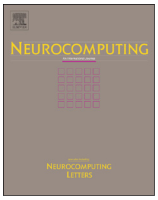
Neurocomputing 149 (2015) 426–434

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Acoustic template-matching for automatic emergency state detection: An ELM based algorithm

Emanuele Principi [a],*, Stefano Squartini [a], Erik Cambria [b], Francesco Piazza [a]

[a] *Department of Information Engineering, Università Politecnica delle Marche, Ancona, Italy*
[b] *School of Computer Engineering, Nanyang Technological University*

ARTICLE INFO

Article history:
Received 6 August 2013
Received in revised form
24 December 2013
Accepted 19 January 2014
Available online 7 September 2014

Keywords:
Extreme Learning Machine
Automatic speech recognition
Support Vector Machine
Dynamic time warping
Template matching

ABSTRACT

Extreme Learning Machine (ELM) represents a popular paradigm for training feedforward neural networks due to its fast learning time. This paper applies the technique for the automatic classification of speech utterances. Power Normalized Cepstral Coefficients (PNCC) are employed as feature vectors and ELM performs the final classification. Both the baseline ELM algorithm and ELM with kernel have been employed and tested. Due to the fixed number of input neurons in the ELM, a length normalization algorithm is employed to transform the PNCC sequence into a vector of fixed length. Length normalization has been performed using two techniques: the first is based on Dynamic Time Warping (DTW) distances, the second on the vectorized outerproduct of trajectory matrix. Experiments have been conducted on the TIDIGITS corpus, to assess the performance on an isolated speech recognition task, and on ITAAL, to validate the system in an emergency detection task in realistic acoustic conditions. The ELM approach has been compared to template matching based on Dynamic Time Warping and to a Support Vector Machine based speech recognizer. The obtained results demonstrated the effectiveness of the approach both in terms of recognition performance and execution times. In particular, classification based on PNCCs, DTW distances and ELM kernel resulted in the best performing algorithm both in terms of recognition accuracy and execution times.

1. Introduction

Nowadays, the majority of automatic speech recognizers are based on hidden Markov models (HMM) [1]. Although HMMs provide state-of-the-art performance in several scenarios, alternative approaches such as template-matching [2,40] and discriminative techniques [3,4,41] are also widely studied. The reasons for template-matching are its low storage requirements and its effectiveness when the amount of training data is limited. Generally, in template-matching, sequences of different lengths are aligned using dynamic time warping (DTW) [2] and classification is based on the distance with a set of reference patterns. The problems with the original DTW formulation are its high computational burden, the low performance in speaker independent tasks and the discrimination between in-domain and out-of-domain sentences. In the literature, particular attention has been devoted to develop efficient versions of DTW for devices with limited computational resources [5–7].

Among discriminative techniques, in the recent years particular attention has been devoted to speech recognition based on Support Vector Machines (SVMs) [8,9]. SVMs have been originally developed to solve binary classification problems of sequences of fixed length, but they can be easily extended to multiclass tasks, e.g., using the “one vs one” or the “one vs all” strategies. However, SVM cannot be directly employed for speech recognition, since input utterances are composed of a varying number of feature vectors. The approaches followed in the literature to solve the problem are either based on hybrid SVM/HMM architectures [10] or on dynamic kernels [9]. Another problem with SVM is the computational demand required in the training phase. This is particularly important in speech recognition tasks since the size of training corpora can be very large [3].

Extreme Learning Machine (ELM) is a discriminative technique that recently gained much interest in the scientific community for its capability to increase training speed over traditional neural networks learning methods [11,12]. Additionally, in recent studies [13] ELM has been compared to SVM and it provided similar accuracies with fast training and testing speed. Regarding the applications, the ELM paradigm has been proposed for system identification in nonstationary environments [14,15], in particular for learning Time-Varying Neural Networks (TV-NN) thus taking full advantage of the speedy training procedure with certain matrix transformations. In [16], the Circular-ELM algorithm is introduced to address the visual quality assessment problem. In [41], an ensemble of ELM and random

* Corresponding author.
E-mail address: e.principi@univpm.it (E. Principi).

http://dx.doi.org/10.1016/j.neucom.2014.01.067
0925-2312/© 2014 Elsevier B.V. All rights reserved.

projections (RP-ELM) is proposed to sharply reduce the number of neurons in the hidden node without affecting the generalization performance in prediction accuracy. As a result, the eventual learning machine always benefits from a considerable simplification in the feature-mapping stage. This allows the RP-ELM model to properly balance classification accuracy and resource occupation. In [42], it is investigated how the high generalization performance, low computational complexity, and fast learning speed of ELM can be exploited to perform analogical reasoning in a vector space model of affective common-sense knowledge. In particular, by enabling a fast reconfiguration of such a vector space, ELM allows the polarity associated with natural language concepts to be calculated in a more dynamic and accurate way and, hence, perform better concept-level sentiment analysis. Several works exist also that applied ELM to classification tasks [17]. In [26], Huang et al. showed how ELM can be applied in multiclass classification directly and that it achieve better generalization performance at faster training speed then Support Vector Machine and least square Support Vector Machine. Savitha and colleagues [18] proposed the "Circular Complex-valued Extreme Learning Machine (CC-ELM)" algorithm for classification. CC-ELM consists in a single hidden layer network with non-linear input and hidden layers and a linear output layer and it has been applied to the acoustic emission signal and mammogram classification problems. In [19], the ELM paradigm has been applied to the classification of music genres. As features, they employed zero crossing rates, energy, root-mean-square, crest factor, spectral centroid, Mel-Frequency Cepstral Coefficients (MFCC) and specific loudness sensation.

