MULTI-ENVIRONMENT MODELS BASED LINEAR NORMALIZATION FOR ROBUST SPEECH RECOGNITION

Luis Buera, Eduardo Lleida, Antonio Miguel, and Alfonso Ortega

University of Zaragoza, Spain
{lbuera,lleida,amiguel,ortega}@unizar.es

ABSTRACT

This paper presents a feature normalization technique based on minimum mean square error, histogram normalization and multi-environment models. Using stereo training data, accurate estimates of the bias between clean and distorted speech cepstral vectors can be provided. With the stereo training data, a non-linear transformation of the distorted cepstral vectors is performed based on minimum mean square error estimation and histogram equalization. Results with SpeechDat Car database show an improvement in the word error rate with regard to linear transformation techniques as SPLICE [1] and MEMLIN [2]. An improvement in word error rate of 67.28% in digits task, and 40.79% in spelling task are obtained.

1. INTRODUCTION

The mismatch between training and testing acoustic conditions is one of the most important reasons for the degradation on the performance of the automatic speech recognition systems [3]. In this paper, we propose a new non-linear transformation feature normalization algorithm based on histogram normalization and the Minimum Mean Square Error (MMSE) estimator.

Feature normalization based on MMSE has been successfully applied on the cepstral domain. In this sense, algorithms like multivariate gaussian based cepstral normalization algorithm, RATZ [4], Stereo based Piecewise LInear Compensation for Environments, SPLICE [1], or Multi-environment Models based LInear Normalization, MEMLIN [2], are some examples. All of them are based on the use of stereo training data to provide accurate estimates of the bias between clean and distorted speech cepstral vectors. Although the nonlinearity between the cepstral vectors of clean and distorted cepstral, the feature normalization is performed by means of a linear transformation. The main difference between SPLICE and MEMLIN is the noise models. SPLICE assumes no explicit noise model and MEMLIN assumes a set of basic noisy environment models. The proposed new algorithm follows the same assumptions that MEMLIN with regard to the noise model but performs a non-linear transformation based on histogram equalization to deal with the nonlinearity between clean and distorted speech in the cepstral domain.

The new algorithm, called Multi-Environment Models based HIstogram Normalization (MEMHIN), as MEMLIN, performs a feature normalization using probabilistic models for the clean speech, for the noisy acoustic environments and for the conditional probability between clean and distorted cepstral vectors. To compensate the differences in the variance between the clean and the distorted speech cepstral vectors, a non linear transformation is learnt for each pair of gaussians, based on histogram equalization. So, the target of this algorithm is to learn a non-linear transformation between clean and distorted feature vectors associated to a pair of gaussians (one for a clean model, and the other one for a noisy model), for each basic defined acoustic environment. This knowledge, the gaussians associated, the conditional probability between clean and noisy gaussians, and the environments are the data used to compensate the mismatch between clean and distorted vectors.

The MEMLIN and MEMHIN algorithms are compared using real distorted speech from the Spanish SpeechDat-Car database and artificial distorted speech (controlled additive distortion).

This paper is organized as follows: in section 2, the MMSE estimator is presented, and the equations for MEMLIN and MEMHIN are obtained. The expressions of MMSE parameters are explained in section 3. The results are shown in section 4. Finally, the conclusions are explained in section 5.

2. MMSE ESTIMATOR

Given the clean feature vector x, and the noisy one, y, the clean estimation vector, $\hat{x}$, can be calculated by MMSE estimation:

This work has been supported by MECD

$$\hat{x} = E[x|y] = \int_x x p(x|y) dx \quad (1)$$

where $p(x|y)$ is the probability density function, PDF, of x given y. In order to evaluate the expression (1) for MEMLIN and MEMHIN, some assumptions are made:

Noisy signal is divided into several basic environments, and for each environment, it will be modelled as a mixture of gaussians:

$$p_e(y) = \sum_{s_y^e} p(y|s_y^e) p(s_y^e) \quad (2)$$

$$p(y|s_y^e) = N(y; \mu_{s_y^e}, \Sigma_{s_y^e}) \quad (3)$$

where e represents the environment index, s_y^e denotes the correspondent gaussian of the noisy model for the e environment, $\mu_{s_y^e}$, $\Sigma_{s_y^e}$, $p(s_y^e)$ are the mean vector, the diagonal covariance matrix, and the weight associated to s_y^e, and $p(y|s_y^e)$ is the probability of noisy feature vector, given the noisy gaussian.

