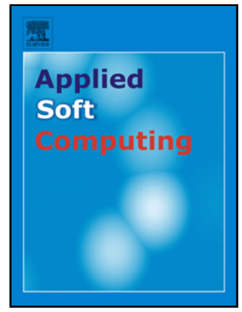
Accepted Manuscript



Title: Evolutionary cepstral coefficients

Authors: Leandro D. Vignolo, Hugo L. Rufiner, Diego H. Milone, John C. Goddard

PII:	S1568-4946(11)00022-6
DOI:	doi:10.1016/j.asoc.2011.01.012
Reference:	ASOC 1066
To appear in:	*Applied Soft Computing*
Received date:	20-11-2009
Revised date:	3-8-2010
Accepted date:	3-1-2011

Please cite this article as: L.D. Vignolo, H.L. Rufiner, D.H. Milone, J.C. Goddard, Evolutionary cepstral coefficients, *Applied Soft Computing Journal* (2008), doi:10.1016/j.asoc.2011.01.012

Evolutionary cepstral coefficients

Leandro D. Vignolo*, Hugo L. Rufiner, Diego H. Milone

Centro de Investigación y Desarrollo en Señales, Sistemas e Inteligencia Computacional, Departamento de Informática, Facultad de Ingeniería y Ciencias Hídricas, Universidad Nacional del Litoral, CONICET, Argentina

John C. Goddard

Departamento de Ingeniería Eléctrica, Iztapalapa, Universidad Autónoma Metropolitana, México

Abstract

Evolutionary algorithms provide flexibility and robustness required to find satisfactory solutions in complex search spaces. This is why they are successfully applied for solving real engineering problems. In this work we propose an algorithm to evolve a robust speech representation, using a dynamic data selection method for reducing the computational cost of the fitness computation while improving the generalisation capabilities. The most commonly used speech representation are the mel-frequency cepstral coefficients, which incorporate biologically inspired characteristics into artificial recognizers. Recent advances have been made with the introduction of alternatives to the classic mel scaled filterbank, improving the phoneme recognition performance in adverse conditions.

In order to find an optimal filterbank, filter parameters such as the central and side frequencies are optimised. A hidden Markov model is used as the classifier for the evaluation of the fitness for each individual. Experiments were conducted using real and synthetic phoneme databases, considering

*Corresponding author.
Centro de Investigación y Desarrollo en Señales, Sistemas e Inteligencia Computacional, Departamento de Informática, Facultad de Ingeniería y Ciencias Hídricas, Universidad Nacional del Litoral, Ciudad Universitaria CC 217, Ruta Nacional No 168 Km 472.4, TE: +54(342)4575233 ext 125, FAX: +54(342)4575224, Santa Fe (3000), Argentina.

Email address: ldvignolo@fich.unl.edu.ar (Leandro D. Vignolo)

URL: http://fich.unl.edu.ar/sinc (Leandro D. Vignolo)

different additive noise levels. Classification results show that the method accomplishes the task of finding an optimised filterbank for phoneme recognition, which provides robustness in adverse conditions.

Keywords:
Automatic speech recognition, evolutionary computation, phoneme classification, cepstral coefficients

1. Introduction

Automatic speech recognition (ASR) systems require a preprocessing stage to emphasize the key features of phonemes, thereby allowing an improvement in classification results. This task is usually accomplished using one of several different signal processing techniques such as filterbanks, linear prediction or cepstrum analysis [1]. The most popular feature representation currently used for speech recognition is mel-frequency cepstral coefficients (MFCC) [2]. MFCC is based on a linear model of voice production together with the codification on a psychoacoustic scale.

However, due to the degradation of recognition performance in the presence of additive noise, many advances have been conducted in the development of alternative noise-robust feature extraction techniques. Moreover, some modifications to the biologically inspired representation were introduced in recent years [3, 4, 5, 6]. For instance, Slaney introduced an alternative [7] to the feature extraction procedure. Skowronski and Harris [8, 9] introduced the human factor cepstal coefficients (HFCC), consisting in a modification to the mel scaled filterbank. They reported results showing considerable improvements over the MFCC. The weighting of MFCC according to the signal-to-noise ratio (SNR) on each mel band was proposed in [10]. For the same purpose, the use of Linear Discriminant Analysis in order to optimise a filterbank has been studied in [11]. In other works the use of evolutive algorithms have been proposed to evolve features for the task of speaker verification [12, 13]. Similarly, in [14] an evolutive strategy was introduced in order to find an optimal wavelet packet decomposition.

Then, the question arises if any of these alternatives is really optimal for this task. In this work we employ an evolutionary algorithm (EA) to find a better speech representation. An EA is an heuristic search algorithm inspired in nature, with proven effectiveness on optimisation problems [15]. We propose a new approach, called evolved cepstral coefficients (ECC), in which

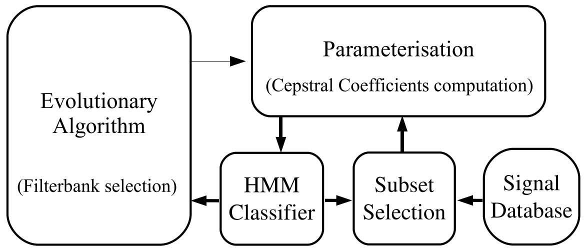


Figure 1: General scheme of the proposed method.

an EA is employed to optimise the filterbank used to calculate the cepstral coefficients (CC). The ECC approach is schematically outlined in Figure 1. To evaluate the fitness of each individual, we incorporate a hidden Markov model (HMM) based phoneme classifier. The proposed method aims to find an optimal filterbank, meaning that it results in a speech signal parameterisation which improves standard MFCC on phoneme classification results. Prior to this work, we obtained some preliminary results, which have been reported in [16].

A problem arises in this kind of optimisation because over-training might occur and resulting filterbanks could highly depend on the training data set. This problem could be overcome by increasing the amount of data, though, much more time or computational power would be needed for each experiment. In this work, instead, we incorporate a training subset selection method similar to the one proposed in [17]. This strategy enables us to train filterbanks with more patterns, allowing generalisation without increasing computational cost.

This paper is organized as follows. First we introduce some basic concepts about EAs and give a brief description of mel-frequency cepstral coefficients. Subsequently, the details of the proposed method are described and its implementation is explained. In the last sections, the results of phoneme recognition experiments are provided and discussed. Finally, some general conclusions and proposals for future work are given.

1.1. Evolutionary algorithms

Evolutionary algorithms are search methods based on the Darwinian theory of biological evolution [18]. This kind of algorithms present an implicit parallelism that may be implemented in a number of ways in order to increase the computational speed [14]. Usually an EA consists of three operations:

selection, variation and replacement [19]. Selection gives preference to better individuals, allowing them to continue to the next generation. The most common variation operators are crossover and mutation. Crossover combines information from two parent individuals into offspring, while mutation randomly modifies genes of chromosomes, according to some probability, in order to maintain diversity within the population. The replacement strategy determines which of the current members of the population, should be replaced by the new solutions. The population consists of a group of individuals whose information is coded in the so-called chromosomes, and from which the candidates are selected for the solution of a problem. Each individual performance is represented by its fitness. This value is measured by calculating the objective function on a decoded form of the individual chromosome (called the phenotype). This function simulates the selective pressure of the environment. A particular group of individuals (the parents) is selected from the population to generate the offspring by using the variation operators. The present population is then replaced by the offspring. The EA cycle is repeated until a desired termination criterion is reached (for example, a predefined number of generations, a desired fitness value, etc.). After the evolution process the best individual in the population is the proposed solution for the problem [20].

1.2. Mel-frequency cepstral coefficients

Mel-frequency cepstral coefficients are the most commonly used alternative to represent speech signals. This is mainly because the technique is well-suited for the assumptions of uncorrelated features used for the HMM parameter estimation. Moreover, MFCC provide superior noise robustness in comparison with the linear-prediction based feature extraction techniques [21].