In this paper, the ELM paradigm has been applied for the automatic classification of speech utterances, with particular attention to the recognition of distress calls for emergency state detection. The motivation for applying ELM to this task is to achieve better performance respect to DTW-based template-matching approaches with a computational burden lower than SVM. Up to the authors' knowledge, it is the first time that ELM is applied for recognizing speech. The system employs Power Normalized Cepstral Coefficients (PNCC) [20] as low-level features, then normalizes the length of the input utterances so that regardless the number of feature vectors they are mapped to vectors of fixed length. PNCC feature vectors are employed as an alternative to MFCCs in order to improve the robustness of the system against acoustic distortions. For normalizing input utterances, two strategies have been assessed: the first is based on DTW, and consists in calculating the distances between the input utterance and the candidate templates. The second is based on the vectorized outerproduct of trajectory matrix [21]. Utterance length normalization is necessary for ELM since the number of input neurons in the network is fixed. In addition to the baseline ELM algorithm, the ELM with kernel approach has also been tested. The algorithms have been compared to DTW-based template-matching speech recognition and to SVM. Regarding the latter, classification is performed on DTW distances and on the outerproduct of trajectory matrix as in ELM, so that the performance depends only on he classifier. The experiments have been conducted on two corpora: TIDIGITS [22] and ITAAL [23]. TIDIGITS has been employed to evaluated the performance on a well-known corpus and in clean acoustic conditions. ITAAL is a new Italian speech corpus of home automation commands and distress calls recorded with distant and close-talking microphones in normal and shouted speaking styles. Using ITAAL, it is possible to validate the system performance on an emergency detection task and in realistic acoustic conditions, since signals are affected by noise and reverberation. The experimental task consists in the recognition of the correct distress call and in the discrimination between in-domain utterances and out-of-domain ones. The experiments demonstrated that in several tasks the performance of ELM with kernel is comparable or superior than SVM with lower training and testing times.

The outline of the paper is the following: Section 2 presents a brief overview of ELM. Section 3 describes ELM-based speech recognition system, the MFCC and PNCC feature extraction pipelines and the utterance length normalization approaches. DTW and SVM based speech recognition algorithms are described in Section 4, and Section 5 presents the experiments conducted to asses the performance of the proposed algorithm. Finally, Section 6 concludes the paper and presents future developments.

2. Overview of Extreme Learning Machine

ELM is a fast learning algorithm designed for single hidden layer feedforward neural networks (SLFNs). In later works [24,25], ELM has been extended to SLFNs where the hidden layer need not to be neuron alike. In ELM, the input weights of SLFNs do not need to be tuned and they can be randomly generated, whereas the output weights are analytically determined using the least-square method, thus allowing a significant training time reduction.

Consider a set of N labeled training samples $\{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)\}$, where $\mathbf{x}_i \in \mathbb{R}^I$ and $t_i \in \{-1, 1\}$, and a SLFN with I input neurons and L hidden neurons (Fig. 1). For binary classification, the ELM decision function is the following:

$$f_L(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{L} \beta_i h_i(\mathbf{x})\right) = \text{sign}(\mathbf{h}(\mathbf{x})\boldsymbol{\beta}). \tag{1}$$

In the equation, the vector $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_L]^T$ contains the weights connecting hidden neurons and output neurons, while $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \ldots, h_L(\mathbf{x})]$ is the output of the hidden layer with respect to the input $\mathbf{x}$. In general, $\mathbf{h}(\mathbf{x}) = [G(\mathbf{a}_1, b_1, \mathbf{x}), \ldots, G(\mathbf{a}_L, b_L, \mathbf{x})]$ and $G(\mathbf{a}, b, \mathbf{x})$ is a nonlinear piecewise continuous function that satisfies ELM universal approximation capability theorems, and $\{\mathbf{a}_i, b_i\}_{i=1}^L$ are randomly generated. In this paper, $G(\mathbf{a}, b, \mathbf{x})$ assumes the form of the sigmoid function since it provided the best performance in the experiments.

Defining the hidden-layer output matrix $\mathbf{H}$ as

$$\mathbf{H} = \begin{bmatrix} h_1(\mathbf{x}_1) & \cdots & h_L(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ h_1(\mathbf{x}_N) & \cdots & h_L(\mathbf{x}_N) \end{bmatrix}, \tag{2}$$

training the ELM consists in minimizing $\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|$ and $\|\boldsymbol{\beta}\|$, where $\mathbf{T} = [t_1, t_2, \ldots, t_N]^T$. The solution to the problem can be calculated as the minimum norm least-square solution of the linear system:

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T}, \tag{3}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of matrix $\mathbf{H}$. By computing output weights analytically, ELM allows achieving good generalization performance with speedy training phase.

ELM can be also applied to multiclass classification problems [26]. Without entering into the details, in this paper the multi-output nodes technique has been employed. This consists in considering as the label of the input data the index of the output node with the highest output value.

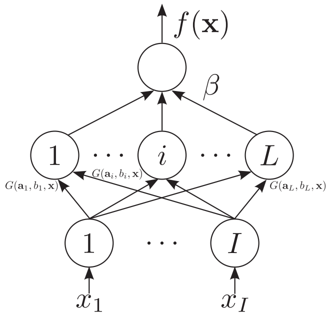

$f(\mathbf{x})$
β
1
i
L
$G(\mathbf{a}_1, b_1, \mathbf{x})$
$G(\mathbf{a}_i, b_i, \mathbf{x})$
$G(\mathbf{a}_L, b_L, \mathbf{x})$
1
I
x_1
x_I

Fig. 1. ELM with I input neurons and L hidden neurons.

2.1. Extreme Learning Machine with kernels

In kernel-based ELM [26], $\mathbf{h}(\mathbf{x})$ is unknown, and the output function of the classifier is written as

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} K(\mathbf{x},\mathbf{x}_1) \\ \vdots \\ K(\mathbf{x},\mathbf{x}_N) \end{bmatrix}^T \left(\frac{\mathbf{I}}{C} + \boldsymbol{\Omega}\right)^{-1} \mathbf{T}, \tag{4}$$

where $\boldsymbol{\Omega}$ is defined so that each element $\Omega_{i,j} = h(\mathbf{x}_i) \cdot h(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$. $K(\cdot,\cdot)$ is a kernel function as in SVM, and in this work assumes the form a radial basis function. It is interesting to note that differently from standard ELM, the number of hidden neurons must not be known in advance.