Clean feature vector is modelled with a distribution of mixture gaussians:

$$p(x) = \sum_{s_x} p(x|s_x) p(s_x) \quad (4)$$

$$p(x|s_x) = N(x; \mu_{s_x}, \Sigma_{s_x}) \quad (5)$$

where s_x denotes the correspondent gaussian of the clean model, μ_{s_x}, Σ_{s_x}, and $p(s_x)$ are the mean, diagonal covariance matrix, and the weight associated to s_x. $p(x|s_x)$ is the probability of the clean feature vector, given the clean gaussian.

The third assumption is consider the clean feature vector, x as a function of the noisy one, y, the clean model gaussian, s_x, and the noisy environment model gaussian, s_y^e:

$$x \simeq f(y, s_x, s_y^e) \quad (6)$$

With all these assumptions, (1) can be approximated in the following way, using Bayes theorem:

$$\begin{aligned} \hat{x} &\simeq \int_x \sum_e \sum_{s_x} \sum_{s_y^e} f(y, s_x, s_y^e) p(x, s_x, s_y^e|y) dx = \\ &= \sum_e \sum_{s_x} \sum_{s_y^e} f(y, s_x, s_y^e) p(s_x|s_y^e, y) p(s_y^e|y) \end{aligned} \quad (7)$$

where $p(x, s_x, s_y^e|y)$ is the probability of x, s_x, and s_y^e given y. It can be seen, using Bayes theorem, as $p(x, s_x, s_y^e|y) = p(x|s_x, s_y^e) p(s_y^e|y) p(s_x|s_y^e, y)$, where $p(x|s_x, s_y^e)$ is the probability of clean feature vector, given the noisy and clean gaussians, $p(s_y^e|y)$ is the probability os the noisy gaussian, given y, and $p(s_x|s_y^e, y)$ is the probability of the clean gaussian, given the noisy one and the noisy feature vector. On the other hand, in order to calculate $p(s_y^e|y)$, Bayes theorem can be used: $p(s_y^e|y) = p(e|y) p(s_y^e|e, y)$, where $p(e|y)$ is the probability of e environment, given the noisy feature vector, and $p(s_y^e|e, y)$ is the probability of the noisy gaussian, given the environment and y. To simplify the notation, α_e will be used in the following expressions instead of $p(e|y)$; so $p(e|y) = \alpha_e$.

The expression of the clean feature estimation, made in the third assumption, (2), is the main difference between MEMLIN, and MEMHIN. MEMLIN uses a linear function (8), and MEMHIN uses a non linear one, obtained by histogram equalization [5] (9):

$$x \simeq f_{MEMLIN}(y, s_x, s_y^e) = y - r_{s_x, s_y^e} \quad (8)$$

$$x \simeq f_{MEMHIN}(y, s_x, s_y^e) = C^{-1}_{x, s_x, s_y^e}(C_{y, s_x, s_y^e}(y)) \quad (9)$$

where r_{s_x, s_y^e} is the independent term of the linear transformation for MEMLIN. It is associated to each pair of gaussians: one of the noisy model of a certain environment, s_y^e, and the other one of the clean model, s_x. In MEMHIN function, C_{x, s_x, s_y^e} is the clean feature vectors cumulative probability associated to s_x and s_y^e gaussians, and C^{-1}_{x, s_x, s_y^e} is the inverse function. C_{y, s_x, s_y^e} is the noisy feature vectors cumulative probability associated to s_x and s_y^e gaussians.

With these approximations, ((8), and (9)) the final expressions of (7) for MEMLIN, and MEMHIN are (10), and (11), respectively:

$$\hat{x}_t \simeq y_t - \sum_{s_x} \sum_e \sum_{s_y^e} \alpha_{e,t} p(s_y^e|y_t) p(s_x|s_y^e, y_t) r_{s_x, s_y^e} \quad (10)$$

$$\hat{x}_t \simeq \sum_{s_x} \sum_e \sum_{s_y^e} \alpha_{e,t} p(s_y^e|y_t) p(s_x|s_y^e, y_t) C^{-1}_{x, s_x, s_y^e}(C_{y, s_x, s_y^e}(y_t)) \quad (11)$$

