FALA 2010
VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop

# Predictive vector quantization using the M-algorithm for distributed speech recognition

*Jose Enrique Garcia, Alfonso Ortega, Antonio Miguel, Eduardo Lleida*

Communications Technology Group (GTC)
Aragon Institute for Engineering Research (I3A), University of Zaragoza, Spain
jegarlai,ortega,amiguel,lleida@unizar.es

## Abstract

In this paper we present a predictive vector quantizer for distributed speech recognition that makes use of a delayed decision coding scheme, performing the optimal codeword searching by means of the M-algorithm. In single-path predictive vector quantization coders, each frame is coded with the closest codeword to the prediction error. However, prediction errors and quantization errors of future frames will be influenced by previous quantizations, in such a way that choosing an instantaneous coding with the best codeword for each frame do not offer the optimal codeword sequence. The M-algorithm presents the advantage of obtaining a global minimization of the quantization error by maintaining the M-best quantization hypotheses for each frame, in a multipath coding approach outperforming the single-path predictive vector quantizer. In this work, the chosen cost function is the Euclidean distance between the sequence of prediction errors and the sequence of quantized values. The method has been tested for coding MFCC coefficients in Distributed Speech Recognition systems, making use of a non-linear predictive vector quantization on a large vocabulary task. Experimental results show that using this global optimization, lower bit rates can be achieved than using the single-path coding non-linear predictive vector quantizer without degradation in terms of WER.

**Index Terms**: distributed speech recognition, predictive vector quantizer, delayed decision coding, M-algorithm

## 1. Introduction

Distributed Speech Recognition (DSR) is the paradigm in which high performance automatic speech recognition applications (ASR) can be developed, releasing more expensive computing resources in the client side. A DSR system is composed of two main modules, the client or user module, where speech acquisition, feature extraction and feature compression are performed, and the server or recognition module, where both feature decompression and ASR decoding are carried out. DSR is usually the solution adopted when the client computing capability is limited, as it occurs in mobile devices, or just for releasing memory and processing resources in the client side, as it occurs in speech-enabled web browsing applications [1]. The main feature of DSR is that low bit-rate compression algorithms can be used without degrading the recognition accuracy. Another option for performing ASR in a client-server architecture consists of sending coded speech instead of coded acoustic features, which is known in the literature as Network Speech Recognition (NSR). However, several studies have shown that the performance is drastically reduced using state of the art speech codecs at low bit-rate conditions [1] [2]. The main reason is that most speech coding algorithms are designed for maximizing speech perceptual quality, not for maximizing speech recognition performance. Because the available network bandwidth is a scarce resource, it is convenient to use compression algorithms that provide transmission rates as low as possible, provided that recognition rates are not reduced.

Differential Vector Quantization (DVQ) is a compression method that exploits both inter-frame and intra-frame mutual information, existing in feature vectors (e.g. MFCC). On the one hand, temporal correlation between adjacent frames, due to both, the overlapping of the windowing step and the relatively slow variation of speech production, is exploited by means of linear prediction. On the other hand, intra-frame redundancy is exploited by means of Vector Quantization. A previous work [3] concluded that using DVQ in a connected digit task, a bit-rate as low as 2.1 kbps could be reached obtaining the same recognition performance than without quantization, while traditional VQ methods obtained a poor recognition performance at bit-rates lower than 3.5 kbps, even worse when noisy channels were evaluated.

The differential vector quantizer was improved by means of a non-linear predictive Vector Quantization scheme based on a Multi-Layer Perceptron (PVQ-MLP) [4]. It makes use of Artificial Neural Networks for predicting each coefficient individually using additional energy information, while prediction errors are quantized jointly by using Vector Quantization. With this non-linear predictive schema, both prediction gain and recognition accuracy improvements were reported, compared to the DVQ that makes use of an order one linear predictor.

