Robust Distributed Speech Recognition Using Histogram Equalization and Correlation Information

Pedro M. Martinez, Jose C. Segura, Luz Garcia

Department of Signal Theory, Networking and Communications
University of Granada, Spain

pmmartinez@auna.com, segura@ugr.es, luzgm@ugr.es

Abstract

In this paper, we propose a noise compensation method for robust speech recognition in DSR (Distributed Speech Recognition) systems based on histogram equalization and correlation information. The objective of this method is to exploit the correlation between components of the feature vector and the temporal correlation between consecutive frames of each component. The recognition experiments, including results in the Aurora 2, Aurora 3-Spanish and Aurora 3-Italian databases, demonstrate that the use of this correlation information increases the recognition accuracy.

Index Terms: Distributed Speech Recognition, noise compensation, histogram equalization, correlation information

1. Introduction

At present, the voice communication systems tend to take progressively away from the analogical world toward the digital world. Cellular phones and voice over IP (VoIP) services work in this technology, where the analogical voice signal is digitized before transmitting it. This digital processing allows to implement more and more complex functions that meet new necessities, such as the automatic speech recognition (ASR). This function can be very useful in those tasks which have traditionally been accomplished via buttons, but it also opens the doors to new services.

In practice, the implementation of an ASR system on every client's terminal can be unviable. The devices should have enough storage and processing ability to perform the whole ASR process, and this isn't always possible. Distributed Speech Recognition (DSR) appears to solve this problem, because the ASR system is distributed between the client and server. In this client-server architecture, the feature extraction of speech is performed locally at the client, where they are compressed and transmitted to a remote server, where the recognition system is implemented.

The speech features used are based on the Mel Frequency Cepstral Coefficients (MFCC) [1], which are the most commonly used parameters in currently available speech recognition systems. Their use achieves very high level of accuracy in clean speech environment, but results decrease quickly if the voice signal is affected by additive noise. This is because the speech recognition systems are generally trained with speech acquired under clean conditions and this doesn't model speech acquired under noisy conditions accurately.

Additive noise causes nonlinear distortion on coefficients value space and we have to use some compensation method to minimize this effect. In [2] and [3], MFCCs are compressed by using linear prediction and in [4], [5] and [6] DCT and 2D DCT is used. Histogram Equalization has been studied in [7] and [8], in order to improve the robustness of speech recognition systems. Other approaches have also been proposed (see for example [9]) that differ in the domain of application of HEQ. In [10], the authors show that the information of interframe correlation is very useful to improve the recognition.

In this paper, we propose a noise compensation method based on histogram equalization. This equalization is based on the hypothesis that, sorting the local coefficient values of the current frame, the position of the current frame in this order doesn't change significantly when the speech signal is affected by an additive noise. In other words, although noise changes all individual coefficient values, their local order statistics remain similar. This can be represented as a histogram, or cumulative distribution function.

Moreover, in order to exploit existing correlation between coefficients, it is logical to use a histogram-based vector quantization to quantize together each pair of MFCC parameters, as it is exposed in [8]. Additionally, another implicit information exists in MFCC values and it can be used to improve the quantization. It is the temporal correlation, or interframe correlation, between values of each coefficient. This is the main contribution of this paper. In a first step, we prove that this information by itself improves the quantization, since it increases the recognition accuracy. In a second step, we propose a method that uses both correlations (temporal correlation and correlation between coefficients), in order to improve the recognition performance as much as possible using all the information available.

The layout of this paper is as follows: in Section 2 the quantization method is described detailedly; Section 3 shows and discuss the results obtained with the Aurora 2, Aurora 3-Spanish and Aurora 3-Italian databases; finally, in Section 4 we expose the conclusions.

