Jacobian Adaptation based on the Frequency-Filtered Spectral Energies

Alberto Abad, Climent Nadeu, Javier Hernando and Jaume Padrell

TALP Research Center
Universitat Politècnica de Catalunya, Barcelona, Spain
{alberto, climent, javier, jaume}@talp.upc.es

Abstract[†]

Jacobian Adaptation (JA) of the acoustic models is an efficient adaptation technique for robust speech recognition. Several improvements for the JA have been proposed in the last years, either to generalize the Jacobian linear transformation for the case of large noise mismatch between training and testing or to extend the adaptation to other degrading factors, like channel distortion and vocal tract length. However, the JA technique has only been used so far with the conventional mel-frequency cepstral coefficients (MFCC). In this paper, the JA technique is applied to an alternative type of features, the Frequency-Filtered (FF) spectral energies, resulting in a more computationally efficient approach. Furthermore, in experimental tests with the database Aurora1, this new approach has shown an improved recognition performance with respect to the Jacobian adaptation with MFCCs.

1. Introduction

Recognition systems in real world applications are severely degraded by mismatch between training and testing conditions. This mismatch is generally associated to noise environmental conditions, channel distortion and articulatory effects, and the need for alleviating these problems has become an important challenge of the speech technology. Many robust recognition techniques have been proposed in the last years, including novel speech representations and acoustic model compensation techniques.

Jacobian Adaptation (JA) proposed in [1] belongs to this second approach. In some previous works [1, 2, 3, 4] it has been shown that JA is an effective and relatively low computational cost technique in comparison with other model adaptation techniques, like Parallel Model Combination (PMC) [2].

In this work, we propose to use JA with the Frequency-Filtered (FF) spectral representation [5], instead of the widely used mel-frequency cepstrum coefficients (MFCC). The FF parameter set basically consists in the substitution of the Discrete Cosine Transformation (DCT) by a simple low-order filtering of the frequency sequence of spectral energies. The fact that the FF features lie in the frequency domain, implies a lower computational cost of the adaptation algorithm.

In section 2 we first review JA and FF representation. Then, the Frequency Filtering Jacobian Adaptation (FF-JA) technique is presented. This new technique is compared to the Jacobian Adaptation with MFCC (MFCC-JA), and to an optimization previously proposed in [3], and its computational advantage is showed. In section 3, experimental results with the Aurora1 database, which has been used for developing the ETSI DSR-AFE standard front-end [6], are presented. In them, FF-JA has shown an improved recognition performance with respect to the MFCC-JA, and also with respect to mean subtraction of either the cepstral coefficients (MFCC-MS, i.e. the well known CMS) or the FF features (FF-MS).

2. The Frequency Filtering Jacobian Adaptation approach

2.1. Fundamentals of Jacobian Adaptation

Jacobian Adaptation is an acoustic model compensation technique based on a well-known approximation, which consists in that little modifications of the variables of an analytic function affect this function in a linear dependent way with the partial derivatives. In [1], this approach is extensively presented and it is shown that it is possible to find a simple and efficient linear function able to adapt Continuous Density Hidden Markov Models (CDHMM) from certain noisy training conditions to others. More recent contributions proposed modifications for the adaptation of other degrading factors like channel distortion and vocal tract length [7], but in this paper we will only adapt the mean vectors of the CDHMM. The equation that represents the transformation is as follows

$$\hat{C}_{s+n} = C_{s+n} + \frac{\partial C_{s+n}}{\partial C_n}\left(Cn_{tar} - Cn_{ref}\right) \tag{1}$$

where $\partial C_{s+n}/\partial C_n$ is the Jacobian matrix, $\hat{C}_{s+n}$ and C_{s+n} are respectively the new adapted and the original noisy speech cepstrum mean vector, and Cn_{ref} and Cn_{tar} are the reference and the target noise cepstrum vector, i.e. the noise present in the training signals and the actual noise present in the recognition phase respectively. Considering that relationship between cepstra of speech, noise and noisy speech is

$$C_{s+n} = F\left[\log\left\{\exp\left(F^{-1}Cs\right) + \exp\left(F^{-1}Cn\right)\right\}\right] \tag{2}$$

where logarithmic and exponential operations are performed individually for each vector's component, it can be demonstrated that Jacobian matrix can be written as

$$\frac{\partial C_{s+n}}{\partial C_n} = Fdiag\left(\frac{N_{ref}}{S + N_{ref}}\right)F^{-1} \tag{3}$$

[†] This work has been partially supported by the Spanish Government project PETRA (TIC-2000-1735-C02-01) and the EU project FAME (IST-2000-28323).

where F is the DCT matrix transformation, F^{-1} is its inverse, N_{ref} is the training noise filter-bank energies (FBE) vector and S is the FBE mixture vector of the noisy speech model. The quotient is computed element by element and *diag( )* is the diagonal matrix formed with the elements of the vector inside.