In this work, another step for improving the compression method is presented. The proposed optimization algorithm solves the limitation of conventional single-path predictive vector quantizers, where the closest codeword for representing a single frame is chosen and sent out to the decoding side, without taking into account that future predictions can offer less quantization errors if a different codeword would be chosen. In order to tackle with this limitation, a global optimization can be done in a delayed decision coding approach, using the M-algorithm [5]. It preserves the M-best quantization hypotheses in each frame, where a minimum cost criterion is followed for choosing them. For the experiments presented in this paper, the Euclidean distance between the sequence of prediction errors and the sequence of codewords has been chosen as the cost function, however other functions for maximizing the recognition accuracy could be chosen.

The M-algorithm optimization, evaluated in the non-linear

This work has been partially funded by the national project TIN2008-06856-C05-04.

predictive vector quantizer schema, has been compared to PVQ-MLP, DVQ, VQ and the codebooks of the ETSI standard. All of them evaluated using the Advanced ETSI Front End (AFE) and Aurora 4 corpus which is a 5kword task with different acoustic environmental conditions including severe noise scenarios.

The remainder of this paper is organized as follows. First, the basics of DSR are briefly introduced in Section 2. In Section 3 an introduction to conventional predictive vector quantizers are presented, in section 4 the optimization algorithm applied to predictive vector quantizers is presented, while the experimental setup and performance evaluation are given in Section 5. Finally, the conclusions are provided in Section 6.

## 2. Distributed Speech Recognition

A feature compression algorithm is usually the last stage of the Front-End in DSR, in order to reduce the transmission bit-rate as much as possible. One of the most extended compression methods for DSR is Vector Quantization (VQ), which uses intra-frame redundancy of feature vectors for reducing the bit-rate providing good recognition performance [1]. The European Telecommunication Standards Institute (ETSI) has incorporated VQ as compression technique for all of its Front-End standards: ETSI 201 108, 202 050 and 202 212.

The ETSI standards Front-Ends offer 13 cepstral coefficients, and the log-energy coefficient, with noise reduction algorithms for the Advanced version and along with fundamental frequency and voicing class information in the Extended Advanced version. The compression stage is based on Vector Quantization of feature vectors pairs, resulting in 7 quantized pairs, in which $C_0$ is jointly quantized with log-energy, and the rest, quantized in adjacent pairs. The bit-rate obtained using this VQ is 4.4 kbps without channel error protection and without pitch and voicing class information.

However, using only VQ in the compression stage presents the main drawback that fail to exploit the strong inter-frame redundancy existing in MFCC vectors. Exploiting such inter-frame redundancy, along with intra-frame redundancy, would potentially lead to an increase in the compression rate. This can be done with a predictive vector quantization scheme, as the systems proposed in [3] [4], and the system presented in this paper. The idea of such schemes is the design of a more efficient source coding algorithm that removes all the non structured redundancy existing in the MFCCs, assuming that this new representation will be more sensitive to channel errors. However, the effect of channel errors can be neutralized by adding structured redundancy, in a lesser amount, by using channel coding techniques. In fact, the most important mobile networks (p.e. WiMaX or WiFi) provide error protection modules, and additionally for IP networks, the TCP protocol can be employed. Of course, these and other interesting known issues regarding the transmission of compressed acoustic features in ASR worth to be studied but they are beyond the scope of this work.

## 3. Predictive Vector Quantization of MFCC

Several compression schemes that make use of signal prediction jointly with Vector Quantization of the residual prediction error, have been successfully used in video and audio compression and, more recently, in DSR [3] [4]. Other predictive approaches for compressing acoustic features for DSR have been studied using order one linear prediction with scalar quantization in [2] and with a two-stage Vector Quantization in [6].

Predictive Vector Quantization with Multi-Layer Perceptron (PVQ-MLP) [4] performs prediction of each coefficient making use of a non-linear function that has an input layer with the latest quantized coefficients, and the latest energy quantized coefficients, outperforming DVQ [3], that employs an order one linear predictor, both in quantization error and recognition accuracy.