2. Description of the Method

As it has been commented, the proposed equalization is based on the hypothesis that, for each coefficient, the position that a frame has inside a sorted list of the values of its local frames isn't significantly changed by the presence of noise. Graphically, it can be shown as a histogram or a cumulative distribution function created by sorting the values of N frames around the current one. An example is shown in Figure 1, where we suppose that we have N frames from a clean utterance, and the same frames if some noise is added to the voice signal. Graphic $F_1(x)$ represents the cumulative distribution function of the N values from the clean utterance, and $F_2(x)$ match with the noisy utterance. According to this, if we only can see noisy values and the current frame has the value x_2, the best estimation we can

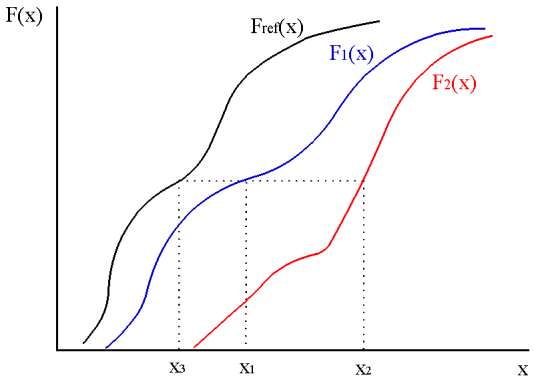

Figure 1: *Example that shows the fundamentals of the histogram-based equalization.*

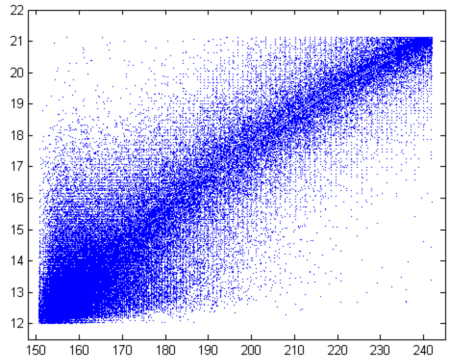

Figure 2: *Cloud of points for the pair C0/logE.*

do about the respective value in the clean utterance is x_1.

Nevertheless, in practice we don't have the cumulative distribution function of each clean utterance, we only have the noisy one, so that we can't estimate this value x_1 directly. The proposed solution is to use the same reference function in all equalizations, that is also used to equalize clean training utterances. As shown in Figure 1, if we equalize the value x_1 from the clean training utterance with the reference histogram $F_{ref}(x)$ we get x_3, and this is the value used to train the ASR system instead of x_1. In the recognition stage, the value x_2 from the noisy utterance is equalized with the same reference histogram $F_{ref}(x)$, so that we also get the value x_3.

As it has been commented, a histogram is created by ordering N local values of a MFCC coefficient and it is important to choose properly this number N. With the used databases, we have found empirically that the best results are reached with N=150. This way, for the short utterances of the database (100-200 frames) we use all frames of the utterance to create the histograms, but for long utterances we split them into segments of 150 frames.

In order to calculate the reference histogram of each MFCC coefficient, we use the information of the clean training utterances of the respective database. We get all possible histograms from these utterances according to the specified number of frames N, and the reference histogram of each coefficient is calculated averaging them. We use these reference histograms to equalize the value of the coeficients of any utterance. This equalization is applied to noisy test utterances for the noise compensation, but it is also used with clean training utterances, because the values used to train the ASR system have to be also transformed into the range of values defined by the reference histogram.

Before the transmision of the equalized values from the client to the server of the DSR system, we need to apply a quantization. We have to define some quantization levels with their respective centroids, and the previously mentioned correlation information is used here. Depending on the amount of information used, we have proposed four different methods detailed below.

2.1. 1D quantization

This is the most simple case, because each coefficient is quantized individually and we don't use any correlation information. The objective is to calculate a reference value of the recognition accuracy that can be used to compare with the results of the following cases in order to check the increases introduced by the correlation information. It's very simple to calculate the quantization levels by using the reference histogram. We take equidistant points on the vertical scale and the correspondences on the horizontal scale based on the histogram are the centroids.

2.2. 2D quantization

The quantization of the equalized MFCC coefficients is performed in pairs to exploit the correlation between them. This information is contained in the position of the centroids and the clean training utterances are used to calculate them. For each frame we take each pair of equalized coefficients obtaining a two-dimensional point in the plane. By doing this for every frame of all the utterances we get a cloud of points that provides information about the probability of the values and their correlation. Figure 2 shows an example of this cloud of points for a pair of coefficients. Then we use the LBG [11] algorithm to calculate the centroids based on this probability distribution with the values previously normalized.