The voice production model commonly used in ASR assumes that the speech signal is the output of a linear system. This means that the speech is the result of a convolution of an excitation signal, $x(t)$, with the impulse response of the vocal tract model, $h(t)$,

$$y(t) = x(t) * h(t), \tag{1}$$

where t stands for continuous time. In general only $y(t)$ is known, and it is frequently desirable to separate its components in order to study the features of the vocal tract response $h(t)$. Cepstral analysis solves this problem by

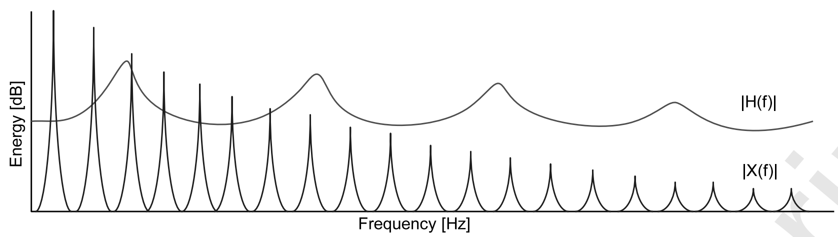


Figure 2: Magnitude spectrums of the excitation signal $X(f)$ and the vocal tract impulse response $H(f)$ from simulated voiced phonemes.

taking into account that if we compute the Fourier transform (FT) of (1) then the equation in the frequency domain is a product:

$$Y(f) = X(f)H(f), \tag{2}$$

where variable f stands for frequency, $X(f)$ is the excitation spectrum and $H(f)$ is the vocal tract frequency response. Then, by computing the logarithm from (2), this product is converted into a sum, and the real cepstrum $C(t)$ of a signal $y(t)$ is computed by:

$$C(t) = IFT\{\log_e |FT\{y(t)\}|\}, \tag{3}$$

where IFT is the inverse Fourier transform. This transformation has the property that its components, which were nonlinearly combined in time domain, are linearly combined in the cepstral domain. This type of homomorphic processing is useful in ASR because the rate of change of $X(f)$ and $H(f)$ are different from each other (Figure 2). Because of this property, the excitation and the vocal tract response are located at different places in the cepstral domain, allowing them to be separated. This is useful for classification because the information of phonemes is given only by $H(f)$.

In order to combine the properties of the cepstrum and the results about human perception of pure tones, the spectrum of the signal is decomposed into bands according to the mel scale. This scale was obtained through human perception experiments and defines a mapping between the physical frequency of a tone and the perceived pitch [1]. The mel scaled filterbank (MFB) is comprised of a number of triangular filters whose center frequencies are determined by means of the mel scale. The magnitude spectrum of the signal is scaled by these filters, integrated and log compressed to obtain a log-energy coefficient for each frequency band. The MFCC are the amplitudes

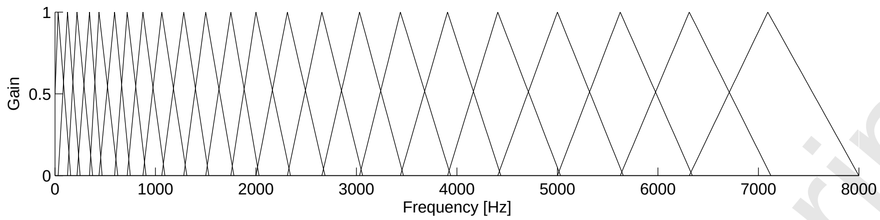


Figure 3: Mel scaled filterbank in the frequency range from 0 to 8kHz.

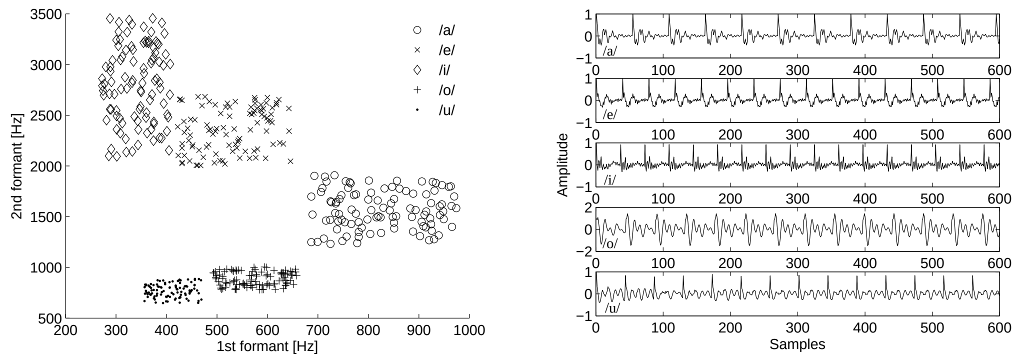
resulting from applying the IFT to the resulting sequence of log-energy coefficients [22]. However, because the argument of the IFT is a real and even sequence, the computation is usually simplified with the cosine transform (CT). Figure 3 shows a MFB comprised of 26 filters in the frequency range from 0 to 8 kHz. As it can be seen, endpoints of each filter are defined by the central frequencies of adjacent filters. Bandwidths of the filters are determined by the spacing of filter central frequencies which depend on the sampling rate and the number of filters. That is, if the number of filters increases, the number of MFCC increases and the bandwidth of each filter decreases.

2. MATERIALS AND METHODS

This section describes the proposed evolutionary algorithm, the speech data and the preprocessing method. First, the details about the speech corpus are given and the ECC method is explained. In the next subsection some considerations about the HMM based classifier are discussed and finally the data selection method for resampling training is explained.

2.1. Speech corpus and processing

For the experimentation, both synthetic and real phoneme databases have been used. In the first case, five Spanish vowels were modelled using the classical linear prediction coefficients [1], which were obtained from real utterances. We have generated different train, test and validation sets of signals which are 1200 samples in length and sampled at 8 kHz. Every synthetic utterance has a random fundamental frequency, uniformly distributed in the range from 80 to 250 Hz. In this way we simulate both male and female speakers. First and second resonant frequencies (formants) were randomly



(a) (b)

Figure 4: Synthetic phoneme database. a) First and second formant frequency distribution. b) Phoneme examples.

modified, within the corresponding ranges, in order to generate phoneme occurrences.

Our synthetic database included the five Spanish vowels /a/, /e/, /i/, /o/ and /u/, which can be simulated in a controlled manner.

Figure 4 shows the resulting formant distribution and some synthetic phoneme examples. White noise was generated and added to all these synthetic signals, so that the SNR of each signal is random and it varies uniformly from 2 dB to 10 dB. As these vowels are synthetic and sustained, the frames were extracted using a Hamming window of 50 milliseconds length (400 samples). The use of a synthetic database allowed us to maintain controlled experimental conditions, in which we could focus on the evolutive method, designed to capture the frequency features of the signals while disregarding temporal variations.

Real phonetic data was extracted from the TIMIT speech database [23]. Speech signals were selected randomly from all dialect regions, including both male and female speakers. Utterances were phonetically segmented to obtain individual files with the temporal signal of every phoneme occurrence. White noise was also added at different SNR levels. In this case, the sampling frequency was 16 kHz and the frames were extracted using a Hamming window of 25 milliseconds (400 samples) and a step-size of 200 samples. All possible frames within a phoneme occurrence were extracted and padded with zeros where necessary. The English phonemes /b/, /d/, /eh/, /ih/ and /jh/ were considered. The occlusive consonants /b/ and /d/ are included because they are very difficult to distinguish in different contexts. Phoneme /jh/ presents special features of the fricative sounds. Vowels /eh/ and /ih/ are commonly chosen because they are close in the formant space. This group of phonemes was selected because they constitute a set of classes which is difficult to classify [24].