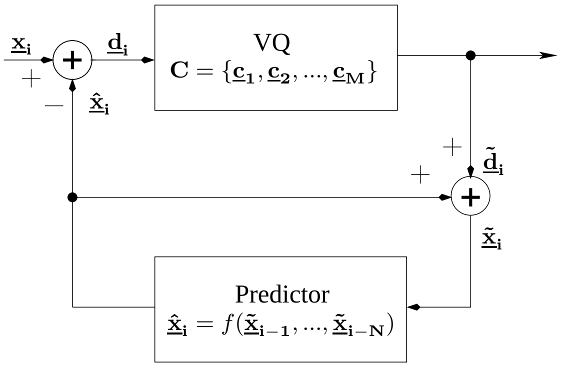
$\underline{\mathbf{x}}_{\mathbf{i}}$
$+$
$-$
$\underline{\mathbf{d}}_{\mathbf{i}}$
$\hat{\underline{\mathbf{x}}}_{\mathbf{i}}$
VQ
$\mathbf{C} = \{\underline{\mathbf{c}}_{\mathbf{1}}, \underline{\mathbf{c}}_{\mathbf{2}}, ..., \underline{\mathbf{c}}_{\mathbf{M}}\}$
$+$
$\tilde{\underline{\mathbf{d}}}_{\mathbf{i}}$
$+$
$\tilde{\underline{\mathbf{x}}}_{\mathbf{i}}$
Predictor
$\hat{\underline{\mathbf{x}}}_{\mathbf{i}} = f(\tilde{\underline{\mathbf{x}}}_{\mathbf{i-1}}, ..., \tilde{\underline{\mathbf{x}}}_{\mathbf{i-N}})$

Figure 1: *Block Diagram of a predictive vector quantization schema.*

The scheme for compressing a group of coefficients with a differential vector quantizer is shown in figure 1. For the $\mathbf{i^{th}}$ frame, each group of cepstral coefficients is denoted as $\underline{\mathbf{x}}_{\mathbf{i}}$. Over this tuple, a prediction is done by using the previous quantized values,

$$\hat{\underline{\mathbf{x}}}_{\mathbf{i}} = f(\tilde{\underline{\mathbf{x}}}_{\mathbf{i-1}}, ..., \tilde{\underline{\mathbf{x}}}_{\mathbf{i-N}}), \tag{1}$$

where $\mathbf{N}$ is the predictor order.

In a conventional differential vector quantizer, the prediction error, $\underline{\mathbf{d}}_i = \underline{\mathbf{x}}_{\mathbf{i}} - \hat{\underline{\mathbf{x}}}_{\mathbf{i}}$, is quantized by means of a codebook composed of $\mathbf{L}$ codewords, $\mathbf{C} = \{\underline{\mathbf{c}}_{\mathbf{1}}, \underline{\mathbf{c}}_{\mathbf{2}}, ..., \underline{\mathbf{c}}_{\mathbf{L}}\}$. The quantization procedure consists of choosing, for each frame, the closest codeword $\underline{c}_j$, using the Euclidean distance:

$$\tilde{\underline{\mathbf{d}}}_{\mathbf{i}} = \arg\min_{\underline{c}_j}\{|\underline{c}_j - \underline{\mathbf{d}}_{\mathbf{i}}|^2\} = \underline{\mathbf{d}}_{\mathbf{i}} + \underline{\mathbf{e}}_{\mathbf{i}} \tag{2}$$

where $\underline{c}_j$ is the $\mathbf{j^{th}}$ codeword, that will be sent out to the decoder side, and $\underline{\mathbf{e}}_{\mathbf{i}}$ is the quantization error. The quantized prediction error $\tilde{\underline{\mathbf{d}}}_{\mathbf{i}}$ is used to obtain the reconstructed coefficients $\tilde{\underline{\mathbf{x}}}_{\mathbf{i}} = \hat{\underline{\mathbf{x}}}_{\mathbf{i}} + \tilde{\underline{\mathbf{d}}}_{\mathbf{i}}$, that are also obtained in the decoder, and employed for predicting the forthcoming frames using (1).

Note that the reconstructed coefficients can be also expressed as

$$\tilde{\underline{\mathbf{x}}}_{\mathbf{i}} = \underline{\mathbf{x}}_{\mathbf{i}} + \underline{\mathbf{e}}_{\mathbf{i}}, \tag{3}$$

where it can be observed that the quantization error of the coefficients is the same than the quantization error of the prediction error.