3. Application of Extreme Learning Machine to speech recognition

3.1. Problem formulation

Consider a training corpus $\mathcal{T} = \{(\mathbf{U}_1, C_1), \dots, (\mathbf{U}_K, C_K)\}$ where $\mathbf{U}_k = \{\mathbf{u}_{k,1}, \dots, \mathbf{u}_{k,L_k}\}$, $\mathbf{u}_{k,l}$ is the $D \times 1$ low-level feature vector of utterance k at the time frame index l and C_k is the corresponding label. Given a test utterance $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{L_y}\}$, the problem is finding the corresponding label $C_y \in \{C_1, C_2, \dots, C_K\}$ based on a certain classification criterion. In this work, we suppose that the training corpus $\mathcal{T} = \mathcal{I} \bigcup \mathcal{O}$, i.e., that it is composed of two subsets: $\mathcal{I} = \{(\mathbf{U}_1, C_1), \dots, (\mathbf{U}_{K_I}, C_{K_I})\}$ is the set of in-domain utterances that represent sentences containing semantically meaningful content. The set $\mathcal{O} = \{(\mathbf{U}_{K_I+1}, C_{K_I+1}), \dots, (\mathbf{U}_K, C_K)\}$ is the set of out-of-domain utterances that model sentences that should be discarded by the system. Note that $C_i = C_j = C_G \quad \forall i, j \in \{K_I+1, \dots, K\}$, i.e., out-of-domain utterances are all associated to the same class label.

In order to apply ELM for determining the class label C_y, the input utterances must be mapped to vectors of fixed-length. This because the number of input neurons in the ELM is fixed, thus the length of the input vector must be known a priori. In this work, two methods for normalizing the length of the input utterances have been employed: in the first, ELM operates on the DTW distances between the test utterance and each template utterance. In the second, ELM operates on the vectorized outerproduct of trajectory matrix [21]. The block-scheme of the proposed approach is shown in Fig. 2.

Before describing the proposed approaches more in details, a brief overview of two feature extraction algorithms for obtaining low-level descriptors will be provided.

3.2. Feature extraction

Several features have been proposed in the literature for the extraction of meaningful characteristics from speech signals. In this paper, two types of features have been addressed: the first are MFCCs [27], a popular choice in most automatic speech recognition system. The second are PNCCs [20], similar to MFCCs but they are more robust to noise and reverberation distortions. This section briefly describes the two feature extraction pipelines.

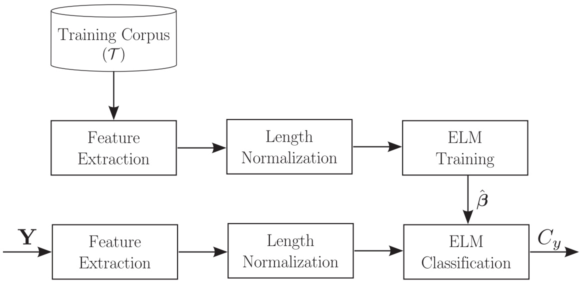


Fig. 2. Block scheme of the ELM-based speech recognition approach.

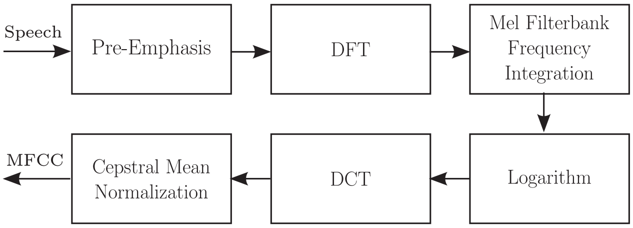

Speech
Pre-Emphasis
DFT
Mel Filterbank Frequency Integration
Logarithm
DCT
Cepstral Mean Normalization
MFCC

Fig. 3. The MFCC feature extraction pipeline.

3.2.1. Mel-Frequency cepstral coefficients

The block-scheme of the MFCC feature extraction pipeline is shown in Fig. 3. The first processing step is the *pre-emphasis* of the input speech signal. Pre-emphasis consists in filtering the signal with a filter whose transfer function is

$$H(z) = 1 - \alpha z^{-1}, \tag{5}$$

where usually $0.9 < \alpha \le 1.0$. The objective of pre-emphasis is to remove the DC components and to raise the high-frequency part of the spectrum, which has a 6 dB/decade decay for human speech on average.

The signal is then segmented into partially overlapped frames of length 10–30 ms. A common choice for the window function is the Hamming window, whose form is the following:

$$w(n) = \begin{cases} 0.54 - 0.46\ \cos(2\pi n/L), & 0 \le n \le L, \\ 0 & \text{otherwise}, \end{cases} \tag{6}$$

where L is the frame length in samples, and n denotes the time index.

For each frame, the Discrete Fourier Transform (DFT) is calculated and filtered with a filterbank composed of a set of triangular filters uniformly spaced on the mel scale. This is a non-linear transformation that maps a frequency f to the corresponding mel-scaled frequency g using the following expression:

$$g = 1127\ \log\left(1 + \frac{f}{700}\right). \tag{7}$$

Denoting with $S(i)$ the DFT of a speech frame and i the frequency bin, the output of the "Mel Filterbank & Frequency Integration" block is

$$H(k) = \sum_{i = ini(k)}^{end(k)} |S(i)|^2 W_k(i), \quad k = 1, 2, \dots, N \tag{8}$$

where $H(k)$ and $W_k(i)$ are the output and the frequency response of the kth filter respectively, $ini(k)$ and $end(k)$ are starting and ending frequency indices of that filter and N is the number of filters in the bank.

The final steps for the calculation of the jth MFCC $c(j)$ is the logarithm of the filterbanks outputs and their Discrete Cosine Transform (DCT):

$$c(j) = \sum_{k=1}^{N} \log\left[H(k)\right] \cos\left[\frac{\pi j}{N}(k - 0.5)\right], \quad j = 0, 1, \dots, M-1 \le N \tag{9}$$

Cepstral Mean Normalization (CMN) is usually applied to MFCCs to increase the robustness against channel distortions. CMN consists in subtracting the mean of each cepstral coefficients calculated over

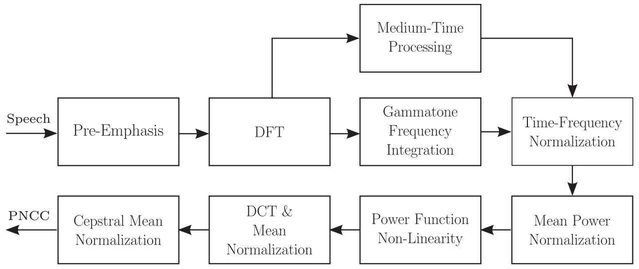

Speech
Pre-Emphasis
DFT
Medium-Time Processing
Gammatone Frequency Integration
Time-Frequency Normalization
Mean Power Normalization
Power Function Non-Linearity
DCT & Mean Normalization
Cepstral Mean Normalization
PNCC

Fig. 4. The PNCC feature extraction pipeline.

the entire utterance:

$$c'_t(n) = c_t(n) - \frac{1}{T}\sum_{l=0}^{T-1} c_l(n), \quad t = 0, 1, \ldots, T-1 \tag{10}$$

where t denotes the time frame index and T is the utterance length in frames.