For simplicity we introduced the steps for the computation of CC in the continuous time and frequency domains. Although, in practice we use digital signals and the discrete versions of the transforms mentioned in Section 1.2. For both MFCC and ECC the procedure is as follows. First, the spectrum of the frame is normalised and integrated by the triangular filters, and every coefficient resulting from integration is then scaled by the inverse of the area of the corresponding filter. As in the case of Slaney's filterbank [7], we give equal weight to all coefficients because this is shown to improve results. Then the discrete cosine transform (DCT) is computed from the log energy coefficients. As the number of filters n_f in each filterbank is not fixed, we set the number of output DCT coefficients to $[n_f/2] + 1$.

2.2. Evolutionary cepstral coefficients

The MFB shown in Figure 3, commonly used to compute cepstral coefficients, reveals that the search for an optimal filterbank can involve adjusting several of its parameters, such as: shape, amplitude, position and size of each filter. However, trying to optimise all the parameters together is extremely complex, so we decided to maintain some of the parameters fixed.

We carried out this optimisation in two different ways. In the first case, we considered non-symmetrical triangular filters, determined by three parameters each. These three parameters correspond to the frequency values where the triangle for the filter begins, where the triangle reaches its maximum, and where it ends. This is depicted in Figure 5, where the mentioned parameters are called a_i, b_i and c_i respectively. They are coded in the chromosome as integer values, indexing the frequency samples. The size and overlap between filters are left unrestricted in this first approach. The number of filters was also optimised by adding one more gene to the chromosome (n_f in Figure 5). This last element in the chromosome indicates that the first n_f filters are currently active. Hence, the length of each chromosome is three times the maximum number of filters allowed in a filterbank, plus one.

In a second approach, we decided to reduce the number of optimisation parameters. Here, triangular filters were distributed along the frequency

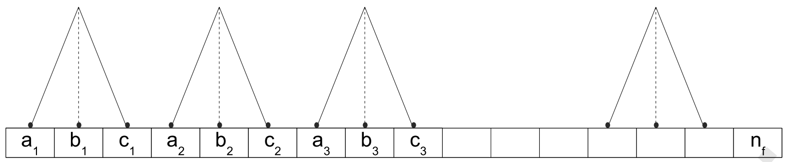


Figure 5: Scheme of the chromosome codification.

band, with the restriction of half overlapping. This means that only the central positions (parameters c_i in Figure 5) were optimised, and the bandwidth of each filter was adjusted by the preceding and following filters. In this case, the number of filters was optimised too.

In other approaches [13], polynomial functions were used to encode the parameters which were optimised. Here, in contrast, all the parameters are directly coded in the chromosome. In this way the search is simpler and the parameters are directly related to the features being optimised.

Each chromosome represents a different filterbank, and they are initialized with a random number of active filters. In the initialization, the position of the filters in a chromosome is also random and follows a discrete uniform distribution over the frequency bandwidth from 0 Hz to half the sampling frequency. The position, determined in this way, sets the frequency where the triangle of the filter reaches its maximum. Then, in the case of the three-parameter filters, a binomial distribution centred on this position is used to initialize the other two free parameters of the filter.

Before variation operators are applied, the filters in every chromosome are sorted by increasing order with respect to their central position. A chromosome is coded as a string of integers and the range of values is determined by the number of samples in the frequency domain.

The EA uses the roulette wheel selection method [25], and elitism is incorporated into the search due to its proven capabilities to enforce the algorithm's convergence under certain conditions [18]. The elitist strategy consists in maintaining the best individual from one generation to the next without any perturbation. The variation operators used in this EA are mutation and crossover, and they were implemented as follows. Mutation of a filter consists in the random displacement of one of its frequency parameters, and this modification is made using a binomial distribution. This mutation operator can also change, with the same probability, the number of filters in a filterbank. Our one-point crossover operator interchanges complete filters between different chromosomes. Suppose we are applying the crossover operator on two parents, for instance A and B. Then, if parent B contains more active filters than parent A, the crossover point is a random value between 1 and the n_f value of parent A. All genes (filters and n_f) beyond that point in either chromosome string are swapped between the two parents, resulting in an offspring with the same n_f of the first parent and an offspring with the same n_f of the second parent.

The selection of individuals is also conducted by considering the filterbank represented by a chromosome. The selection process should assign greater probability to the chromosomes providing the better signal representations, and these will be those that obtain better classification results. The proposed fitness function consists of a phoneme classifier, and the recognition rate will be the fitness value for the individual being evaluated.

2.3. HMM based classifier

In order to compare the results to those of state of the art speech recognition systems, we used a phoneme classifier based on HMM with Gaussian mixtures (GM). This fitness function uses tools from the HMM Toolkit [26] for building and manipulating hidden Markov models. These tools rely on the Baum-Welch algorithm [27] which is used to find the unknown parameters of an HMM, and on the Viterbi algorithm [28] for finding the most likely state sequence given the observed events in the recognition process.

Conventionally, the energy coefficients obtained from the integration of the log magnitude spectrum are transformed by the DCT to the cepstral domain. Besides the theoretical basis given on Section 1.2, this has the effect of removing the correlation between adjacent coefficients. Moreover, it also reduces the feature dimension.

Even though DCT has a fixed kernel and cannot decorrelate the data as thoroughly as data-based transforms [29], MFCC are close to decorrelated. The DCT produces nearly uncorrelated coefficients [30], which is desirable for HMM based speech recognizers using GM observation densities with diagonal covariance matrices [31].

2.4. Dynamic subset selection for training

A problem in evolutionary optimisation is that it requires enormous computational time. Usually, fitness evaluation takes the most time since it requires the execution of some kind of program against problem specific data. In our case, for instance, we need to train and test an HMM based classifier using a phoneme database. This implies that the time for the evolution is

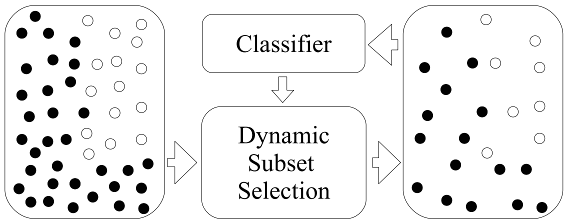


Figure 6: Scheme of the dynamic subset selection method.

proportional to the size of the data needed for fitness evaluation, as well as the population size and the number of generations. On the other hand, the data used for fitness evaluation dramatically influences the generalisation capability of the optimised solution. Hence, there is a trade off between the generalisation capability and the computational time.

In this work we propose the reduction of computational costs and the improvement of generalisation capability by evolving filterbank parameters on a selected subset of train and test patterns, which is changed during each generation. The idea of active data selection in supervised learning was originally introduced by Zhang et al. for efficient training of neural networks [32, 33]. Motivated by this work, Gathercole et al. introduced some training subset selection methods for genetic programming [17]. These methods are also useful in evolutionary optimisation, allowing us to significantly reduce the computation time while improving generalisation capability.

While in [17] only one training data set was considered, our subset selection method consists in changing the test subset, as well as the training subset, in every generation of the EA. For the test set, the idea is to focus the EA attention onto the cases that were mostly misclassified in previous generations and the cases that were not used recently.

In order to illustrate this, an example with two classes of two-dimensional patterns is outlined in Figure 6. The subset is selected from the original data set according to the classification results. The algorithm randomly selects a number of cases from the whole training and test sets every generation, and a test case has more probability to be selected if it is difficult or has not been selected for several generations. Another difference with the method proposed in [17] is that the size of test and train subsets remains strictly the same for all generations. In the first generation the testing subset is selected assigning the same probability to all cases. Then, during generation g, a weight $W_i(g)$ is determined for each test case i. This weight is the sum of the current difficulty of the case, $D_i(g)$, raised to the power d, and the age of the case, $A_i(g)$, raised to the power a,

$$W_i(g) = D_i(g)^d + A_i(g)^a. \tag{4}$$

The difficulty of a test case is given by the number of times it was misclassified and its age is the number of generations since it was last selected. Exponents d and a determine the importance given to *difficult* and *unselected* cases respectively. Given the sample size and other characteristics of the training data, these parameters are empirically determined. Each test case is given a probability $P_i(g)$ of being selected. This probability is given by its weight, multiplied by the size of the selected subset, S, and divided by the sum of the weights of all the test cases:

$$P_i(g) = \frac{W_i(g) * S}{\sum_j W_j(g)}. \tag{5}$$

When a test case i is selected, its age A_i is set to 1 and, if it is not selected, its age is incremented. While evaluating the EA population, difficulty D_i is incremented each time the case i is misclassified.