## 4. M-algorithm optimization for Predictive Vector Quantization

Let $\mathbf{X} = \{\underline{\mathbf{x}}_{\mathbf{1}}, ..., \underline{\mathbf{x}}_{\mathbf{t}}, ..., \underline{\mathbf{x}}_{\mathbf{T}}\}$ be the original $\mathbf{T}$ frame sequence of coefficients to be quantized, $\tilde{\mathbf{X}} = \{\tilde{\underline{\mathbf{x}}}_{\mathbf{1}}, ..., \tilde{\underline{\mathbf{x}}}_{\mathbf{t}}, ..., \tilde{\underline{\mathbf{x}}}_{\mathbf{T}}\}$ the reconstructed coefficient sequence, $\tilde{\mathbf{D}} = \{\underline{\mathbf{d}}_{\mathbf{1}}, ..., \underline{\mathbf{d}}_{\mathbf{t}}, ..., \underline{\mathbf{d}}_{\mathbf{T}}\}$ the prediction error sequence, and $\mathbf{U} = \{\underline{\mathbf{u}}_{\mathbf{1}}, ..., \underline{\mathbf{u}}_{\mathbf{t}}, ..., \underline{\mathbf{u}}_{\mathbf{T}}\}$ the sequence of chosen codewords sent out to the decoder side, where $\underline{\mathbf{u}}_{\mathbf{t}} \in \mathbf{C}$.

The minimum squared quantization error for the whole sequence can be computed as:

$$\xi = \min_{\{\tilde{\mathbf{x}}_1,...,\tilde{\mathbf{x}}_\mathbf{T}\}} \sum_{t=1}^{\mathbf{T}} |\underline{\mathbf{x}}_t - \tilde{\underline{\mathbf{x}}}_t|^2 = \min_{\{\mathbf{u}_1,...,\mathbf{u}_\mathbf{T}\}} \sum_{t=1}^{\mathbf{T}} |\underline{\mathbf{d}}_t - \underline{\mathbf{u}}_t|^2 \quad (4)$$

However, note that for the instantaneous decision method (2) used in single-path predictive vector quantizers, there is no guarantee that $\xi$ could be obtained due to the fact that,

$$\min_{\{\mathbf{u}_1,...,\mathbf{u}_\mathbf{T}\}} \sum_{t=1}^{\mathbf{T}} |\underline{\mathbf{d}}_t - \underline{\mathbf{u}}_t|^2 \leq \sum_{t=1}^{\mathbf{T}} \min_{\mathbf{u_t}} |\underline{\mathbf{d}}_t - \underline{\mathbf{u}}_t|^2, \quad (5)$$

where the second term in (5) is the squared error obtained by a typical predictive vector quantizer that performs decisions in a frame by frame basis, as in (2). The inequality (5) is valid for predictive vector quantizers, since the term $\underline{\mathbf{d}}_t$ (containing the prediction error for frame $\mathbf{t}$) depends on previous codeword decisions, and previous values of the signal,

$$\underline{\mathbf{d}}_t = h(\underline{\mathbf{u}}_{\mathbf{t-1}}, ..., \underline{\mathbf{u}}_\mathbf{1}, \underline{\mathbf{x}}_\mathbf{t}, \underline{\mathbf{x}}_{\mathbf{t-1}}, ..., \underline{\mathbf{x}}_\mathbf{1}). \quad (6)$$

Note that in (5), the equality holds if $\underline{\mathbf{d}}_t$ is memoryless.

The problem that we want to solve is to choose the codeword sequence $\mathbf{U}$ that minimizes the Euclidean distance between the sequence of original coefficients $\mathbf{X}$ an the sequence of reconstructed coefficients $\tilde{\mathbf{X}}$. The exact solution to this coding problem could be obtained by using a brute force approach, computing the Euclidean distance for all possible codeword sequences $\mathbf{U}$. However, that is computationally intractable even for small values of $\mathbf{T}$ and $\mathbf{L}$.