3.2.2. Power normalized cepstral coefficients

In the literature, several approaches have been proposed to improve the robustness of speech recognition systems against noise and reverberation. Speech enhancement techniques, such as spectral subtraction [28], Ephraim & Malah log-spectral amplitude estimator [29], or dereverberation frameworks such as [30] operate before the feature extraction pipeline. Other approaches, such as Vector Taylor Series speech enhancement [31] or single [32] and multi-channel MFCC-MMSE [33] modify the extraction algorithm. An alternative approach consists in using a different set of features that are intrinsically more robust than MFCCs. Recently, PNCCs [20] have demonstrated their effectiveness at the cost of a modest increment of computational burden. Fig. 4 illustrates the main steps needed for the extraction of PNCCs: the main innovations with respect to MFCCs reside in the replacement of the logarithmic non-linearity with a power function law and the introduction of the "Medium-Time Processing".

The first stages of the extraction pipeline are the same of the MFCC extraction one. The first difference in PNCCs calculation is the replacement of the mel-spaced filterbank with a gammatone one [34]. The motivation behind this choice is that the latter slightly improves the recognition accuracy. The subsequent steps mark the real difference between PNCCs and MFCCs. The "Medium-Time Processing" stage exploits a longer-duration temporal analysis (e.g., 5 frames) to estimate the noise floor level and to subtract it from the instantaneous power of the input signal. Instead of directly using the filtered signal, the output of the "Medium-Time Processing" stage is a transfer function that modulates the original signal in the "Time-Frequency Normalization" step. In the "Mean Power normalization" stage, the signal power is normalized dividing the input by a running average of the overall power. In MFCCs, the logarithm non-linearity is applied to the output of the mel filter-bank. Here, instead, a power function non-linearity with exponent 1/15 is applied. The motivation arises from studies on the non-linear curve that relates the sound pressure level in dB to the auditory-nerve firing rate. Experiments demonstrated that replacing the logarithmic non-linearity with the power-function one improves the recognition accuracy [20]. The final stages of the PNCC pipeline are the computation of the DCT and the mean normalization as in the MFCC pipeline.

3.3. Classification based on DTW distances

Classification based on DTW distances operates transforming the sequence of feature vectors of an input utterance into a vector of fixed length by calculating the distance between the input utterance and each utterance of set $\mathcal{I}$. Referring to the notation introduced in Section 3.1, this means that an input utterance $\mathbf{U}$ is mapped to a fixed length vector of size $I \times 1$ as follows:

$$\mathbf{v} = [d(\mathbf{X}, \mathbf{U}_1), d(\mathbf{X}, \mathbf{U}_2), \ldots, d(\mathbf{X}, \mathbf{U}_I)]^T, \tag{11}$$

where $d(\cdot, \cdot)$ represents the DTW distance. The DTW algorithm will be briefly described in the next sections.

The training set of the ELM is created normalizing the lengths the lengths of the entire training corpus $\mathcal{T}$. This means that each vector $\mathbf{v}_i$ is obtained as follows:

$$\mathbf{v}_i = [d(\mathbf{U}_i, \mathbf{U}_1), \ldots, d(\mathbf{U}_i, \mathbf{U}_I)]^T. \tag{12}$$

3.4. Classification based on the outerproduct of trajectory matrix

Differently from the previous section, where ELM input were the DTW distances between an input utterance and template patterns, in this section classification is performed transforming directly input feature vectors. The technique adopted in this work is the outerproduct of trajectory matrix, proposed in [21] for SVM-based speech recognizers.

Given an input utterance composed of L feature vectors of dimension D, the trajectory matrix is an $L \times D$ matrix defined as

$$\mathbf{U} = [\mathbf{u}_1^T, \mathbf{u}_2^T, \ldots, \mathbf{u}_L^T]^T. \tag{13}$$

The outerproduct trajectory matrix is then defined as

$$\mathbf{Z} = \mathbf{U}^T \mathbf{U}. \tag{14}$$

Regardless the number of frames L in the input utterance, $\mathbf{Z}$ is a $D \times D$ matrix. Note, also, that $\mathbf{Z}$ is symmetric, thus it contains $D(D+1)/2$ unique elements. The final feature vector $\mathbf{z}$ is a $D(D+1)/2 \times 1$ vector obtained vectorizing the outerproduct trajectory matrix and choosing the unique elements.

4. Alternative approaches

The proposed approaches based on ELM have been compared to template matching based on DTW and to the SVM. The two techniques will be now briefly reminded.

4.1. DTW-based speech recognition

The DTW algorithm makes possible to measure the dissimilarity between two sequences of different length. The algorithm is based on dynamic programming principles and it will be here briefly reminded: for an exhaustive discussion, refer to [2].

Denoting with $\mathbf{X}$ and $\mathbf{Y}$ two speech utterance represented by the sequences of feature vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{T_x})\}$ and $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{T_y}\}$ respectively, the objective of DTW is to calculate a measure of their dissimilarity. This can be defined in terms of the short-time spectral distortion $d(\mathbf{x}_{i_x}, \mathbf{y}_{i_y})$, that in this paper is represented by the Euclidean distance between the two feature vectors. Denoting $d(\mathbf{x}_{i_x}, \mathbf{y}_{i_y})$ with $d(i_x, i_y)$ for simplicity of notation, where $i_x = 1, 2, \ldots, T_x$ and $i_y = 1, 2, \ldots, T_y$, the global dissimilarity between the two patterns can be defined as

$$d(\mathbf{X}, \mathbf{Y}) = \min_{\phi} d_{\phi}(\mathbf{X}, \mathbf{Y}) = \min_{\phi} \left\{ \sum_{k=1}^{T} d(\phi_x(k), \phi_y(k)) m(k) / M_{\phi} \right\}. \tag{15}$$

In the equation, $\phi = (\phi_x, \phi_y)$ is the pair of warping function that maps the pattern $\mathbf{X}$ to the pattern $\mathbf{Y}$, $m(k) \leq 0$ is a weighting coefficient and M_{ϕ} is a normalizing factor.