However, a problem arises when using an elitist strategy together with this method. As train and test subsets change, the best individual at a given time may no longer be the best one for the next generation. Although, probably it is still a good individual, we decided to maintain the best chromosome from the previous generation and assign the classification result from the current subset as its fitness.

3. Results and discussion

3.1. Synthetic Spanish phonemes

We conducted different EA runs and we found the best results when we evolved only the central filter positions and the number of filters, which we allowed to vary from 17 to 32. For the EA, the population size was set to 100 individuals and crossover rate was set to 0.8. The mutation rate, meaning the probability of a filter to have one of its parameters changed, was set to 0.1.

During the EA runs we used a set of 500 training signals and a different set of 500 test signals to compute the fitness for every individual. In this case, training and testing sets remained unchanged during the evolution. Each

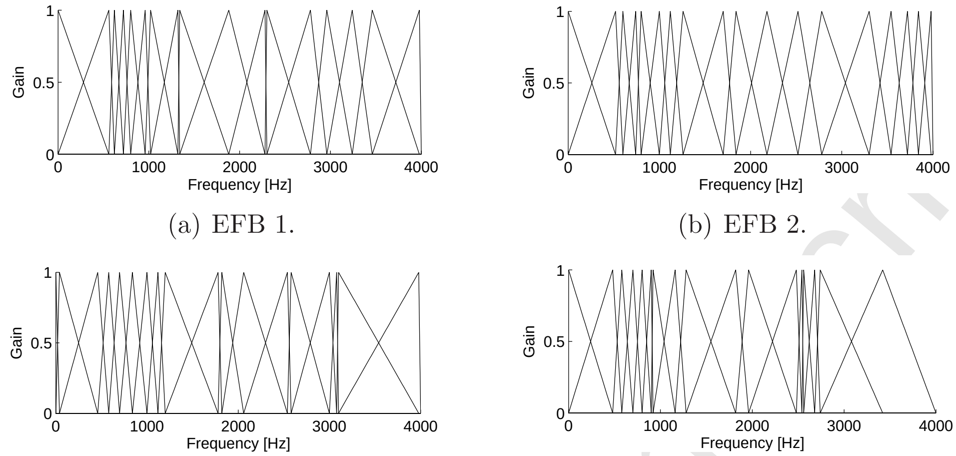
Table 1: Average classification rates (percent) for synthetic phonemes.

FB	# filters	# coeff	Validation test	
			DCM	FCM
EFB 1	17	9	*95.20*	**97.00**
EFB 2	18	10	**95.40**	*96.80*
EFB 3	18	10	93.00	*96.40*
EFB 4	17	9	94.60	96.20
MFB	23	13	94.80	96.20
MFB	17	9	93.00	95.20

run was terminated after 100 generations without any fitness improvement. When a run was finished, we took the twenty best filterbanks according to their fitness, and we made a validation test with another set of 500 signals. From this validation test we selected the two best filterbanks, discarding those that were over-optimised (those with higher fitness but with lower validation result).

Table 1 summarizes the validation results for filterbanks from two different optimisations, and includes the classification results obtained using the standard MFB on the same data sets. The fourth column contains the classification results obtained when using an HMM with diagonal covariance matrices (DCM), and the fifth column contains the results obtained when using an HMM with full covariance matrices (FCM). Evolved filterbanks (EFB) 1 and 2 were obtained using HMM with DCM as fitness during the optimisation, while EFBs 3 and 4 were obtained using HMM with FCM. It can be observed that we obtained filterbanks that perform better than MFB when using FCM-HMM. Also, it is important to notice that MFB also performs better using FCM-HMM.

Figure 7 shows these four EFBs. One feature they all have in common is the high density of filters from approximately 500 to 1000 Hz, which could be related to the distribution of the first frequency formant (Figure 4). Moreover, considering the second formant frequency, it can be noticed that these groups of filters could distinguish phonemes /o/ and /u/ from the others. Another common trait in these four filterbanks is that the frequency range from 0 to 500 Hz is covered by only two filters, although, in EFB 3 there is a narrow filter from 0 to 40 Hz, besides these two. This narrow filter isolates the peaks at zero frequency from the phoneme information. Another likeness



(a) EFB 1.

(b) EFB 2.

(c) EFB 3.

(d) EFB 4.

Figure 7: Filterbanks optimised for phonemes /a/, /e/, /i/, /o/ and /u/ from our synthetic database.

is that, in the band from approximately 1000 to 2500 Hz, the four filterbanks show similar filter distribution. On the other hand, a feature which is present only in the second filterbank is the attention given to high frequencies, as opposed to MFB, and taking higher formants into account.

3.2. Real English phonemes

In the second group of experiments the best results were obtained when considering non-symmetrical triangular filters, determined by three parameters each. Also in this case, the number of filters in the filterbanks was allowed to vary from 17 to 32. For the fitness computation we used a dynamic data partition of 1000 training signals and 400 test signals, and an HMM based classifier with FCM. The data partition used during the EA runs was changed every generation according to the strategy described in Section 2.4, and phoneme samples were dynamically selected from a total of 6045 signals available for training and 1860 signals available for testing. As mentioned in Section 2.4, some preliminary experiments were carried out in order to set difficulty and age exponents (parameters d and a in equation 4). Given the sample size and using different combinations, we found that a good choice is to set both parameters d and a to 1.0.

As in the experiments with synthetic phonemes, a EA run was ended

Table 2: Classification rates for English phonemes (percent). Average over ten train/test partitions. Filterbanks optimised at 0 dB SNR.

FB	# filters	# coeff	-5dB	0dB	20dB	clean	Diff
A0	32	17	24.76	32.62	58.26	65.54	0.44
A1	17	9	20.26	26.02	62.16	62.62	−9.68
A2	21	11	20.16	21.34	59.56	60.00	−19.68
A3	29	15	24.34	32.92	66.08	64.32	6.92
A4	19	10	20.38	26.32	63.64	61.22	−9.18
A5	19	10	20.52	26.24	60.62	60.26	−13.10
A6	21	11	**31.10**	**35.78**	61.52	60.80	**8.46**
A7	29	15	22.58	30.52	63.90	64.58	0.84
A8	25	13	22.94	30.76	62.10	62.08	−2.86
A9	22	12	23.60	31.54	63.54	66.14	4.08
MFB	23	13	20.00	23.18	**68.40**	**69.16**	

after 100 generations without any fitness improvement, and we took the ten best filterbanks according to their fitness. The settings for the parameters of the EA were also the same values given in Section 3.1. We made validation tests with ten different data partitions consisting of 2500 train patterns and 500 test patterns each. Moreover, these validation tests were made using test sets at different SNR levels.

Here we show the classification results of filterbanks obtained from three EA runs which only differ in the noise level used for train and test sets for the fitness computation. Table 2 shows average classification results comparing filterbanks optimised for signals at 0 dB SNR against standard MFB, using DCM-HMM. We tested the best ten EFBs at different SNR, always training the classifier with clean signals. Each one of these results were obtained as the average of the classification with ten different data partitions. The last column gives the accumulated difference between each of the first ten rows and the last row, the higher values indicate the best filterbanks. For example, in Table 2, we obtain the value 0.44 in the first row by adding the difference of the values from column 4 to column 7 in the first row, from those in row 11. As the number of filters is one of the optimised parameters, we compare all the EFBs against a MFB composed of 23 filters, which is a standard setup in speech recognition. It can be seen that when testing at -5 and 0 dB SNR the EFB A6 performs much better than MFB. From this we can assume that the distribution of filters in EFB A6 allows to distinguish better the formant

Table 3: Classification rates for English phonemes (percent). Average over ten train/test partitions. Filterbanks optimised at 20 dB SNR.