In this paper we make use of the M-algorithm [5] in order to get an approximate solution to this problem that performs better than the single frame decision. The method consists of a synchronous evaluation algorithm, where in a frame by frame basis the M-best hypotheses (with minimum accumulated cost) are mantained. Before frame $\mathbf{t}$ is evaluated, a hypothesis is composed of an accumulated cost $\mathbf{a_{t-1}}$, an index history $\underline{\mathbf{i}}_{\mathbf{t-1}} = \{\mathbf{i}_1, ..., \mathbf{i}_{\mathbf{t-1}}\}$, and a history of reconstructed coefficients $\underline{\tilde{\mathbf{r}}}_{\mathbf{t-1}} = \{\underline{\tilde{\mathbf{x}}}_\mathbf{1}, ..., \underline{\tilde{\mathbf{x}}}_{\mathbf{t-1}}\}$, For each one of the $\mathbf{M}$ active hypothesis at frame $\mathbf{t}$, an instantaneous prediction error is extracted:

$$\underline{\mathbf{d}}_\mathbf{t} = \underline{\mathbf{x}}_\mathbf{t} - f(\underline{\tilde{\mathbf{x}}}_{\mathbf{t-1}}, ..., \underline{\tilde{\mathbf{x}}}_{\mathbf{t-N}}) \quad (7)$$

With this prediction error, the instantaneous Euclidean distance $\mathbf{o_{t,j}}$ is obtained for each codeword index $\mathbf{j} = \mathbf{1}..\mathbf{L}$,

$$\underline{\mathbf{o}}_{\mathbf{t},\mathbf{j}} = |\underline{c}_j - \underline{\mathbf{d}}_\mathbf{t}|^2 \quad (8)$$

Finally, the accumulated cost for that hypotheses propagated through the codeword index $\mathbf{j}$ is,

$$\mathbf{a}_{\mathbf{t},\mathbf{j}} = \mathbf{a}_{\mathbf{t-1}} + \underline{\mathbf{o}}_{\mathbf{t},\mathbf{j}} \quad (9)$$

If the evaluated hypothesis is selected as valid, the accumulated cost, the index history and the reconstructed coefficients are updated. The total number of prediction hypotheses that are evaluated at frame $\mathbf{t}$ become $\mathbf{ML}$. However, only the M-best hypothesis (with less accumulated cost $\mathbf{a}_{\mathbf{t},\mathbf{j}}$) are conserved for processing the next frame. When the last frame $\mathbf{T}$ is reached, the index history $\underline{\mathbf{i}}_\mathbf{T}$ with the lowest accumulated cost $\mathbf{a_T}$ is sent out to the receiver side, that performs the reconstruction of the coefficients like in a conventional predictive vector quantizer.

In a typical single-path predictive vector quantizer the codebook index of a quantized frame is chosen in a frame by frame basis independently of future quantizations (2), and only one prediction hypothesis is conserved in each frame. The optimization algorithm for $\mathbf{M} = \mathbf{1}$ is equivalent to the typical single-path predictive vector quantizer, however, with higher values of $\mathbf{M}$ lower quantization errors are obtained.

Table 1: Different bit-allocations for different bit-rates explored from 1.4 to 2.0 kbps

| Bit-rate | $C_1C_2$ | $C_3C_4$ | $C_5C_6$ | $C_7C_8C_9$ | $C_{10}C_{11}C_{12}$ | $C_0$ E |
|---|---|---|---|---|---|---|
| 1.4 | 3 | 3 | 2 | 2 | 2 | 2 |
| 1.6 | 3 | 3 | 3 | 3 | 2 | 2 |
| 1.8 | 3 | 3 | 3 | 3 | 3 | 3 |
| 2.0 | 4 | 4 | 3 | 3 | 3 | 3 |

