated in any other set.

I.3. Use of the Compression Algorithm in other HMM systems

The above algorithm which has been introduced to reduce the cost of estimation of a hard Vector Quantizer in a DHMM can be also adopted to simplify a Semi continuous hidden Markov system (SCHMM) or a continuous HMM (CHMM) system. The two systems (SCHMM & CHMM) generally are both based on the use of mixtures of gaussian density functions to estimate the probability of an acoustic input vector. In those systems the probability obtained in the i-Gaussian pdf is estimated using the following formula:

$$F(Zjfl, i) = N(Zjfl, \mu i, \Sigma i) \qquad (14)$$

where:

Zjfl is the float input vector Zj

Σi is the matrix of covariance of the pdf i

μi is the mean vector of the pdf i

The first simplification usually adopted in the ASR systems is the assumption that the coefficients of the input space are uncorrelated to only use a diagonal covariance matrix. In this case the eq. 14 is computed in the following way.

$$F(Zjfl, i) \;=\; K(n, \delta i)\exp\left(-\frac{1}{2}\sum_{k}^{n}\left(\frac{Zjfl(k)-\mu i(k)}{\delta i(k)}\right)^{2}\right) \qquad (15)$$

If we perform a second assumption that the variances for the different pdf are the same, the algorithm introduced can also be used to reduce the number of parameters and to optimize the computational of the system. Experimentally, in some ASR systems (the CSR system used in Limsi-Philips) it has been shown that the accuracy is degraded very lightly when using only a global vector of variances. In these simplified systems the number of coefficients are reduced approximately by 2. When using only a global vector of variances the cost of computation can also be reduced by 2 computing the cuadratic error measure for all the pdf in a similar spirit as it has been explained for a hard Vector Quantizer system.

I.4. Main References:

Huang X.D., Ariki Y., Jack M.A., “Hidden Markov Models for Speech Recognition”, Edinburgh university Press, 1990.

Linde Y., Buzo A., and Gray R.M., “An Algorithm for Vector Quantizer Design,” *IEEE Trans. Comm.,* 28, 1, pp. 84-95, 1980

Rabiner L., “A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, Proceedings of IEEE, Vol. 77, No. 2, Feb-1989.