FB	# filters	# coeff	-5dB	0dB	20dB	clean	Diff
B0	20	11	20.04	22.24	62.30	63.06	−13.10
B1	19	10	22.18	30.06	53.76	64.12	−10.62
B2	22	12	22.44	30.24	60.68	64.96	−2.42
B3	19	10	21.38	27.84	68.08	67.80	4.36
B4	19	10	21.10	26.72	62.40	64.52	−6.00
B5	19	10	22.06	**34.54**	55.56	64.46	−4.12
B6	18	10	20.22	31.92	**68.44**	66.64	**6.48**
B7	19	10	**22.88**	31.98	64.44	67.26	5.82
B8	18	10	21.58	27.90	64.04	61.88	−5.34
B9	19	10	22.82	31.08	64.28	68.04	5.48
MFB	23	13	20.00	23.18	68.40	**69.16**	

frequencies from the noise frequency components. This means that the use of the evolved filterbank results in features which are more robust than the standard parameterisation.

The same comparison is made in Tables 3 and 4 for filterbanks optimised using signals at 20 dB SNR and clean signals respectively. Again, we can see that some EFBs perform considerably better than the MFB with noisy test signals, and there is even an improvement at 20 dB SNR in these cases.

From these three groups of EFBs we selected some of the best EFBs and further tested them at 5, 10, 15 and 30 dB SNR. The average results from ten data partitions can be found in Table 5, as well as the results for the MFB, HFCC and Slaney filterbanks. For the HFCC 30 filters were considered, one filter was added to the filterbank proposed in [34] because the sampling frequency used in our experiments is higher. The bandwiths of the filters in HFCC are controlled by a parameter called E-factor, which was set to 5, based on the recognition results shown in [34]. As suggested, the first 13 cepstral coefficients were considered. The Slaney filterbank was comprised of 40 filters, as proposed in [7], and 20 cepstral coefficients were computed.

It can be seen that the EFBs perform better than the standard MFB when the SNR in testing signals is lower than the SNR in the training signals. Moreover, EFB C4 and EFB B6 outperform the Slaney filterbank in all noise conditions considered except in the case of −5 dB SNR. On the other hand, the EFBs perform better than the HFCC filterbank at the lower SNRs,

Table 4: Classification rates for English phonemes (percent). Average over ten train/test partitions. Filterbanks optimised for clean signals.

FB	# filters	# coeff	-5dB	0dB	20dB	clean	Diff
C0	21	11	20.56	27.94	64.14	63.48	−4.62
C1	18	10	20.08	34.20	61.26	60.66	−4.54
C2	19	10	20.28	27.74	62.62	60.72	−9.38
C3	18	10	21.94	30.32	62.70	64.36	−1.42
C4	18	10	20.56	**36.88**	**69.82**	68.08	**14.60**
C5	18	10	22.26	30.42	65.14	63.40	0.48
C6	19	10	20.30	30.16	64.82	62.62	−2.84
C7	18	10	20.16	30.66	63.22	61.96	−4.74
C8	18	10	**26.52**	33.56	56.62	64.00	−0.04
C9	18	10	20.40	26.68	66.88	66.22	−0.56
MFB	23	13	20.00	23.18	68.40	**69.16**	

this is from −5 dB to 15 dB SNR. These improvements may be better visualized in Figure 8, where it is easy to appreciate that EFB C4 outperforms MFB in the range from 0 dB to 15 dB SNR. It can be seen that MFB is not outperformed for 30 dB SNR and clean signals, however this behaviour is common to most robust ASR methods [35]. For instance, the HFCC filterbank outperform MFB for noisiest cases, however, above 20 dB SNR the improvements are smaller. Moreover, the degradation of recognition performance is proportional to the mismatch between the SNR of the training set and the SNR of the test set [36, 4].

Figure 9 shows the selected EFBs from Table 5. As we stated before, one feature they all have in common is the wide bandwidth of most of the filters, compared with the MFB. This coincides with the study in [34] about the effect of wider filter bandwidth on noise robustness. In all the EFBs we can also see high overlapping between different filters, as there was not any constraint about this in the optimisation. However, this high overlapping which results in correlated CC could be beneficial for classification with full covariance matrix HMM. We can observe the grouping of a relatively high number of filters in the frequency band from 0 Hz to 4000 Hz in the case of EFB C4, which gives the best results for noisy test signals.

In order to analyse what information these representations are capturing, we recovered an estimate of the short-time magnitude spectrum using the method proposed in [37]. Which consists in scaling the spectrogram of

Table 5: Classification rates for English phonemes (percent). Average over ten train/test partitions.

FB	-5dB	0dB	5dB	10dB	15dB	20dB	30dB	clean
A3	24.34	32.92	37.68	46.36	52.98	66.08	65.04	64.32
A6	**31.10**	35.78	44.38	46.88	53.12	61.52	60.36	60.80
B6	20.22	31.92	55.12	67.20	**68.84**	68.44	67.20	66.64
B7	22.88	31.98	36.86	44.42	49.64	64.44	67.58	67.26
C4	20.56	**36.88**	**60.30**	**68.32**	68.70	69.82	67.42	68.08
C5	22.26	30.42	34.38	44.32	57.28	65.14	63.52	63.40
MFB	20.00	23.18	37.90	44.68	51.42	68.40	**69.80**	69.16
HFCC	20.24	25.98	47.26	62.78	67.68	**70.54**	69.42	**70.36**
Slaney	29.94	30.28	36.44	54.76	60.66	62.02	61.52	62.78

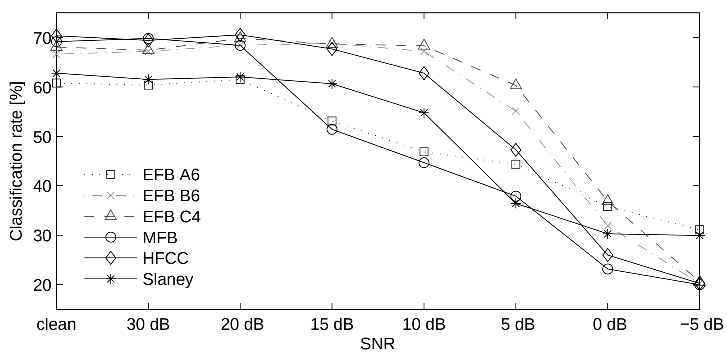
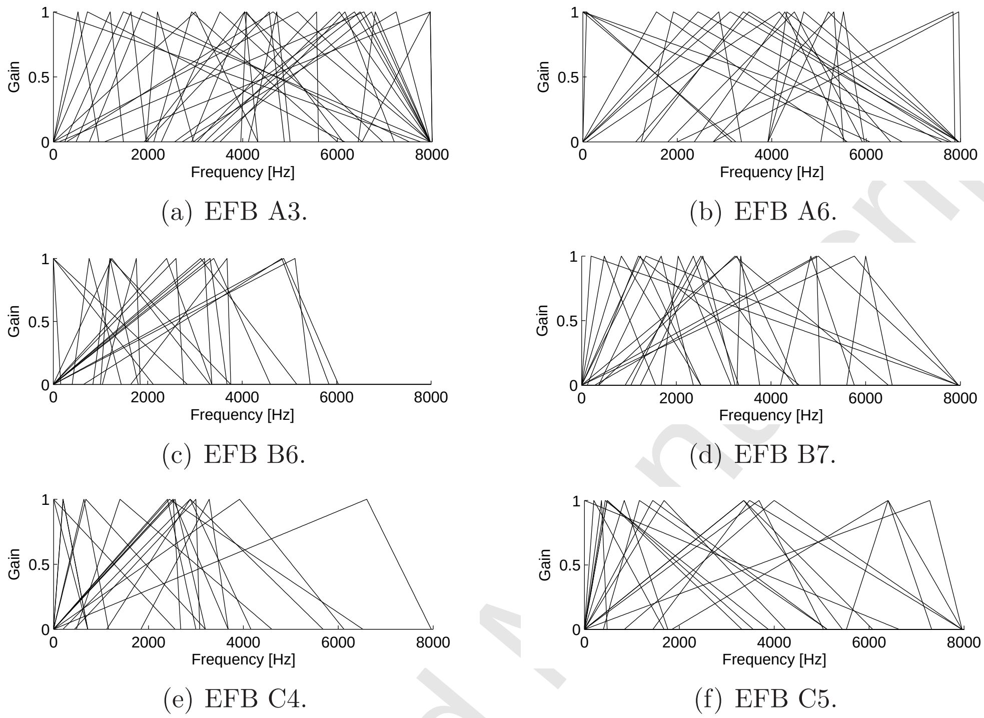