Conventional JA algorithm can be summarized in two main steps. In the training phase the reference noise is estimated in order to calculate the Jacobian matrices of every mixture model using equation (3). In the recognition phase the actual testing noise is estimated to upgrade the models using equation (1) and the Jacobian matrices previously computed.

The α-JA [3] is a modified version of the original one that alleviates the problem of JA for large mismatch between training and testing. The expression of the α-JA matrix is

$$\frac{\partial C_{s+n}}{\partial C_n} = F \cdot diag\left(\frac{\alpha N_{ref}}{S + \alpha N_{ref}}\right) F^{-1} \qquad (4)$$

It is shown in [3] that this last approximation gets better results than the original one. In this paper we will use the alpha modified version, but we will just refer to as Jacobian Adaptation.

2.2. The FF feature representation

Logarithmic filter-bank energies (FBE) are typical spectral measurements in most current speech recognition systems. The discrete cosine transform is applied to compute, from the set of energies, a set of uncorrelated features, the so-called mel-frequency cepstral coefficients, which is probably the most widely used spectral representation in speech recognition.

In [5], the authors proposed a computational simple alternative to the DCT, called Frequency Filtering. The FF features have generally shown an equal or better recognition performance than the MFCCs [8], and, unlike them, the FF features show a frequency meaning. The FF technique consists of a filtering operation, typically with the second order filter

$$H(z) = z - z^{-1} \qquad (5)$$

In matrix notation [8],

$$C_F = H \cdot log(S) \qquad (6)$$

where C_F is the vector of the frequency-filtered parameters, S is the vector of (linear) FBEs and H is the matrix:

$$H = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & -1 & 0 \end{pmatrix} \qquad (7)$$

2.3. The FF-JA formulation

As it was done for the cepstral coefficients, we may define the Jacobian approximation for the adaptation of the mean vectors of the acoustic models based on the frequency-filtered spectral energies. This FF-JA technique can be simply rewritten substituting in the original formulation, the cepstrum vectors (C) with the FF vectors (C_F), and the DCT matrix and its inverse with the filter matrix H in (7) and its inverse, that is:

$$\hat{C}_{F_{s+n}} = C_{F_{s+n}} + H \cdot diag\left(\frac{\alpha N_{ref}}{S + \alpha N_{ref}}\right) H^{-1} \cdot \left(C_{Fn_{tar}} - C_{Fn_{ref}}\right) \qquad (8)$$

$$H^{-1} = \begin{pmatrix} 0 & -1 & 0 & -1 & 0 & -1 & \dots & -1 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & \dots & -1 & 0 & -1 \\ 1 & 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & -1 & 0 & -1 \\ 1 & 0 & 1 & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 & 1 & 0 & \dots & 0 & 1 & 0 \end{pmatrix} \qquad (9)$$

This inverse matrix is only valid when the number of feature coefficients is even. This can not be considered an important constraint. In fact, MFCC-JA needs the inclusion of the zeroth cepstral coefficient to properly perform the inversion.

Some computational advantages of the new FF-JA approach can be expected. On the one hand, the calculation of the FF parameters simply consists of subtracting two band energies. On the other hand, the frequency physical meaning of the FF representation may be useful for implementing robust techniques such as JA.

2.4. Computational advantage of FF-JA

JA is a fast adaptation algorithm, but it needs a lot of memory space to store all the Jacobian matrices and this can be a handicap in some embedded systems. An optimization algorithm has been previously presented in [3], which basically consists of expressing Jacobian matrices as functions of a basis of canonical matrices, which can be reduced by using Principal Component Analysis (PCA-JA). This approach allows reduction in both memory and computational cost of the JA.