2.3. 2DT quantization

The main idea is that the value of a frame and the value of the next frame are connected. In this case the coefficients are taken individually, but we use the value of the current frame and the difference with the value of the previous frame in order to exploit the temporal correlation. As in 2D case, this information

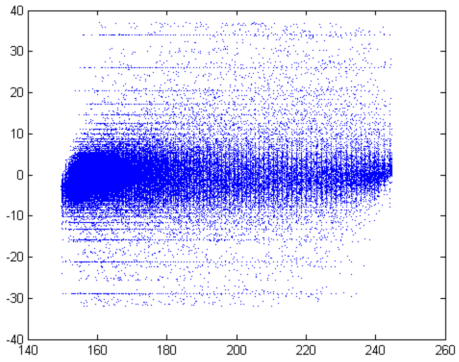

Figure 3: *Cloud of points for the pair C0/difference with the previous frame.*

is also contained in the position of the centroids. The first step is to calculate the reference histogram of the difference between frames, because these values must be also equalized. In order to calculate the centroids we also get a cloud of points from the clean training utterances and use the LBG algorithm. The difference is that in this case each point match with the equalized value of a coefficient and the equalized value of the difference with the previous frame. For that reason, the distribution of these points provides information about temporal correlation. Figure 3 shows an example of this cloud of points.

The DSR server receives each quantization number and searches for the respective centroid in the look-up table, that has two parts: the value of the coefficient and the value of the difference. We need to reconstruct only the MFCC coefficients, but we use the difference to refine the estimated value of the coefficient in the previous frame. If the centroid of the coefficient C in the previous frame $(t-1)$ is $cent_{t-1} = (coef_{t-1}, dif_{t-1})$ and the centroid in the current frame t is $cent_t = (coef_t, dif_t)$, we say that $C(t) = coef_t$ and the difference dif_t is used to refine the estimated value of the coefficient in the previous frame $C(t-1) = coef_{t-1}$. According to the current frame, the coefficient in the previous frame must be $C(t-1) = (coef_t - dif_t)$, so that we take an average of both estimated values:

$$C(t-1) = (coef_{t-1} + (coef_t - dif_t))/2$$

In the next frame $(t+1)$, the value of the difference dif_{t+1} will be used to refine the estimated value of the coefficient in the current frame $C(t) = coef_t$.

2.4. 4D quantization

We take the coefficients in pairs, as in 2D case, but the difference with the previous frame is also calculated for every coefficient, as in 2DT case, so we obtain points in a four-dimensional space. We also use the clean training utterances to obtain the information about temporal correlation and correlation between coefficients, that is used to find the centroids. In quantization phase we calculate the distance to every centroid and the smallest one is taken. Empirically we have proved that, instead of using euclidean distance (each component of the distance vector is squared), better results are obtained raising each component to the power of four.

As in 2DT case, the value of the difference is used in the DSR server to refine the estimated value of the coefficient in the previous frame. The same equation as in 2DT case is used for each of the two coefficients of each pair.

3. Results and Discussion

Digit recognition experiments were done with Aurora 2, Aurora 3 Spanish and Aurora 3 Italian databases. In the first case, training was performed on clean data and testing was done with the 10 different types of noise that are considered in this database (grouped in sets A, B and C). In each type SNR from 20dB to 0dB were tested. With Aurora 3 Spanish and Aurora 3 Italian databases, we used the training/test conditions from High-Mismatch (HM) experiment.

The ETSI DSR standard Aurora front-end [12] was used for the MFCC feature extraction, obtaining 14 coefficients to be quantized (C1-C12, together with C0 and logarithmic frame energy (logE)). For the recognition process the software HTK 3.2 has been used. It takes 13 coefficients (C0 is excluded), together with their corresponding delta and acceleration coefficients. The final feature vector dimension is 39.