where t is a temporal index. With the same mathematical theory, and others assumptions, the expressions of RATZ and SPLICE can be obtained. For RATZ algorithm, the clean feature vector is modelled as a mixture of gaussians ((4), and (5)), and the approximation for x is:

$$x \simeq f_{RATZ}(y, s_x) = y - r_{s_x} \quad (12)$$

where r_{s_x} is the independent term of the linear transformation and it only depends on the clean gaussian. Finally, the estimator clean vector is obtained as:

$$\hat{x}_t \simeq y_t - \sum_{s_x} p(s_x|y_t) r_{s_x} \quad (13)$$

For SPLICE technique, noisy feature vectors are modelled with one mixture of gaussians, and the approximation for x is:

$$x \simeq f_{SPLICE}(y, s_y) = y - r_{s_y} \quad (14)$$

where s_y is the correspondent noisy gaussian, and r_{s_y} is the independent term of the linear transformation. It only depends on the noisy gaussian. The estimator clean vector is obtained as:

$$\hat{x}_t \simeq y_t - \sum_{s_y} p(s_y|y_t) r_{s_y} \quad (15)$$

As it can be seen in [2], where SPLICE, MEMLIN, and other techniques are compared, the use of clean and several noisy environments, produces a more specifical transformation that models more properly the mismatch between clean and noisy feature vectors, and it produces a great improvement. On the other hand, the use of linear transformations supposes that the correspondent environment does not produces a variance transformation between clean and noisy feature vectors, given a pair of gaussians; only a mean shift. A non linear transformation assumes a possible variance transformation, and this is the main improvement of MEMHIN concerning MEMLIN. As it is well known that convolutional noise in time domain produces, mainly, a mean shift in Mel Cepstrum domain [6], and additive noise produces, principally, a variance transformation [6], it is reasonable to think that MEMHIN will obtain better results than MEMLIN when additive noise is more important, and this improvement will be less significant when convolutional noise was the most important noise in the environment.

In order to observe the effect that MEMLIN produces in a certain feature component and environment, the fig.1 is presented. In upper plot, the relation between the second MFCC coefficient for clean test signal (y axis), and the correspondent noisy one (x axis) is presented. The environment selected is E1 (car stopped and motor running). The lower plot is the same relation when the noisy signal is normalized with MEMLIN technique and 32 gaussians for noisy and clean environments. The represented line in two figures is $x = y$. It can be seen that MEMLIN obtains a good approximation to the ideal line. The behavior for MEMHIN and for others environments and MFCC coefficients is very similar.

3. MMSE PARAMETERS ESTIMATION

In order to calculate the estimator vector, $\hat{x}_t$, for MEMLIN and MEMHIN, some variables have to be obtained: $\alpha_{e,t}$, $p(s_y^e|y_t)$, $p(s_x|s_y^e, y_t)$, r_{s_x,s_y^e}, C_{x,s_x,s_y^e} and C_{y,s_x,s_y^e}. The first two, as are dependent of noisy feature vector, are computed during recognition. The other ones have to be obtained in a training process in which stereo data is needed.

For $\alpha_{e,t}$ an iterative solution is considered. Each moment, t, a noisy feature vector is available, y_t. The

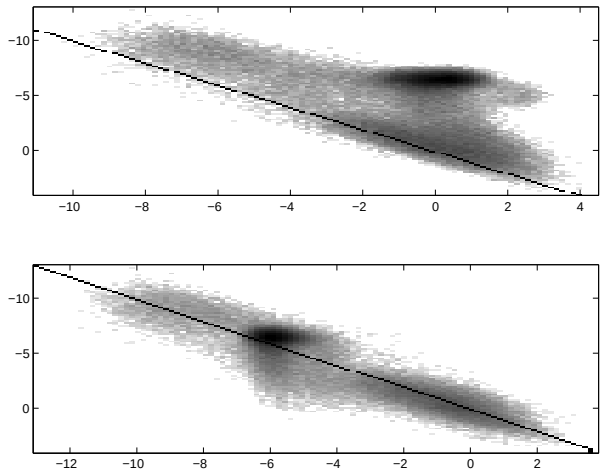

Fig. 1. Clean and noisy second MFCC coefficient representation, and clean and normalized second MFCC coefficient representation

calculation of the environment weight in this moment will be:

$$\alpha_{e,t} = \beta \cdot \alpha_{e,t-1} + (1-\beta)\frac{p_e(y_t)}{\sum_e p_e(y_t)} \quad (16)$$

where β is the memory constant. $\alpha_{e,0}$ are considered uniform for all environments. Also, $p(s_y^e|y_t)$ can be calculated using (2), (3), and Bayes:

$$p(s_y^e|y_t) = \frac{p(y_t|s_y^e)p(s_y^e)}{\sum_{s_y^e} p(y_t|s_y^e)p(s_y^e)} \quad (17)$$