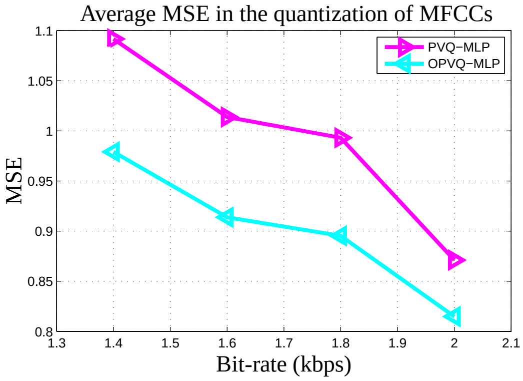

Figure 2: Average MSE in the quantization of MFCCs, in the test01 of Aurora 4

## 5. Performance Evaluation

In order to evaluate the performance of the proposed optimized predictive vector quantization method, OPVQ-MLP an extensive set of recognition experiments was carried out on a large vocabulary task, under different channel and noise conditions. The presented quantization scheme has been compared to the rest of quantization techniques exposed in [3][4] (PVQ-MLP, DVQ, variable length VQ, and the fixed length ETSI VQ).

The number of hypotheses per frame in the optimization algorithm OPVQ-MLP, $\mathbf{M}$, was fixed to 10, in a trade-off between computational complexity and quantization error performance, since it was observed that higher values of $\mathbf{M}$ did not reduce significantly the quantization error. The optimization algorithm was applied to each sub-vector group individually, in such a way that there was 10 hypothesis by frame and group. In the OPVQ-MLP and PVQ-MLP methods, MFCC sub-vectors were grouped as shown in Table 1, for testing bit-rates between 1.4 and 2.0 Kbps. For testing bit-rates between 700 and 4200 bps in the methods PVQ-MLP, DVQ and VQ, MFCC sub-vectors were grouped by pairs, as defined in the ETSI standard encoder, and the bit-rate was obtained assigning the same number of bits for each one of the 7 pairs, in such a way that using 1,2,3,4,5,6 bits by pair, a bit-rate of 700, 1400, 2100, 2800, 3500, 4200 bps is obtained, since a 100 frames per second rate is considered,

For all the experiments, the codebooks of OPVQ-MLP, PVQ-MLP, DVQ, and VQ were trained using different numbers of codewords under different train/test conditions. For training all the quantizers, we used the same training set that the one

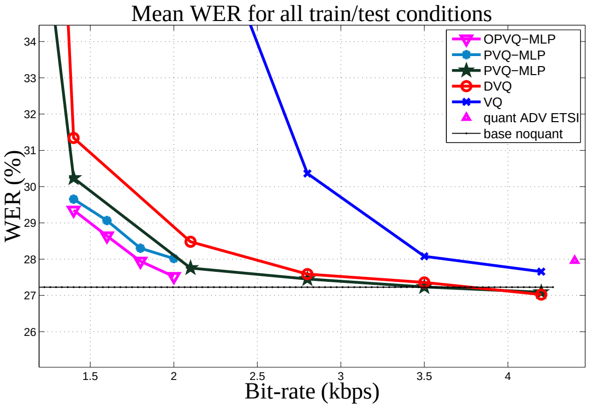

Figure 3: Mean results for all train-test combinations

used for training the acoustic models, so acoustic models were always adapted to the compression algorithms under all conditions.

The experiments for the ASR performance evaluation were carried out with the 8 kHz part of Aurora 4 database [7], designed by the Aurora Working Group of the ETSI. This database was conceived for developing robust Front-Ends and speech processing modules to be used in DSR systems. It is composed of a 5kword vocabulary based on DARPA Wall Street Journal (WSJ0) and contains 3 training sets (with 7138 utterances each one) and 14 test sets (with 330 utterances each one). Several acoustic environments are defined for composing 3 different train sets.

The recognizer and training tool employed for all the experiments was HTK, using a similar setup to that used in HIWIRE project for evaluating Aurora 4 database [8], that is, ETSI Advanced Front-End (AFE), cross-word tree-based tied-state triphones for acoustic models, with 3 states in each unit, and a GMM of 6 components for modeling the observation probability in each state. The language model employed was a back-off bigram.

Fig. 2 shows the quantization error for both methods with bit-rates ranging from 1.4 kbps to 2.0 kbps. It can be seen that OPVQ-MLP curve is always under PVQ-MLP curve, for all the evaluated bit-rates showing that a better quantization has been obtained thanks to the proposed optimization algorithm. These curves were obtained with test number one of Aurora 4, and the codebooks trained with clean signal. However, for all train-test combinations explored, the quantization performance is better for the OPVQ-MLP method than for the PVQ-MLP method.