	Bit Rate	Set A	Set B	Set C	Average	Rel. Imp.
MFCC	4,4 Kbps	61,34	55,75	66,14	**61,08**	**00,00**
1D - 32	7,0 Kbps	81,86	82,93	80,64	**81,81**	**53,26**
2D - 64	4,2 Kbps	83,00	84,84	82,23	**83,36**	**57,24**
2DT - 64	8,4 Kbps	82,96	84,18	81,47	**82,87**	**55,99**
4D - 256	5,6 Kbps	83,58	85,15	83,17	**83,97**	**58,81**
4D - 512	6,3 Kbps	83,89	85,26	83,58	**84,24**	**59,51**

Table 1: *Results with Aurora 2, detailed for sets A, B and C.*

	MFCC	1D - 32	2D - 64	2DT - 64	4D - 256	4D - 512
SNR20	94,07	97,01	97,50	97,07	97,49	97,56
SNR15	85,04	94,87	95,53	95,09	95,65	95,75
SNR10	65,52	89,86	90,97	90,23	91,00	91,27
SNR5	38,61	77,59	79,49	78,96	80,23	80,47
SNR0	17,09	50,90	54,43	54,39	56,26	56,84

Table 2: *Results with Aurora 2, detailed for the different SNR values.*

The computational complexity of the proposed method is the same of that of the AFE in the 2D case and twice in the 4D case for an equal number of centroids.

3.1. Experiments with Aurora 2

Table 1 lists the accuracy recognition results of HTK with Aurora 2 testing sets (A,B and C). The first row lists the results using original MFCCs, without any noise compensation method. The second row lists the results using 1D quantization method with 32 levels, and the performance is better because of the histogram-based equalization. In each case the number of quantization levels has been choosen to achieve the best results. For 2D quantization with 64 levels, in the third row, the recognition increases in comparison with 1D quantization because of the use of the correlation information. The last three rows list the most interesting results of this paper. For 2DT quantization with 64 levels the results increase in comparison with 1D quantization. This confirm that the use of the temporal correlation information improve the recognition accuracy. Moreover, the use of both correlations in 4D quantization improve the previous results. In addition to the recognition accuracy, the last column lists the "Relative Improvement" that is the relative error reduction over the MFCC baseline.

Table 2 lists the detailed results for all SNR values from 20dB to 0dB that is very interesting. We can see that in the first three rows, with non-critical noise (20dB, 15 dB and 10dB), the temporal correlation don't cause significant improvements. However, in the last row (under very poor SNR conditions) the use of the temporal correlation information is very notable and we get a considerable increase of recognition accuracy for 2DT and 4D quantizations in comparison with 1D quantization. Therefore, the temporal correlation information used in this method introduces robustness against noise.

3.2. Experiments with Aurora 3 Spanish

Table 3 lists the recognition accuracy results with this database under HM conditions. As with Aurora 2, we show in different rows the results with MFCC directly and 1D, 2D, 2DT, 4D-256 and 4D-512 quantizations. In this case, besides the recognition accuracy ("Accuracy"), the table lists the percentage of digits

	Bit Rate	%Correct	Accuracy	Rel. Imp.
MFCC	4,4 Kbps	67,91	43,25	00,00
1D - 32	7,0 Kbps	85,29	82,50	69,16
2D - 64	4,2 Kbps	87,10	84,33	72,39
2DT - 64	8,4 Kbps	87,46	83,94	71,70
4D - 256	5,6 Kbps	89,14	84,57	72,81
4D - 512	6,3 Kbps	89,56	84,96	73,50

Table 3: *Results with Aurora 3 Spanish.*

	Bit Rate	%Correct	Accuracy	Rel. Imp.
MFCC	4,4 Kbps	66,67	44,44	00,00
1D - 32	7,0 Kbps	75,93	71,10	47,98
2D - 64	4,2 Kbps	77,72	74,33	53,80
2DT - 64	8,4 Kbps	79,87	76,54	57,78
4D - 256	5,6 Kbps	81,23	78,14	60,66

Table 4: *Results with Aurora 3 Italian.*

correctly recognized ("%Correct") that doesn't take in mind the number of insertions. As we can see for 2DT and 4D quantizations the use of the temporal correlation information improve the recognition accuracy. Also, we wish to emphasize that the increase of results is quite larger for "%Correct". The insertions problem is essentially caused by utterances with long inter-digit silences. The great level of noise that are added in those moments leads the ASR system into error, because it finds digits where there isn't any. The last column lists the "Relative Improvement".