The complete DTW algorithm is derived from the above definitions and imposing a set of time-normalization constraints. Without entering into the details, the constraints allow the dissimilarity $d(\mathbf{X},\mathbf{Y})$ to be redefined as

$$d(\mathbf{X},\mathbf{Y}) \triangleq \frac{D(T_x,T_y)}{M_\phi}. \tag{16}$$

The algorithm for calculating the global distance in the grid beginning at $(1,1)$ and ending at (T_x,T_y) is the following:

(1) Initialization:

$$D(1,1) = d(1,1)m(1). \tag{17}$$

(2) Recursion: For $1 \le i_x \le T_x$, $1 \le i_y \le T_y$

$$D(i_x,i_y) = \min\left\{\begin{array}{l} D(i_x-1,i_y)+d(i_x,i_y), \\ D(i_x-1,i_y-1)+2d(i_x,i_y), \\ D(i_x,i_y-1)+d(i_x,i_y) \end{array}\right\}. \tag{18}$$

(3) Termination:

$$d(\mathbf{X},\mathbf{Y}) = \frac{D(T_x,T_y)}{M_\phi}. \tag{19}$$

Note that in Eq. (18), Sakoe and Chiba [2] local constraints and slope weights have been applied. The DTW algorithm here described has been implemented in Matlab code and employed in all the experiments described in later sections.

In DTW-based speech recognition, the class label of an input utterance is determined selecting the sentence whose template has the smallest distance with the input utterance pattern. The problem with this solution is that, whatever the user spoke, the system will always produce a recognition result. In other words, the system is not able to discriminate between in-domain utterances and out-of-domain ones. The problem can be overcome accepting the outcome only if the output distance is below a predefined threshold. Such a threshold is estimated from the distributions of the distances of the true positive examples (i.e., the set $\mathcal{I}$) and of the true negative ones (i.e., the set $\mathcal{O}$). Fig. 5 shows an example of the two distributions, which for simplicity are assumed gaussians: the threshold value can be chosen determining the intersection point between the two distributions. Since the two distributions overlap, a certain percentage of false positives (i.e., incorrectly accepted out-of-domain utterances) and of false negatives (i.e., incorrectly rejected in-domain utterances) have to be taken into account.

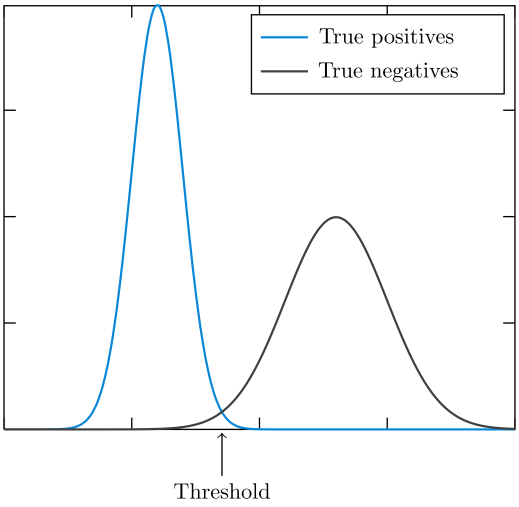


Fig. 5. Distributions of true positives and true negatives distances.

4.2. SVM-based speech recognition

A survey on SVM applied to speech recognition tasks has been presented in [35]. SVMs are binary classifiers that decide whether an input vector $\mathbf{x}$ belongs to class $+1$ or to class -1 based on the following discriminant function:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i t_i K(\mathbf{x},\mathbf{x}_i) + d, \tag{20}$$

where $t_i \in \{+1,-1\}$, $\alpha_i > 0$ and $\sum_{i=1}^{N} \alpha_i t_i = 0$. The terms $\mathbf{x}_i$ are the "support vectors" and d is a bias term that together with the α_i are determined during the training process of the SVM. The kernel function $K(\cdot,\cdot)$ can assume different forms [36]. In this work, the radial basis function (RBF) kernel $K(\mathbf{x},\mathbf{x}_i) = \exp(-\gamma\|\mathbf{x}-\mathbf{x}_i\|^2)$ has been employed. The input vector $\mathbf{x}$ is classified as $+1$ if $f(\mathbf{x}) \ge 0$ and -1 if $f(\mathbf{x}) < 0$.

As with ELM, SVM classifies input utterances based on the DTW distances or on the vectorized outerproduct of trajectory matrix. For SVM, several alternatives have been proposed to deal with variable length input sequences [9]. However, in this work we adopt the same strategies employed for ELM so that the performance depends only on the classifiers.

Since SVMs are binary classifiers, a strategy must be adopted also to deal with multiclass problems. The most popular techniques are "one versus one" and "one versus all": in this work, the "one versus all" technique has been employed. LIBSVM [37] has been employed both in the training and testing phases of the SVM. For selecting the values of the C and γ parameters, a grid search has been performed as suggested in [37].

5. Experiments

Two corpora have been employed to evaluate the proposed approaches: TIDIGITS [22] and ITAAL [23]. In TIDIGITS, the algorithms have been evaluated in an isolated digit recognition task. ITAAL is a recently developed speech corpus of distress calls and home automation commands in Italian, and it has been employed to assess the performance in a more realistic scenario. In both experiments, the speech signals have been downsampled to 16 kHz.