In order to compute $p(s_x|s_y^e, y_t)$, r_{s_x,s_y^e}, C_{x,s_x,s_y^e} and C_{y,s_x,s_y^e}, a training process with available stereo data for each environment is needed: $X_e = \{x_1^e, ..., x_{t_e}^e, ..., x_{T_e}^e\}$, for clean feature vectors and $Y_e = \{y_1^e, ..., y_{t_e}^e, ..., y_{T_e}^e\}$ for noisy ones, with $t_e \in [1, T_e]$.

The conditional probability, $p(s_x|s_y^e, y_t)$, can be considered time independent, and it is estimated with the stereo training data by relative frequency:

$$p(s_x|s_y^e, y_t) \simeq p(s_x|s_y^e) = \frac{C_N(s_x|s_y^e)}{N} \quad (18)$$

where $C_N(s_x|s_y^e)$ is the number of times that the most probable pair of gaussians is s_x, and s_y^e for each pair of stereo feature vectors of e environment, and N is the number of times that the most probable gaussian for noisy vector is s_y^e in e environment.

For MEMLIN, the calculate of r_{s_x,s_y^e} (20) can be obtained by minimizing the weighted square error, E_{s_x,s_y^e} (19):

$$E_{s_x,s_y^e} = \sum_{t_e} p(s_x|x_{t_e}) p(s_y^e|y_{t_e})(x_{t_e} - y_{t_e} + r_{s_x,s_y^e})^2 \quad (19)$$

	E1	E2	E3	E4	E5	E6	E7	MWER
T1: C0-C0	0.38	2.06	1.40	0.50	0.57	0.16	0.0	0.86
T1: C0-C2	4.29	11.08	11.61	14.79	14.49	11.27	20.07	11.53
T1: C2-C2	1.14	4.80	2.80	3.38	4.39	1.59	1.36	3.07
T2: C0-C0	11.81	12.45	12.94	11.03	11.92	7.41	8.86	11.36
T2: C0-C2	24.94	35.19	40.04	42.73	46.49	39.46	56.00	38.58
T2: C2-C2	24.87	27.96	30.42	25.38	32.04	22.73	29.43	27.59

Table 1. WER digits and spelling tasks baselines results

	E1	E2	E3	E4	E5	E6	E7	MWER	MIMP
MEMLIN 8-8	1.73	5.75	4.62	7.52	9.91	6.83	10.88	6.27	52.64
MEMLIN 16-16	1.44	5.66	4.20	5.89	7.53	5.87	8.50	5.25	61.03
MEMLIN 32-32	1.05	5.57	4.20	5.01	7.34	4.92	6.46	4.79	65.65
MEMHIN 8-8	1.63	5.66	4.90	7.27	9.06	6.51	10.2	6.01	54.84
MEMHIN 16-16	1.34	5.83	4.06	6.39	7.91	6.03	7.82	5.37	60.31
MEMHIN 32-32	0.96	5.15	4.06	5.51	7.15	4.60	6.46	**4.67**	**67.28**

Table 2. WER results with MEMLIN, and MEMHIN techniques for digits task

$$r_{s_x,s_y^e} = \frac{\sum_{t_e} p(s_x|x^e_{t_e})p(s^e_y|y^e_{t_e})(y^e_{t_e} - x^e_{t_e})}{\sum_{t_e} p(s_x|x^e_{t_e})p(s^e_y|y^e_{t_e})} \quad (20)$$

where $p(s_x|x^e_{t_e})$ is the probability of s_x given the clean feature vector. It can be calculated with (4), and (5), with Bayes theorem in a similar way as (17):

$$p(s_x|x^e_{t_e}) = \frac{p(x_{t_e}|s_x)p(s_x)}{\sum_{s_x} p(x_{t_e}|s_x)p(s_x)} \quad (21)$$

On the other hand, for MEMHIN, and associated to each pair of gaussianas (one for the clean model, and one for noisy environment one), the n bands histograms for each component of the feature vector are obtained (the components are considered independent [5]). In order to do that, the components for each pair of gaussianas are weighted by $p(s_x|x_{t_e})p(s^e_y|y_{t_e})$. With the histograms, C_{x,s_x,s^e_y} and C_{y,s_x,s^e_y} are calculated, cumulating the bands.