Figure 8: Performance of the best EFBs compared with MFB (English phonemes).

a white noise signal by the short-time magnitude spectrum recovered from the cepstral coefficients. Figures 10 and 11 show the spectrograms of sentence `SI648` from TIMIT corpus, with additive noise at 50 dB and 10 dB SNR respectively. Figure 10 shows that wide filters of the EFB blur energy coefficients along the frequency axis, and it is more difficult to notice the formant frequencies, though this information is not lost. Moreover, phoneme classification is made easier by removing information related to pitch. On the other hand, from Figure 11 it can be seen that when the signal is noisy, the relevant information is clearer in the spectrogram reconstructed from ECC. This is because the filter distribution and bandwidths of EFB C4 allow the relevant information on higher frequencies to be conserved, which is hidden by noise when using MFCC.



(a) EFB A3. (b) EFB A6.
(c) EFB B6. (d) EFB B7.
(e) EFB C4. (f) EFB C5.

Figure 9: Filterbanks optimised for phonemes /b/, /d/, /eh/, /ih/ and /jh/ from TIMIT database.

Table 6 exhibits the confusion matrices for MFB and EFB C4, obtained when testing with signals at 10 and 15 dB SNR. From these matrices, it can be seen that phonemes /eh/ and /ih/ are mostly misclassified using MFB and they are frequently well classified using EFB C4. In fact, when the SNR is high, the performance in the classification of each of the five phonemes is similar for both MFB and EFB C4. However, the lower the SNR, the more MFB fails to classify phonemes /eh/ and /ih/. These are mostly confused with phonemes /b/ and /d/, while the success rate for phonemes /b/, /d/ and /jh/ is barely affected. On the other hand, when using EFB C4 the effect of noise degrades the success rate for all phonemes uniformly, but none of them are as confused as in the case of MFB. That is, not only the average of success rate is higher, but also the variance between phonemes is lower. This means that the evolved filterbank provides a more robust parameterisation as it achieves better classification results in the presence of noise.

Table 6: Confusion matrices. Average classification rates (percent) from ten data partitions.

		MFB					EFB C4				
		/b/	/d/	/eh/	/ih/	/jh/	/b/	/d/	/eh/	/ih/	/jh/
15 dB	/b/	***64.7***	34.8	00.0	00.0	00.5	***56.9***	39.7	01.8	01.4	00.2
	/d/	11.7	***83.2***	00.0	00.1	5.00	14.1	***79.9***	00.6	00.9	04.5
	/eh/	33.1	51.0	***05.0***	07.1	03.8	03.9	04.5	***73.5***	18.1	00.0
	/ih/	21.8	45.3	04.7	***18.9***	09.3	12.6	09.9	18.2	***59.3***	00.0
	/jh/	00.1	14.6	00.0	00.0	***85.3***	00.3	25.3	00.2	00.3	***73.9***
					Avg:	51.42				Avg:	**68.70**
10 dB	/b/	***55.4***	44.0	00.0	00.0	00.6	***48.8***	48.6	01.5	00.5	00.6
	/d/	07.4	***89.2***	00.0	00.0	30.4	08.2	***86.4***	00.0	00.0	05.4
	/eh/	25.6	70.6	***00.0***	00.0	30.8	03.7	06.5	***77.4***	12.4	00.0
	/ih/	13.5	68.6	00.0	***00.0***	17.9	09.1	10.3	22.9	***57.7***	00.0
	/jh/	00.0	21.2	00.0	00.0	***78.8***	00.2	28.3	00.0	00.2	***71.3***
					Avg:	44.68				Avg:	**68.32**

3.3. Statistical dependence of ECC

As we mentioned in Section 2.3, MFCC are almost uncorrelated and are suitable for the utilization of HMM. However, this assumption of weak statistical dependence may not be true for the ECC. As Figure 9 shows, filter bandwidth and overlapping is usually higher for the optimised filterbanks than MFB. This means that the energy coefficients contain highly redundant information, and DCT may not be enough to obtain near decorrelated coefficients in this case. In fact, we have studied and compared the statistical dependence of MFCC and ECC, and noticed that optimised coefficients show, in general, higher correlation. Figure 12 shows the correlation matrices of 10 cepstral coefficients computed over 1500 frames. In order to make this comparison, we used a MFB consisting on 18 filters, the same number of filters in the optimised filterbank named C4. Correlation coefficients corresponding to MFB are shown on top and those corresponding to the optimised filterbank C4 at the bottom. As can be seen, correlation matrices show high statistical dependence between cepstral coefficients corresponding to phonemes /eh/ and /ih/, and this is much more noticeable for the case of the evolved filterbank. In order to obtain a measure of the statistical dependence, the sum of the correlation coefficients for each phoneme was obtained. These values can be seen on Table 7, and they were computed as $\sum_i \sum_j |M_{i,j}| - \mathrm{trace}(|M|)$, where M is the matrix of correlation coefficients. From these values we can also see that ECC are more correlated than the

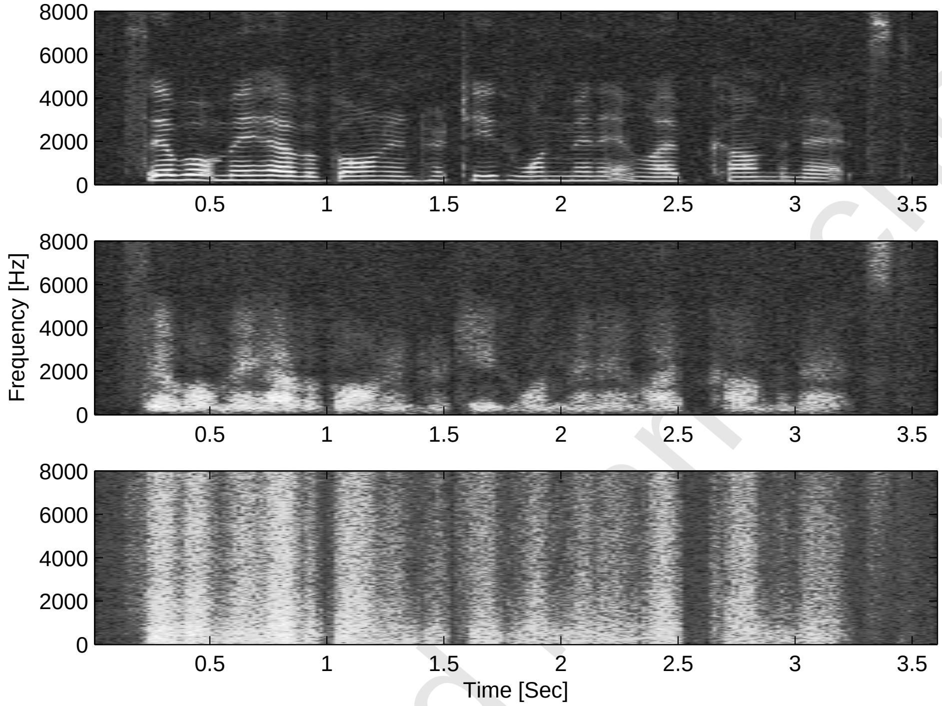


Figure 10: Spectrograms for sentence SI648 from TIMIT corpus at 50dB SNR. Computed from the original signal (top), reconstructed from MFCC (middle) and reconstructed from ECC (bottom).