Due to class unbalance, the performance has been assessed using the average F_1-Measure defined as

$$F_1 = \frac{1}{N}\sum_{i=1}^{N} F_1(i), \tag{21}$$

where N is the total number of classes and $F_1(i)$ is the F_1-Measure of class i. The F_1-Measure of class i is calculated from its precision $P(i)$ and recall $R(i)$:

$$P(i) = \frac{A(i,i)}{\sum_i A(i,j)}, \tag{22}$$

$$R(i) = \frac{A(i,i)}{\sum_j A(i,j)}, \tag{23}$$

$$F_1(i) = 2\frac{P(i)R(i)}{P(i)+R(i)}. \tag{24}$$

where $A(i,j)$ represents the number of times that the true class label i has been classified as class j (i.e., $A(i,j)$ is the element of row i and column j of the class confusion matrix).

Regarding the parameters of feature extraction pipelines, the MFCC one has been configured as follows:

- pre-emphasis coefficient (μ): 0.97;
- frame-size/frame-shift: 25 ms/10 ms;
- number of filters in the mel-filterbank: 23.

The PNCC pipeline has been configured similar to the MFCC one, with the only difference being the number of filters of the gammatone filterbank, which has been set to 40. The remaining parameters have been set as in [20].

The experiments report the best performance achieved varying the algorithm parameters. Regarding ELM, the number of neurons has been varied from 10 to 1000 with an increment of 10 neurons at each iteration. Optimal values of the ELM kernel and SVM parameters C and γ have been selected using a grid search as suggested in [37]. In particular, C has been varied from 2^{-5} to 2^{15} and γ from 2^{-15} to 2^{3} incrementing the exponent by 2 at each iteration.

As aforementioned, the LIBSVM [37] implementation of SVM has been used in the experiments. Regarding ELM and ELM kernel, they have been implemented in ANSI C language based on the reference Matlab code available here [38].

5.1. Experiments on the TIDIGITS corpus

The data set of this experiment combines the original training and testing sets of the TIDIGITS corpus. This dataset has been divided into two sets, one containing single-digits utterances having a total duration of 79.96 s and the other containing sequences of digits utterances having a total duration of 424.26 s. Single-digits utterances represent in-domain sentences (i.e., the set $\mathcal{I}$), while multiple-digits ones represent out-of-domain sentences (i.e., the set $\mathcal{O}$) that have to be discarded. The total number of speakers is 225, with 111 males and 114 females. The number of utterances per speaker is 77, with 22 single-digit utterances, and 55 multiple-digits utterances. The experiment has been conducted using 9-fold cross-validation, with each fold composed of 25 speakers. The parameters values employed for obtaining the reported results are shown in Table 1.

Table 2 shows the results obtained with MFCC and PNCC coefficients (highest F_1-Measures are shown in bold). Observing the values, it is evident that the performance obtained using the two features is very close. This result could be expected, since the TIDIGITS corpus has been recorded in quiet conditions, so both training and testing signals are noise and reverberation free. PNCC coefficients are particularly effective when testing signals are distorted with additive noise or reverberation, so it is reasonable that their performance is very close to the MFCC one.

Comparing the results of ELM, ELM kernel and SVM, it can be noticed that both when features and distances are employed for classification, ELM kernel and SVM perform similarly and they outperform the ELM algorithm. In distance-based classification, SVM slightly outperforms ELM kernel, while in features-based classification the opposite occurs. Comparing features- and distance-based classification, ELM and SVM perform better with the former, while ELM kernel gives similar results. All the approaches outperform DTW both with MFCCs and PNCCs, and with features- and distance based classification.

5.1.1. Evaluation of execution times

The advantage of ELM respect to SVM is mostly speed than accuracy, so it is worth analyzing the performance of the algorithms also in terms of training and testing execution times. The performance here is evaluated in terms of “Real-Time Factor” (RTF), defined as the ratio between execution time and the duration of the data set. Measures have been conducted on a PC equipped with an Intel Core i7-3520 M CPU running at 2.90 GHz and with 8 GB of RAM. Note that differently from [13], here the training execution time does not include the time spent for parameter tuning.

The results are shown in Table 3. It is evident from the values that ELM and ELM kernel are the most performing algorithms both in the distance-based classification and in features-based one.

Table 1
Parameter values employed in TIDIGITS experiments.

Algorithm	MFCC	PNCC
Distance*−*based		
ELM (neurons)	670	880
ELM kernel (C,γ)	$2, 2^{-3}$	$2, 2^{-3}$
SVM (C,γ)	$2^{7}, 2^{-7}$	$2^{7}, 2^{-7}$
Features*−*based		
ELM (neurons)	690	830
ELM kernel (C,γ)	$2^{3}, 2$	$2, 2$
SVM (C,γ)	$2^{3}, 2^{-3}$	$2^{3}, 2^{-3}$

Table 2
F_1-Measure (%) on the development and test sets of the TIDIGITS corpus with MFCC and PNCC coefficients. In each column, highest values are shown in bold.

Algorithm	MFCC	PNCC
Distance*−*based		
ELM	85.87	85.90
ELM kernel	91.37	91.52
SVM	**92.47**	**92.05**
Features*−*based		
ELM	80.34	80.50
ELM kernel	91.41	91.63
SVM	89.89	90.01
DTW	74.24	74.29

Table 3
Real-time factors for the training and testing sets of the TIDIGITS corpus.

Algorithm	Training RTF (%)	Testing RTF (%)
Distance*−*based		
ELM	0.011	0.001
ELM kernel	0.014	0.003
SVM	0.017	0.010
Features*−*based		
ELM	0.012	0.002
ELM kernel	0.019	0.011
SVM	0.027	0.035

5.2. Experiments on the ITAAL corpus

ITAAL[1] is an Italian corpus of home automation commands and distress calls spoken by 20 native Italian speakers (10 males, 10 females) [23]. Each utterance has been acquired with a close-talking microphone and with an array composed of four microphones. In this set of experiments, the recognition performance is evaluated on the close-talking microphone signal and on the central microphone signal of the array. The reverberation time of the acquisition room was 0.72 s. The average signal-to-noise ratio of the close-talking microphone signals is 51.46 dB, and the one of the distant microphone signals is 34.08 dB. Each person spoke the corpus sentences standing in front of the microphone array at a distance of 3 m and she/he was asked to read three times 15 home automation commands, 5 distress calls, both in normal and shouted conditions. The vocabulary is composed of 24 words. Additional details of ITAAL are provided in [23].