RATZ needs a training process with stereo data in order to obtain the following variables: r_{s_x}, and $p(s_x|y_t)$. The mathematic technique is similar to MEMLIN. To calculate $p(s_x|y_t)$, it is used (4), and (5), with Bayes theorem in a similar way as (17). r_{s_x} is calculated ((23)) minimizing the weighted square error, E_{s_x} (22):

$$E_{s_x} = \sum_{t_e} p(s_x|x_{t_e})(x_{t_e} - y_{t_e} + r_{s_x})^2 \quad (22)$$

$$r_{s_x} = \frac{\sum_{t_e} p(s_x|x^e_{t_e})(y^e_{t_e} - x^e_{t_e})}{\sum_{t_e} p(s_x|x^e_{t_e})} \quad (23)$$

SPLICE training process is similar to MEMLIN one; $p(s_y|y)$ is obtained in a similar way of (17), and r_{s_y} is calculated by minimizing the weighted square error, E_{s_x}:

$$E_{s_y} = \sum_{t_e} p(s_y|y_{t_e})(x_{t_e} - y_{t_e} + r_{s_y})^2 \quad (24)$$

$$r_{s_y} = \frac{\sum_{t_e} p(s_y|y^e_{t_e})(y^e_{t_e} - x^e_{t_e})}{\sum_{t_e} p(s_y|y^e_{t_e})} \quad (25)$$

4. RESULTS

A set of experiments have been carried out using the Spanish SpeechDat Car database [7]. Seven environments are defined: car stopped, motor running (E1), town traffic, windows close and climatizer off (silent conditions) (E2), town traffic and noisy conditions: windows open and/or climatizer on (E3), low speed, rough road, and silent conditions (E4), low speed, rough road, and noisy conditions (E5), high speed, good road, and silent conditions (E6), and high speed, good road, and noisy conditions (E7).

The tasks used are isolated and continuous digits (task T1), and spelling (task T2). All the phrases are 16 KHz sampled. The clean signals are recorded with a close talk

	E1	E2	E3	E4	E5	E6	E7	MWER	MIMP
MEMLIN 8-8	21.00	26.84	26.99	29.69	36.31	32.57	43.14	29.40	34.03
MEMLIN 16-16	20.46	27.12	27.10	27.48	34.57	29.76	42.29	28.39	37.44
MEMLIN 32-32	20.00	26.98	27.88	26.98	34.10	28.74	40.29	28.00	38.89
MEMHIN 8-8	21.31	26.63	26.77	28.59	34.96	31.29	42.86	28.82	35.56
MEMHIN 16-16	20.54	26.84	27.54	26.68	33.31	29.76	41.43	28.01	38.50
MEMHIN 32-32	20.08	26.08	27.99	26.78	32.28	27.84	41.43	**27.44**	**40.79**

Table 3. WER results with MEMLIN, and MEMHIN techniques for spelling task

	E1	E2	E3	E4	E5	E6	E7	MWER
C0-C2 (0dB)	33.27	29.59	29.79	35.21	36.70	23.81	25.85	31.55
C0-C2 (5dB)	12.77	9.43	13.01	13.03	11.63	6.35	1.36	10.64
C0-C2 (10dB)	4.19	4.20	6.99	4.76	3.81	2.86	0.0	4.19
C0-C2 (15dB)	2.00	2.49	4.61	2.51	1.81	0.95	0.0	2.24

Table 4. Baseline results: WER for several SNRs in digits task

	E1	E2	E3	E4	E5	E6	E7	MWER	MIMP
MEMLIN 8-8	20.32	20.93	22.52	24.44	22.78	15.56	8.84	20.63	35.57
MEMLIN 16-16	19.37	19.21	20.84	22.81	22.31	15.08	8.16	19.49	39.38
MEMLIN 32-32	18.22	18.1	19.44	21.05	19.45	12.38	6.12	17.70	45.30
MEMHIN 8-8	16.3	17.67	21.82	19.8	20.50	15.08	8.16	17.98	43.94
MEMHIN 16-16	15.34	17.58	19.86	18.80	19.35	14.60	6.46	17.05	46.96
MEMHIN 32-32	14.96	18.10	19.86	18.67	18.49	14.13	6.46	**16.86**	**47.49**