FF-JA also allows reducing memory space requirements due to the special characteristics of the filter matrices. If we define the parameters

$$\gamma_k = \frac{\alpha N_k}{S_k + \alpha N_k} \qquad (10)$$

where N_k and S_k are respectively the k-th component of noise and noisy speech FBE vectors, it can be demonstrated that JA-FF matrices can be written as

$$J = \begin{pmatrix} \gamma_2 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \gamma_1 & 0 & \gamma_1 - \gamma_3 & 0 & \gamma_1 - \gamma_3 & \dots & \gamma_1 - \gamma_3 & 0 & \gamma_1 - \gamma_3 \\ \gamma_4 - \gamma_2 & 0 & \gamma_4 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma_3 & 0 & \gamma_3 - \gamma_5 & \dots & \gamma_3 - \gamma_5 & 0 & \gamma_3 - \gamma_5 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \gamma_{n-3} & 0 & \gamma_{n-3} - \gamma_{n-5} \\ \gamma_n - \gamma_{n-2} & 0 & \gamma_n - \gamma_{n-2} & 0 & \gamma_n - \gamma_{n-2} & 0 & \dots & 0 & \gamma_n & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & \gamma_{n-1} \end{pmatrix} \qquad (11)$$

This structure permits that Jacobian matrices can be easily reconstructed from the parameters γ_k. Concretely, if we call *Nf* the dimension of the filter matrix and *Ng* the number of Gaussians to adapt, only *Ng* vectors of *Nf* components are needed to store, whereas in the MFCC-JA approach it is necessary to store *Ng* matrices of *NfxNf* components. The new FF-JA approach is also more efficient in terms of memory space than the optimization algorithm in [3], which needs to store *Ng* vectors of *Nf* components and N_{PCA} canonical matrices of *NfxNf* components.

However, the reconstruction algorithm implies an obvious computational cost increase in the recognition phase. This drawback can be compensated in the matrix-per-vector computation due to the sparseness of FF-JA matrices. MFCC-JA needs *Ng* full matrix-per-vector computations. PCA-JA needs N_{PCA} full matrix-per-vector and $N_{PCA}xNg$ scalar-per-vector computations. FF-JA needs *Ng* sparse matrix-per-vector computations. The number of elements different from zero of the sparse matrices is about $Nf^2/4$ and some additional operations are needed for the reconstruction of the matrices.

Table 1 shows a comparison between MFCC-JA, PCA-JA and FF-JA. To measure the computational cost, only additions and multiplications are considered, and it is also assumed that Gaussian means and all matrices and vectors needed for each adaptation algorithm are stored in memory.

	Memory	Operations
MFCC-JA	$Ng \cdot Nf^2$ (117600)	$2Ng \cdot Nf^2$ (235200)
PCA-JA	$Ng \cdot Nf + N_{PCA} \cdot Nf^2$ (9772)	$2N_{PCA} \cdot \{Nf^2 + Ng \cdot Nf\}$ (120344)
FF-JA	$Ng \cdot Nf$ (8400)	$Ng \cdot \{Nf^2/2 + 2Nf\}$ (75600)

Table 1: Comparison of Jacobian Adaptation algorithms in terms of memory and computations. In brackets, an example with *Ng*=600, *Nf*=14 and N_{PCA}=7.

3. Evaluation tests

3.1. Experimental set-up

Experimental evaluation has been carried out with the Aurora1 database, which consists of the TIdigits speech database with artificially added noise [6]. It provides 422 speech continuous digit realizations of different speakers for each training SNR and for four different noise conditions: ‘exhibition hall’, ‘babble noise’, ‘car moving’ and ‘suburban train’ noise. 1000 testing realizations are provided for six different SNR levels and the same four noisy scenarios.

Recognition system was implemented with HTKv3.0 Toolkit. HTK libraries were modified to include the FF representation. CDHMM of digits were defined with 18 states and 3 mixtures for each state. Silence model was composed by 5 states and 6 mixtures for each state and the interword silence model is formed with 3 states and 6 mixtures for each state. After pre-emphasizing signals with a zero at 0.97, Hamming windowed frames of 25 ms were taken every 10 ms. 14 cepstral coefficients, including the zeroth cepstral one, were obtained from a 23 component log-FBE vector in experiments with MFCC. When FF parameterization was considered, also 14 coefficients were obtained from a 14 log-energies vector. The first and second derivatives are also included in both representations. The filter used in the FF parameterization is the one proposed in equation (5).