MFCC for the set of phonemes we have considered.

The statistical dependence which is present in ECC implies that GM observation densities with diagonal covariance matrices (DCM) may not be the best option. Hence we decided to use full covariance matrices instead, to model the observation density functions during the optimisation. Moreover, as the MFCC are not completely decorrelated, they also allowed the classifier to perform better when using full covariance matrices (FCM) (See Table 1).

4. Conclusion and future work

A new method has been proposed for evolving a filterbank, in order to produce a cepstral representation that improves the classification of noisy speech signals. Our approach successfully exploits the advantages of evolutionary computation in the search for an optimal filterbank. Free parameters

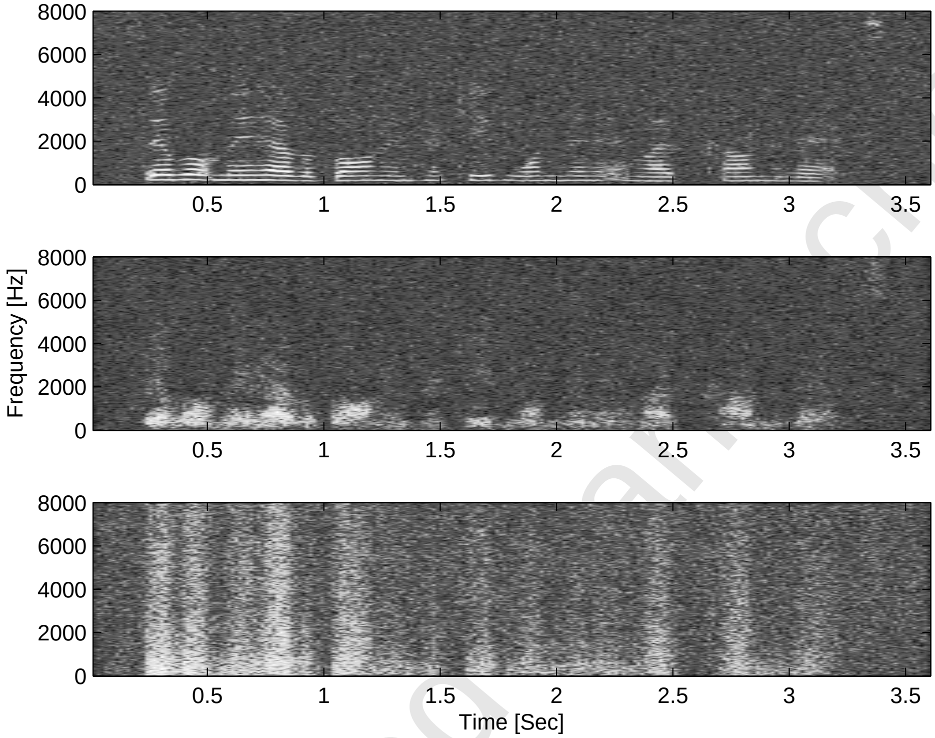


Figure 11: Spectrograms for sentence `SI648` from TIMIT corpus at 10dB SNR. Computed from the original signal (top), reconstructed from MFCC (middle) and reconstructed from ECC (bottom).

and codification provided a wide search space, which was covered by the algorithm due to the design of adequate variation operators. Moreover, the data selection method for resampling prevented the overfitting without increasing computational cost.

The obtained representation provides a new alternative to classical approaches, such as those based on a mel scaled filterbank or linear prediction, and may be useful in automatic speech recognition systems. Experimental results show that the proposed approach meets the objective of finding a more robust signal representation. This approach facilitates the task of the classifier because it properly separates the phoneme classes, thereby improving the classification rate when the test noise conditions differ from the training noise conditions. Moreover, with the use of this optimal filterbank the robustness

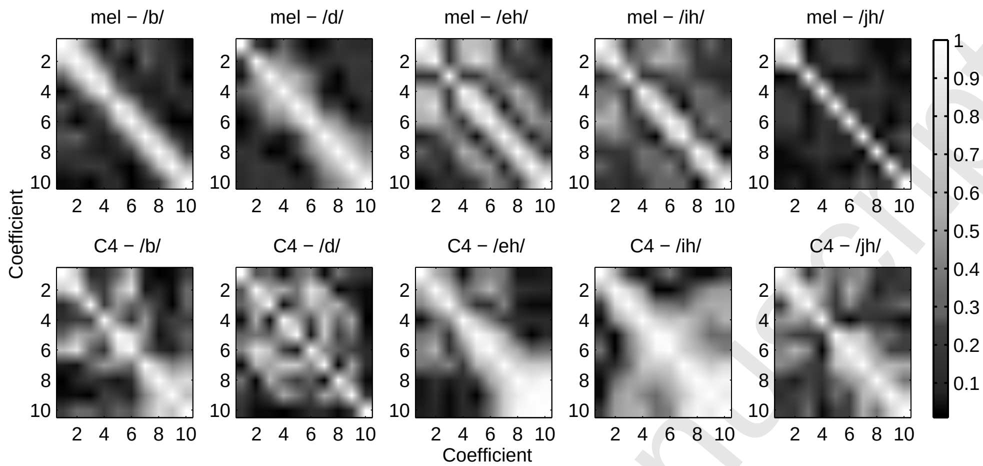


Figure 12: Correlation matrices of MFCC (top) and ECC (bottom).

Table 7: Sum of correlation coefficients.

	/b/	/d/	/eh/	/ih/	/jh/
MFB	20.9	24.9	30.4	27.2	11.2
C4	28.8	27.5	33.1	45.5	32.2

of an ASR system can be improved with no additional computational cost. These results also suggest that there is further room for improvement over the psychoacoustic scaled filterbank.

In future work, the utilisation of other search methods, such as particle swarm optimisation and scatter search will be studied. Different variation operators can also be considered as a way to improve the results of the EA. Moreover, the search for an optimal filterbank could be carried out by evolving different parameters. The possibility of replacing the HMM based classifier by another objective function, in order to reduce computational cost, will also be studied. In particular, we will consider fitness functions which incorporate information such as the gaussianity and the correlation of the coefficients, as well as the class separability.

References

[1] L. Rabiner, B.-H. Juang, Fundamentals of Speech Recognition, Prentice Hall PTR, 1993.

[2] S. V. Davis, P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, IEEE Transactions on Acoustics, Speech and Signal Processing 28 (1980) 57–366.

[3] B. Nasersharif, A. Akbari, SNR-dependent compression of enhanced Mel sub-band energies for compensation of noise effects on MFCC features, Pattern Recognition Letters 28 (11) (2007) 1320 – 1326, advances on Pattern recognition for speech and audio processing.

[4] X. Zhou, Y. Fu, M. Liu, M. Hasegawa-Johnson, T. Huang, Robust Analysis and Weighting on MFCC Components for Speech Recognition and Speaker Identification, in: Multimedia and Expo, 2007 IEEE International Conference on, 2007, pp. 188–191.

[5] H. Bŏril and P. Fousek and P. Pollák, Data-Driven Design of Front-End Filter Bank for Lombard Speech Recognition, in: Proc. of INTERSPEECH 2006 - ICSLP, Pittsburgh, Pennsylvania, 2006, pp. 381–384.

[6] Z. Wu, Z. Cao, Improved MFCC-Based Feature for Robust Speaker Identification, Tsinghua Science & Technology 10 (2) (2005) 158 – 161.