[1] Audio samples are available at url: http://www.a3lab.dii.univpm.it/projects/itaal

In this task, distress calls represent the set $\mathcal{I}$, i.e., they are considered as in-domain sentences that should be accepted and classified correctly. Home automation commands represent the set $\mathcal{O}$, i.e., they are the out-of-domain utterances that should be discarded. The data for each speaker, microphone and vocal effort consists in 15 in-domain utterances (5 distress calls each repeated three times) and 15 out-of-domain utterances (1 repetition of the 15 home-automation commands). As with the TIDIGITS corpus, the experiments have been performed with cross-validation (20 folds, i.e., leave-one-out). The parameters values employed for obtaining the reported results are shown in Tables 4 and 6.

Table 5 shows the results obtained on headset microphone signals, while Table 7 the ones obtained on the distant microphone signals. Highest values are shown in bold. The advantage of using PNCCs instead of MFCCs is particularly evident in DTW, and in distance-based classification, where the algorithms reach an average performance improvement of about 5% with respect to MFCCs. On the other hand, in features-based classification the advantage of using PNCCs is not so evident and the algorithm taking most of the advantage from them is ELM kernel, where the F_1-Measure increases of 1.5%.

Distance-based classification is the most performing approach, in particular when coupled with PNCC coefficients. On average, ELM, both with MFCCs and with PNCCs, benefits the most from using distance-based classification, with an average performance improvement of 7.86% with MFCCs and 12.88% with PNCCs. On the contrary, ELM kernel performance is similar when MFCCs are employed, while with PNCCs the performance improvement amounts to 4.54%. The same can be observed with SVM, where the performance improvement with PNCCs is 4.48%.

Comparing ELM-based approaches with SVM, on average the most performing algorithm is ELM kernel which gives a performance improvement of more than 1% both with MFCCs and PNCCs in distance classification. Overall, the most performing approach is ELM kernel with distance-based classification.

Table 4
Parameter values employed in ITAAL experiment with headset signals. Highest values are shown in bold.

	Headset, Normal		**Headset, Shout**	
Algorithm	**MFCC**	**PNCC**	**MFCC**	**PNCC**
Distance−based				
ELM (neurons)	70	70	120	210
ELM kernel (C, γ)	2^{-3}, 2^{-3}	2^{-3}, 2^{-3}	2^{7}, 2	2^{-1}, 2^{-3}
SVM (C, γ)	2^{7}, 2^{-3}	2^{15}, 2^{-13}	2^{5}, 2^{-3}	2^{5}, 2^{-3}
Features−based				
ELM (neurons)	140	150	170	110
ELM kernel (C, γ)	2^{11}, 2^{3}	2^{-1}, 2^{-3}	2^{11}, 2^{3}	2^{7}, 2^{3}
SVM (C, γ)	2^{3}, 2^{-3}	2^{7}, 2^{-3}	2^{7}, 2^{-3}	2^{7}, 2^{-5}

Table 5
F_1-Measure (%) on the ITAAL headset microphone development sets with MFCC and PNCC coefficients. In each column, highest values are shown in bold.

	Headset, Normal		**Headset, Shout**	
Algorithm	**MFCC**	**PNCC**	**MFCC**	**PNCC**
Distance−based				
ELM	87.93	92.89	85.32	90.34
ELM kernel	90.40	**95.73**	**89.47**	**94.04**
SVM	**91.36**	95.65	86.96	93.57
Features−based				
ELM	79.63	79.87	73.82	72.86
ELM kernel	90.59	92.58	84.21	87.46
SVM	89.48	88.38	88.07	89.22
DTW	83.83	92.72	86.56	92.56

Table 6
Parameter values employed in ITAAL experiment with distant microphone signals. In each column, highest values are shown in bold.

	Distant, Normal		**Distant, Shout**	
Algorithm	**MFCC**	**PNCC**	**MFCC**	**PNCC**
Distance−based				
ELM (neurons)	170	230	180	180
ELM kernel (C, γ)	2, 2^{-3}	2^{-1}, 2^{-3}	2^{-1}, 2^{-3}	2^{7}, 2
SVM (C, γ)	2^{7}, 2^{-7}	2^{5}, 2^{-5}	2^{13}, 2^{-7}	2^{5}, 2^{-1}
Features−based				
ELM (neurons)	130	100	110	110
ELM kernel (C, γ)	2^{-1}, 2^{-3}	2^{3}, 2^{3}	2^{7}, 2^{3}	2^{3}, 2^{3}
SVM (C, γ)	2^{7}, 2^{-3}	2^{3}, 2^{-3}	2^{7}, 2^{-5}	2^{3}, 2^{-5}

Table 7
F_1-Measure (%) on the ITAAL distant microphone development sets with MFCC and PNCC coefficients.

	Distant, Normal		**Distant, Shout**	
Algorithm	**MFCC**	**PNCC**	**MFCC**	**PNCC**
Distance−based				
ELM	81.19	85.12	81.05	87.84
ELM kernel	86.18	**89.47**	**83.72**	**91.51**
SVM	84.43	87.89	82.41	89.14
Features−based				
ELM	77.63	79.40	72.98	72.54
ELM kernel	**88.46**	88.48	83.33	84.07
SVM	86.71	85.23	83.48	85.49
DTW	79.50	87.61	79.59	87.57

Table 8
Real-time factors for the training and testing sets of the ITAAL corpus.

Algorithm	**Training RTF (%)**	**Testing RTF (%)**
Distance−based		
ELM	0.029	0.009
ELM kernel	0.054	0.025
SVM	0.088	0.041
Features−based		
ELM	0.008	0.003
ELM kernel	0.015	0.006
SVM	0.041	0.055

5.2.1. Evaluation of execution times

As in the experiments with TIDIGITS corpus, the performance of ELM and SVM algorithms has been evaluated in terms of RTF. Table 8 shows the obtained results. Observing the values, the same conclusions of TIDIGITS experiments can be drawn: ELM is the faster algorithm both in the training and testing phases, and both with in the distance-based and in the features-based approach. ELM kernel is about twice slower than ELM, while SVM remains the slowest algorithm.