Fig. 3 shows the mean Word Error Rate for all the experimental conditions described before (each one of 42 combinations train set - test set). As it can be seen, the DVQ performance is superior to the one obtained with conventional VQ methods for all code-book lengths, and OPVQ-MLP schema outperforms PVQ-MLP, DVQ and VQ methods. The degradation of DVQ method compared to a system without quantization is small for bit-rates above 2.1 kbps. However, the PVQ-MLP method can reach a bit-rate as low as 1.8 kbps with similar WER to ETSI quantizer, at 4.4 kbps, and slightly better than WER achieved by DVQ at 2.1 kbps, and the proposed OPVQ-MLP method can reach similar bit-rates than PVQ-MLP, but with less WER.

In comparative terms, it is worth pointing out that OPVQ-MLP at 1.6 kbps, PVQ-MLP at 1.8 kbps and DVQ at 2.1 kbps perform as well as VQ at 3.5 kbps and the AFE compression method at 4.4 kbps, with a small WER degradation over the baseline. Respect to the recognition results in different conditions It was observed that the behavior of the compression methods for different matching conditions is very homogeneous in comparative terms.

## 6. Conclusion

In this paper, a delayed decision coding algorithm for predictive vector quantization, the M-algorithm, has been evaluated in recognition experiments on a large vocabulary task, using Aurora 4 database. This algorithm extracts an optimal codeword sequence in the quantization process, in an efficient way, without evaluating all possible codeword combinations by maintaining the M-best hypotheses for each frame. Experimental results with M=10 showed that the proposed OPVQ-MLP quantizer outperforms the method PVQ-MLP, that decides the current codeword that must be sent to the back end in a single frame decision approach. The bit-rate that can be reached using OPVQ-MLP is 1.6 Kbps with a 5.1% of relative WER degradation with respect to an ASR system without quantization. Similar results in terms of WER can be obtained with the ETSI standards compression method, which makes use of 4.4 kbps, implying 175% of bandwidth increase relative to the proposed 1.6 kbps OPVQ-MLP method.

This study shows that the M-algorithm, in a delayed decision coding approach for predictive vector quantizers, can be used in order to get a minimum cost global optimization. In addition, a reduction in the necessary bit-rate without WER degradation has been reported on a large vocabulary task under different noise conditions, which highlights the benefits of using this method as compression stage in Distributed Speech Recognition.

## 7. References

[1] V. Digalakis, L. Neumeyer, and M. Perakakis, “Quantization of cepstral parameters for speech recognition over the world wide web,” in *proceedings of Icassp*, 1998.

[2] N. Srinivasamurthy, A. Ortega, and S. Narayanan, “Efficient scalable encoding for distributed speech recognition,” *Speech Communication*, 2006.

[3] J. E. Garcia, A. Ortega, A. Miguel, and E. Lleida, “Differential vector quantization of feature vectors for distributed speech recognition,” in *Interspeech*, 2009.

[4] J. E. Garcia, A. Ortega, A. Miguel, and E. Lleida, “Non-Linear Predictive Vector Quantization of Feature Vectors for Distributed Speech Recognition,” in *Interspeech*, 2010.

[5] F. Jelinek and J.B.Anderson, ”Instrumentable Tree Encoding of Information Sources”, IEEE Trans. on Information Theory, pp. 118-119, January 1971

[6] G. N. Ramaswamy and P. S. Gopalakrishnan, “Compression of acoustic features for speech recognition in network environments,” in *Icaasp*, 1998.

[7] G. Hirsch, “Experimental framework for the performance evaluation of speech recognition front-ends on a large vocabulary task,” ETSI STQ Aurora DSR Working, Tech. Rep., June 2001.

[8] J. C. Segura, T. Ehrette, A. Potamianos, D. Fohr, I. Illina, P. A. Breton, R. G. V. Clot, M. Matassoni, and P. Maragos, “The HIWIRE database, a noisy and non-native english,” Hiwire consortium, Tech. Rep., april 2007. [Online]. Available: http://www.hiwire.org