Table 5. WER results with SNR 0dB with MEMLIN, and MEMHIN techniques for digits task

	E1	E2	E3	E4	E5	E6	E7	MWER	MIMP
MEMLIN 8-8	9.20	7.38	11.61	9.40	8.20	5.71	0.68	8.15	25.99
MEMLIN 16-16	8.53	7.03	10.77	8.40	8.58	5.08	0.68	7.71	30.47
MEMLIN 32-32	8.15	6.70	9.09	7.64	7.72	4.92	0.34	7.06	37.62
MEMHIN 8-8	7.57	7.46	8.25	7.14	7.72	4.92	0.34	6.93	37.82
MEMHIN 16-16	7.00	6.76	8.81	5.89	7.53	5.08	0.0	6.55	42.73
MEMHIN 32-32	6.52	6.52	8.10	5.89	7.53	5.08	0.0	**6.37**	**44.44**

Table 6. WER results with SNR 5dB with MEMLIN, and MEMHIN techniques for digits task

	E1	E2	E3	E4	E5	E6	E7	MWER	MIMP
MEMLIN 8-8	3.55	3.34	5.59	3.63	3.91	2.54	0.0	3.55	19.95
MEMLIN 16-16	2.68	3.00	5.17	3.51	3.71	2.38	0.0	3.19	31.1
MEMLIN 32-32	2.68	3.09	5.00	3.38	3.24	2.38	0.0	3.08	33.86
MEMHIN 8-8	2.97	3.69	5.03	2.76	2.86	1.75	0.0	3.04	33.39
MEMHIN 16-16	3.07	3.00	4.90	2.88	2.76	1.59	0.0	2.88	41.04
MEMHIN 32-32	2.97	3.00	5.03	2.76	2.76	1.43	0.0	**2.84**	**42.35**

Table 7. WER results with SNR 10dB with MEMLIN, and MEMHIN techniques for digits task

	E1	E2	E3	E4	E5	E6	E7	MWER	MIMP
MEMLIN 8-8	1.82	2.40	4.34	2.51	1.62	0.95	0.0	2.12	10.76
MEMLIN 16-16	1.73	2.23	4.06	2.38	1.52	0.95	0.0	2.00	24.07
MEMLIN 32-32	1.63	2.23	4.06	2.26	1.52	0.95	0.0	1.97	26.14
MEMHIN 8-8	1.82	2.32	3.50	1.75	1.52	0.95	0.0	1.88	25.39
MEMHIN 16-16	1.92	2.23	3.64	1.75	1.43	0.95	0.0	1.88	29.57
MEMHIN 32-32	1.73	2.23	3.64	1.63	1.43	0.95	0.0	**1.83**	**32.71**

Table 8. WER results with SNR 15dB with MEMLIN, and MEMHIN techniques for digits task

microphone (Shune SM-10A), which it is called C0, and the noisy signals are recorded by a microphone placed on the car ceiling in front of the driver (Peiker ME15/V520-1): it is C2. The SNR range for the clean signals goes from 20 to 30 dB, and for the noisy signals goes from 5 to 20 dB. 12 MFCC and energy are computed each 10 ms using a 25 ms hamming window.

The feature normalization techniques are applied over the 12 MFCC and delta energy. The clean and noisy models are built for these feature vectors with 8, 16, or 32 gaussians.

For recognition, environment E1 has 200 phrases, E2 223, E3 136, E4 152, E5 200, E6 120, and E7 has 56 phrases, on the other hand, the feature vector is composed of the 12 normalized MFCC with cepstral mean substraction, the first and second derivative and the normalized delta energy, given a feature vector of 37 coefficients. The context depended acoustic models are composed of 699 one state continuous density HMM with at least 16 gaussians per state. These units are defined dividing each phonetic unit into its left context, the unit without context, and its right context. For example, the Spanish phonetic word /k/ /a/ /s/ /a/, is transformed into /♯ $<$ k/ /k/ /k$>$a/ /k$<$a/ /a/ /a$>$s/ /a$<$s/ /s/ /s$>$a/ /s$<$a/ /a/ /a$>$ ♯/, where ♯ means any context, $<$ represents left context, and $>$ is right context.