[7] M. Slaney, Auditory Toolbox, Version 2, Technical Report 1998-010, Interval Research Corporation, Apple Computer Inc. (1998).

[8] M. Skowronski, J. Harris, Increased MFCC filter bandwidth for noise-robust phoneme recognition, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 1 (2002) 801–804.

[9] M. Skowronski, J. Harris, Improving the filter bank of a classic speech feature extraction algorithm, in: Proceedings of the 2003 International Symposium on Circuits and Systems (ISCAS), Vol. 4, 2003, pp. 281–284.

[10] H. Yeganeh, S. Ahadi, S. Mirrezaie, A. Ziaei, Weighting of Mel Sub-bands Based on SNR/Entropy for Robust ASR, in: Signal Processing and Information Technology, 2008. ISSPIT 2008. IEEE International Symposium on, 2008, pp. 292–296.

[11] L. Burget, H. Heřmanský, Data Driven Design of Filter Bank for Speech Recognition, in: Text, Speech and Dialogue, Lecture Notes in Computer Science, Springer, 2001, pp. 299–304.

[12] C. Charbuillet, B. Gas, M. Chetouani, J. Zarader, Optimizing feature complementarity by evolution strategy: Application to automatic speaker verification, Speech Communication 51 (9) (2009) 724 – 731, special issue on non-linear and conventional speech processing.

[13] C. Charbuillet, B. Gas, M. Chetouani, J. Zarader, Multi Filter Bank Approach for Speaker Verification Based on Genetic Algorithm, Lecture Notes in Computer Science, 2007, pp. 105–113.

[14] L. Vignolo, D. Milone, H. Rufiner, E. Albornoz, Parallel implementation for wavelet dictionary optimization applied to pattern recognition, in: Proceedings of the 7th Argentine Symposium on Computing Technology, Mendoza, Argentina, 2006.

[15] D. B. Fogel, Evolutionary computation, John Wiley and Sons, 2006.

[16] L. Vignolo, H. Rufiner, D. Milone, J. Goddard, Genetic optimization of cepstrum filterbank for phoneme classification, in: Proceedings of the Second International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS 2009), INSTICC Press, Porto (Portugal), 2009, pp. 179–185.

[17] C. Gathercole, P. Ross, Dynamic training subset selection for supervised learning in genetic programming, in: Parallel Problem Solving from Nature – PPSN III, Lecture Notes in Computer Science, Springer, 1994, pp. 312–321.

[18] T. Bäck, Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms, Oxford University Press, Oxford, UK, 1996.

[19] T. Bäck, U. Hammel, H.-F. Schewfel, Evolutionary computation: Comments on history and current state, IEEE Trans. on Evolutionary Computation 1 (1) (1997) 3–17.

[20] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, 1992.

[21] C. R. Jankowski, H. D. H. Vo, R. P. Lippmann, A comparison of signal processing front ends for automatic word recognition, IEEE Transactions on Speech and Audio Processing 4 (3) (1995) 251–266.

[22] J. R. Deller, J. G. Proakis, J. H. Hansen, Discrete-Time Processing of Speech Signals, Macmillan Publishing, NewYork, 1993.

[23] J. S. Garofalo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, DARPA TIMIT acoustic phonetic continuous speech corpus CD-ROM, Tech. rep., U.S. Dept. of Commerce, NIST, Gaithersburg, MD (1993).

[24] K. N. Stevens, Acoustic Phonetics, Mit Press, 2000.

[25] A. E. Eiben, J. E. Smith, Introduction to Evolutionary Computing, SpringerVerlag, 2003.

[26] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, P. Woodland, HMM Toolkit, Cambridge University (2000).
URL http://htk.eng.cam.ac.uk

[27] F. Jelinek, Statistical Methods for Speech Recognition, MIT Press, Cambrige, Masachussets, 1999.

[28] X. D. Huang, Y. Ariki, M. A. Jack, Hidden Markov Models for Speech Recognition, Edinburgh University Press, 1990.

[29] C. Wang, L. M. Hou, Y. Fang, Individual Dimension Gaussian Mixture Model for Speaker Identification, in: Advances in Biometric Person Authentication, 2005, pp. 172–179.

[30] O.-W. Kwon, T.-W. Lee, Phoneme recognition using ICA-based feature extraction and transformation, Signal Process. 84 (6) (2004) 1005–1019.

[31] K. Demuynck, J. Duchateau, D. Van Compernolle, P. Wambacq, Improved Feature Decorrelation for HMM-based Speech Recognition, in: Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP 98), Sydney, Australia, 1998.

[32] B.-T. Zhang, G. Veenker, Focused incremental learning for improved generalization with reduced training sets, in: T. Kohonen (Ed.), Proc.

Int. Conf. Artificial Neural Networks, Vol. 1585, North-Holland, 1991, pp. 227–232.

[33] B.-T. Zhang, D.-Y. Cho, Genetic Programming with Active Data Selection, in: Lecture Notes in Computer Science, Vol. 1585, 1999, pp. 146–153.

[34] M. Skowronski, J. Harris, Exploiting independent filter bandwidth of human factor cepstral coefficients in automatic speech recognition, The Journal of the Acoustical Society of America 116 (3) (2004) 1774–1780.

[35] Y. Gong, Speech recognition in noisy environments: a survey, Speech Commun. 16 (3) (1995) 261–291.

[36] G. M. Davis, Noise reduction in speech applications, CRC Press, 2002.

[37] D. P. W. Ellis, PLP and RASTA (and MFCC, and inversion) in Matlab, online web resource (2005).
URL www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/

Figure 13: Leandro Daniel Vignolo.

LEANDRO D. VIGNOLO was born in San Genaro Norte (Santa Fe), Argentina, in 1981. In 2004 he joined the Laboratory for Signals and Computational Intelligence, in the Department of Informatics, National University of Litoral (UNL), Argentina. He is a teaching assistant at UNL, and he received the Computer Engineer degree from UNL in 2006. He received a Scholarship from the Argentinean National Council of Scientific and Technical Research, and he is currently pursuing the Ph.D. at the Faculty of Engineering and Water Sciences, UNL. His research interests include pattern recognition, signal processing, neural and evolutionary computing, with applications to speech recognition.

HUGO L. RUFINER was born in Buenos Aires, Argentina, in 1967. He received the Bioengineer degree (Hons.) from National University of Entre Ríos, in 1992, the M.Eng. degree (Hons.) from the Metropolitan Autonomous University, Mexico, in 1996 and the Dr.Eng. degree from the University of Buenos Aires in 2005. He is a Full Professor of the Department of Informatics, National University of Litoral and Adjunct Research Scientist at the National Council of Scientific and Technological Research. In 2006, he was awarded by the National Academy of Exact, Physical and Natural Sciences of Argentina. His research interests include signal processing, artificial intelligence and bioengineering.

DIEGO H. MILONE was born in Rufino (Santa Fe), Argentina, in 1973. He received the Bioengineer degree (Hons.) from National University of Entre Rios, Argentina, in 1998, and the Ph.D. degree in Microelectronics and Computer Architectures from Granada University, Spain, in 2003. Currently, he is Full Professor and Director of the Department of Informatics at

Figure 14: Hugo Leonardo Rufiner.

Figure 15: Diego Humberto Milone.

Figure 16: John C. Goddard.

National University of Litoral and Adjunct Research Scientist at the National Council of Scientific and Technological Research. His research interests include statistical learning, pattern recognition, signal processing, neural and evolutionary computing, with applications to speech recognition, computer vision, biomedical signals and bioinformatics.

JOHN C. GODDARD received a B.Sc (1st Class Hons) from London University and a Ph.D in Mathematics from the University of Cambridge. He is a Professor in the Department of Electrical Engineering at the Universidad Autónoma Metropolitana in Mexico City. His areas of interest include pattern recognition and heuristic algorithms applied to optimization problems.