JA of only static components of mean vectors was implemented for MFCC as well as for FF features. In previous works [1], it has been shown that JA of covariance matrices and delta-cepstrum vectors does not contribute in a significant way to recognition improvement. Noise training reference was obtained from the most probable mixture of the middle state of the silence model. Target noise was obtained from the average of the 7 first frames of the testing signal. For JA, the α parameter was set to 3.

3.2. Experimental results

A complete set of experiments has been carried out for six different techniques: MFCC, FF, MFCC-JA, FF-JA, MFCC-MS and FF-MS. The last two techniques are identical to MFCC and FF respectively, but including mean subtraction (MS), that in the case of the MFCC features is the well-known cepstral mean subtraction (CMS) technique.

Training for each technique has been done with each of the four noise environments at three different SNR levels (20, 15 and 10). Test signals are classified into four noise conditions as well, and into six different SNR levels (20, 15, 10, 5, 0 and -5). The complete testing set is used for each training combination, i.e., all test signals are used for testing models trained with a specific noise condition and SNR, and for a given technique.

Table 2 shows the whole set of recognition results with Aurora1 in two ways. On the left side, results depending on the noise conditions are shown. The percentage scores result from averaging across SNR in all testing and training combinations, but without including results for SNR=-5. That is, each result of the table is computed by averaging the 15 results obtained combining the 3 SNR training levels (20, 15 , and 10 dB) and the 5 considered testing levels (20, 15, 10, 5, and 0 dB). On the right side of the table, average results depending on the reference and target SNR level are shown. These results have been computed by averaging all the combinations of noise conditions, i.e., each result is the average of the 16 pairs of reference-target noises having the same SNR for both training and testing. Also, a total average result is shown to easily compare the different techniques. Obviously, the average score per technique is the same at both sides of the table.

From the bottom row (global averages) of the table, we can firstly observe that all the FF-based techniques show better scores than their corresponding MFCC versions, and the highest ones are obtained when FF is used with JA. Actually, a 10,2% relative improvement is obtained in terms of recognition accuracy from FF to FF-JA, which means 27% error rate reduction. Since, for MFCC, only a 4,68% relative improvement is obtained with JA, it seems that the FF features are better matched to the JA technique than the MFCCs.

For both speech representations, a remarkable improvement is obtained if mean subtraction is performed. In fact, MFCC-MS (CMS) clearly outperforms MFCC-JA. Actually, the same surprising observation was already done in [2], where the authors observed how MFCC-JA did not get better performance than CMS and, in some tests, it even performed worse than CMS.

Partial results on the left side of the table show that FF-JA scores are always the highest ones, except when for tests with 'babble' noise, when the best results are obtained with FF-MS. Regarding the right side of the table, it can be seen that a general trend of the partial results is that FF-MS performs better than FF-JA when little SNR differences between training and test are found; however, when the SNR mismatch increases, FF-JA outperforms all the other techniques.

In general, the worst results are always obtained when the 'babble' noise is the testing one for all the techniques and so the total average results are heavily affected by the low performance in that noise condition. In this way, if we compare the total average results excluding 'babble' noise, we obtain that FF-JA shows even more advantage (89,21 %) in front of FF-MS (86,42 %) and MFCC-MS (86,33%).

Finally, although results are not shown in this paper, several multi-condition training experiments have been also carried out. As it could be expected, although JA is an efficient technique to adapt from certain noise conditions to others, it does not perform better than MFCC or FF when all testing noise situations are included in the training set.

4. Conclusions

In this paper, the Jacobian adaptation of CDHMM acoustic models has been used with the frequency-filtered spectral energies to take advantage of the frequency meaning of this kind of features. In this way, besides offering a reduction of the computational load, the new FF-JA approach shows in our experiments a higher average recognition accuracy than both the already reported MFCC-JA technique and the mean subtraction techniques.

5. References

[1] Sagayama S., Yamaguchi Y., Takahashi S. and J. Takahashi, "Jacobian approach to fast acoustic model adaptation", Proc. ICASSP, vol. 2, pp. 835-838, 1997.