The baseline word error rate, WER, results for each environment are presented in table 1. MWER is the mean WER with the seven environments. C0-C0 represents testing clean signal with clean models, C0-C2 represents testing noisy signal with clean models, C2-C2 represents the results testing noisy signal with all environments noisy models.

The WER comparative results between the different techniques for task T1 and task T2 can be seen in table 2 and 3, respectively. Next to the technique, appears the numbers of clean and noisy used model gaussians, 8, 16 or 32. The mean improvement (MIMP) is calculated between C0-C0 and C0-C2, and MWER are presented in the correspondent tables, too. MEMLIN and MEMHIN use all environments to normalize (E1,...,E7). MEMHIN histograms were calculated with 600 bands.

It can be observed that the good behavior of MEMLIN and MEMHIN are similar in the two tasks: MEMLIN obtains an improvement of 65.65% in task T1, and 38.89% in task T2. The improvement of MEMHIN concerning MEMLIN is not very important: 1.63% in task T1, and 1.90% in task T2, but, as it has been said, the strong point of MEMHIN is when additive noise is the important mismatch between clean and noisy feature vectors.

In order to study the improvement of MEMHIN concerning MEMLIN in additive noise environments, an experiment was developed. Digits task clean signal was contaminated with additive noise and different SNRs: 0dB, 5dB, 10dB, and 15dB. Each phrase was contaminated with its own noise. The baseline appears in table 4. The behavior of these two algorithms was studied, and the results can be seen in table 5 (for 0dB), table 6 (for 5dB), table 7 (for

10dB), and table 8 (for 15dB).

The results for 10dB and 15dB and environment E7 are not very representative because there are only 56 phrases in the test corpus. Since the range to calculate the improvement in these cases is 0, MIMP is calculated without the environment E7. The mean improvement of MEMHIN concerning MEMLIN in all SNRs is very significant, and it is more important when the number of gaussians is small. So, the mean WER improvement when 8 gaussians are used is 12.07%, with 16 gaussians it is 8.82%, and, finally, with 32 gaussians, the improvement is 6.02%. This shows that the variance normalization that MEMHIN proposes can be a good solution when the environments are characterized by additive noise.

5. CONCLUSIONS

In this paper it has been compared two techniques based on MMSE estimator: MEMLIN, and MEMHIN. The main difference between them is the transformation proposed to model the differences between noisy and clean feature vectors. MEMLIN uses a linear transformation with pendent equal 1, and MEMHIN proposes a non linear one: this supposes that the environment may produce a mean and a variance shift between clean and noisy feature vectors. The results show a small improvement in most environments when there is an important convolutional noise, but this improvement goes up when the most important mismatch between noise and clean signal is additive noise. In this case, the use of MEMHIN produces a improvement of more than 6%. Anyway, the results with both techniques are significant better than with similar algorithms, like SPLICE [2].

6. REFERENCES

[1] J. Droppo, L. Deng, and A. Acero, "Evaluation of the splice algorithm on the aurora2 database," *in Proc. Eurospeech*, vol. 1, Sep. 2001.

[2] L. Buera, E. Lleida, A. Miguel, and A. Ortega, "Multi-environment models based linear normalization for speech recognition in car conditions," *in Proc. ICASSP*, May. 2004.

[3] S. Sagayama, K. Shimoda, M. Nakai, and H. Shimodaira, "Analytic methods for acoustic model adaptation: a review," *in Proc. Isca ITR-Workshop2001*, pp. 67–76, Aug 2001.

[4] P. Moreno, "Speech recognition in noisy environments," *Ph. D. Thesis*, ECE Department, Carnegie-Mellon University. Apr. 1996.

[5] Angel de la Torre, Antonio M. Peinado, Jose C. Segura, Jose L. Perez, Carmen Benítez, and Antonio J. Rubio, "Histogram equalization of the speech representation for robust speech recognition," *IEEE Trans. on Speech and Audio Processing*, to be plublished. 2004.

[6] Woei-Chyang Shieh and Sen-Chia Chang, "The dependence of feature vectors under adverse noise," *in Proc. Eurospeech*, Sep. 1999.

[7] A. Moreno, A. Noguiera, and A. Sesma, "Speechdat-car: Spanish," *Technical Report SpeechDat*.