3.3. Experiments with Aurora 3 Italian

As we can see in Table 4, the increase in recognition accuracy for 2DT and 4D quantizations is even higher than with the other two databases. We deduce that temporal correlation is more useful in this database than in the others and we confirm the good performance of the method.

We don't show the result with 4D-512 quantization because with this database the recognition doesn't increase if we use more than 256 levels.

4. Conclusions

According to the results, we can assert that the proposed objective has been achieved. We have proved that the inclusion of the temporal correlation information increases the recognition accuracy in a histogram-based noise compensation method. Some secondary objectives have been achieved for this:

1. We have developed a histogram-based noise compensation method with 1D quantization and we have proved that the recognition increases in comparison with the use of the original MFCCs.
2. The previous method has been changed in order to include the correlation between coefficients in the 2D quantization and we have proved that the performance is better than in 1D quantization.
3. We have changed the 1D quantization method in order to include the temporal correlation information. We have proved that this information increases the recognition accuracy.
4. Finally, both previous methods have been combined and we have proved that this 4D quantization increases the reconition accuracy.

5. Acknowledgements

This work has received research funding from the EU 6th Framework Programme, under contract number IST-2002-507943 (HIWIRE, Human Input that Works in Real Environments) and SR3-VoIP projects (TEC2004-03829/TCM) from the Spanish government. The views expressed here are those of the authors only. The Community is not liable for any use that may be made of the information contained therein.

6. References

[1] S.B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In *IEEE Trans. Acoustics Speech and Signal Processing*, pages 357–366 Vol. ASSP–28(4), 1980.

[2] G. Ramaswamy and P. Gopalakrishnan. Compression of acoustic features for speech recognition in network environments. In *IEEE ICASSP*, pages 977–980, 1998.

[3] N. Srinivasamurthy, A. Ortega, Q. Zhu, and A. Alwan. Towards efficient and scalable speech compression schemes for robust speech recognition applications. In *IEEE ICME*, pages 259–252 Vol.1, 2000.

[4] I. Kiss and P. Kapanen. Robust feature vector compression algorithm for distributed speech recognition. In *Eurospeech*, pages 2183–2186, 1999.

[5] Q. Zhu and A. Alwan. An efficient and scalable 2d dct-based feature coding scheme for remote speech recognition. In *IEEE ICASSP*, pages 113–116 Vol.1, Aug 2001.

[6] W. Hsu and L. Lee. Efficient and robust distributed speech recognition (dsr) over wireless fading channels: 2d-dct compression, iterative bit allocation, short bch code and interleaving. In *IEEE ICASSP*, 2004.

[7] A. de la Torre, A.M. Peinado, J.C. Segura, J.L. Prez-Crdoba, M.C. Bentez, and A.J. Rubio. Histogram equalization of speech representation for robust speech recognition. In *IEEE Transactions on Speech and Audio Processing*, pages 355–366 Vol.13 No.3, May 2005.

[8] C. Wan and L. Lee. Histogram-based quantization (hq) for robust and scalable distributed speech recognition. In *Interspeech*, pages 957–960, 2005.

[9] F. Hilger and H. Ney. Quantile based histogram equalization for noise robust speech recognition. In *European Conf. on Speech Communication and Technology*, pages 1135–1138 Vol.2, 2001.

[10] K.K. Paliwal and S. So. Scalable distributed speech recognition using multi-frame gmm-based block quantization. In *ICSLP*, 2004.

[11] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. In *IEEE Transactions on Communications*, pages 84–95, 1980.

[12] *Speech Processing, Transmission and Quality Aspect (STQ); Distributed Speech Recognition; Front-end Feature Extraction Algorithm; Compression Algorithms.* Tech. Rep. Standard ES 201 108, European Telecommunications Standards Institute (ETSI), April 11 2000.