[2] Pärssinen K., Salmela P., Harju M. and Kiss I., "Comparing Jacobian adaptation with cepstral mean normalization and parallel model combination for noise robust speech recognition", Proc. ICASSP, vol. 1, pp. 193-196, 2002.

[3] Cerisara C., Rigazio L. Boman R. and Junqua J-C., "Transformation of Jacobian matrices for noisy speech recognition", Proc. ICSLP, vol. 1, pp. 369-372, 2000.

[4] Sarikaya R. and Hansen J., "Improved Jacobian adaptation for fast acoustic model adaptation in noisy speech recognition", Proc. ICSLP, vol. 3, pp. 702-705, 2000.

[5] Nadeu C., Hernando J. and Gorricho M., "On the decorrelation of filter-bank energies in speech recognition", Proc. Eurospeech, pp. 1381-1384, 1995.

[6] D. Pearce, "Experimental Framework for the Performance Evaluation of Distributed Speech Recognition Front-ends", Aurora Document Number AU/120/98, Sept. 1998.

[7] Shimodaira H., Sakai N., Nakai M. and Sagayama S., "Jacobian joint adaptation to noise, channel and vocal tract length", Proc. ICASSP, vol. 1, pp. 197-200, 2002.

[8] Nadeu C., Macho D. and Hernando J., "Time and frequency filtering of filter-bank energies for robust HMM speech recognition", Speech Communication, vol. 34, pp. 93-114, 2001.

Averaging across SNR							
TR	TEST	MFCC	FF	MFCC-MS	FF-MS	MFCC-JA	FF-JA
HALL	Hall	75,56	79,12	85,93	84,81	74,29	87,00
	Babble	37,79	36,02	46,88	52,43	39,77	47,37
	Train	69,51	68,89	87,91	88,19	76,61	89,47
	Car	72,27	79,18	92,11	91,29	84,98	93,59
BABBLE	Hall	73,29	80,08	67,69	72,52	74,38	80,29
	Babble	81,85	80,72	81,27	82,57	78,04	80,94
	Train	80,20	85,36	77,17	80,90	81,33	84,85
	Car	85,60	88,38	82,93	85,30	87,73	90,00
TRAIN	Hall	78,83	79,25	78,98	80,33	79,55	84,08
	Babble	45,71	45,35	52,43	53,71	46,91	48,62
	Train	87,86	86,72	88,03	88,78	87,79	90,84
	Car	79,54	78,53	89,57	90,23	91,50	93,86
CAR	Hall	66,52	74,21	77,76	77,48	73,04	81,91
	Babble	40,95	39,82	47,66	48,10	38,61	44,88
	Train	65,86	67,84	84,05	83,95	79,94	87,97
	Car	91,70	91,90	92,61	92,72	91,54	94,19
AVERAGE		70,81	72,59	77,06	78,33	74,13	79,99

Averaging across noise conditions							
TR	TEST	MFCC	FF	MFCC-MS	FF-MS	MFCC-JA	FF-JA
SNR 20	20	92,02	92,54	93,23	94,07	92,37	93,75
	15	88,95	89,14	90,78	91,66	89,83	91,09
	10	79,78	80,66	84,51	85,98	82,94	86,31
	5	56,00	60,95	68,78	72,02	60,79	76,40
	0	29,39	31,95	41,88	44,32	32,54	55,51
SNR 15	20	89,47	90,90	91,06	92,17	90,24	92,20
	15	87,45	88,00	88,92	89,97	88,53	89,51
	10	80,92	80,60	84,37	85,15	84,00	85,07
	5	63,10	65,09	72,65	74,33	70,10	76,11
	0	35,80	38,57	48,97	50,47	42,59	57,49
SNR 10	20	84,01	88,07	88,76	90,57	84,84	89,90
	15	84,69	86,84	87,86	88,74	85,70	87,91
	10	80,86	81,71	84,28	84,47	82,88	83,64
	5	67,89	68,78	74,80	75,22	73,96	75,57
	0	41,90	44,99	55,08	55,82	50,59	59,45
AVERAGE		70,81	72,59	77,06	78,33	74,13	79,99

Table 2: On the left side, word accuracy results by averaging across the training (TR) SNR (20, 15 and 10 dB) and the testing SNR (from 20 to 0 dB). On the right side, word accuracy results by averaging across all training and testing noise conditions.