6. Conclusions

This paper presented an ELM approach for the automatic classification of spoken utterances, with particular attention to distress calls recognition for emergency state detection. Both baseline ELM and ELM with kernels approaches have been assessed, and two techniques for normalizing the length of input utterances have been employed: the first is based on DTW-distances and the second of the outerproduct of trajectory matrix. As low-level features, two alternatives have been implemented and tested: MFCCs, i.e., the popular choice in most of today's automatic speech recognition systems, and PNCCs, a recently proposed feature set that increases the robustness of the system against acoustic distortions. The ELM-based approaches have been compared with DTW-based speech recognition and with SVM on the TIDIGITS and ITAAL corpora. The results demonstrated that ELM kernel coupled with PNCCs and classification based on DTW distances achieves superior or comparable performance respect to SVM under all the addressed acoustic conditions, with reduced training and testing times.

In future works, experiments will be carried out to evaluate the system in mismatched conditions, e.g., creating templates using headset signals and testing using distant microphone signals. The influence of pre-processing techniques for reducing such mismatch will be also investigated. In addition, performance will be evaluated varying the number of available data for training the ELM, and alternative ELM algorithms will also be considered [18]. Finally, the use of common-sense computing [39] will be explored in order to perform the automatic classification of spoken utterances at content-, concept-, and context-level [43].

References

[1] G. Saon, J.-T. Chien, Large-vocabulary continuous speech recognition systems: a look at some recent advances, IEEE Signal Process. Mag. 29 (6) (2012) 18–33.
[2] L.R. Rabiner, B.H. Juang, Fundamentals of Speech Recognition, Prentice Hall, Englewood Cliffs, NJ, 1993.
[3] R. Solera-Ure na, D. Martín-Iglesias, A. Gallardo-Antolín, C. Peláez-Moreno, F.D. de Mara, Robust ASR using support vector machines, Speech Commun. 49 (4) (2007) 253–267.
[4] J. Keshet, S. Bengio, Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods, John Wiley & Sons, West Sussex, UK, January 2009.
[5] X. Zhang, J. Sun, Z. Luo, M. Li, Confidence index dynamic time warping for language-independent embedded speech recognition, in: Proceedings of ICASSP, Vancouver, Canada, May 26–31, 2013, pp. 8066–8070.
[6] C. Kim, K.D. Seo, Robust DTW-based recognition algorithm for hand-held consumer devices, IEEE Trans. Consum. Electron. 51 (2) (2005) 699–709.
[7] X. Anguera, Information retrieval-based dynamic time warping, in: Proceedings of Interspeech, Lyon, France, August 25–29 2013, pp. 1–5.
[8] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[9] A.D. Dileep, C.C. Sekhar, Class-specific GMM based intermediate matching kernel for classification of varying length patterns of long duration speech using support vector machines, Speech Commun. 57 (2014) 126–143.
[10] A. Ganapathiraju, J. Hamaker, J. Picone, Hybrid SVM/HMM Architectures for Speech Recognition, in: Proceedings of ICSLP, Beijing, China, October 16–20, 2000, pp. 504–507.
[11] G.-B. Huang, D. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cybern. (2011) 1–16.
[12] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006) 489–501.
[13] J. Chorowski, J. Wang, J.M. Zurada, Review and performance comparison of SVM- and ELM-based classifiers, Neurocomputing 128 (2013) 507–516.
[14] Y. Ye, S. Squartini, F. Piazza, Online sequential extreme learning machine in nonstationary environments, Neurocomputing 116 (20) (2012) 94–101.
[15] C. Cingolani, S. Squartini, F. Piazza, An extreme learning machine approach for training Time Variant Neural Networks, in: Proceedings of IEEE Asia Pacific Conference on Circuits and Systems, Macao, November, 2008, pp. 384–387.
[16] S. Decherchi, P. Gastaldo, R. Zunino, E. Cambria, J. Redi, Circular-ELM for the reduced-reference assessment of perceived image quality, Neurocomputing 102 (2013) 78–89.
[17] G.-B. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (1) (2010) 155–163.
[18] R. Savitha, S. Suresh, N. Sundararajan, Fast learning circular complex-valued extreme learning machine (CC-ELM) for real-valued classification problems, Inf. Sci. 187 (2012) 277–290.
[19] Q.-J.B. Loh, S. Emmanuel, ELM for the Classification of Music Genres, in: Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision, Singapore, December 5–8, 2006, pp. 1–6.
[20] C. Kim, R.M. Stern, Power-normalized coefficients (PNCC) for robust speech recognition, in: Proceedings of International Conference on Acoustics, Speech, and Signal Processing, Kyoto, Japan, March, 2012, pp. 4101–4104.
[21] R. Anitha, D.S. Satish, C.C. Sekhar, Outerproduct of trajectory matrix for acoustic modeling using support vector machines, in: Proceedings of IEEE Workshop on Machine Learning for Signal Processing, 29 September–1 October 2004, Sao Luis, Brazil, 2004, pp. 355–363.
[22] R. Leonard, A database for speaker-independent digit recognition, in: Proceedings of ICASSP 9, March 1984, pp. 328–331.
[23] E. Principi, S. Squartini, F. Piazza, D. Fuselli, M. Bonifazi, A Distributed system for recognizing home automation commands and distress Calls in the Italian Language, in: Proceedings of Interspeech, Aug. 25–29 2013, Lyon, France, 2013, pp. 2049–2053.
[24] G.-B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (October (16–18)) (2007) 3056–3062.
[25] G.-B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, Neurocomputing 70 (October (16–18)) (2008) 3460–3468.
[26] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man. Cybern. B 42 (2) (2012) 513–529.
[27] S.B. Davis, P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, in: IEEE Trans. Acoust. Speech Signal Process. 28 (4) (1980) 357–366.
[28] S. Boll, Suppression of acoustic noise in speech using spectral subtraction, IEEE Trans. Acoust. Speech Signal Process. 27 (2) (1979) 113–120.
[29] Y. Ephraim, D. Malah, Speech enhancement using a minimum mean-square error log-spectral amplitude estimator, IEEE Trans. Acoust. Speech Signal Process. ASSP-33 (1985) 443–445, April.
[30] R. Rotili, E. Principi, S. Squartini, B. Schuller, A real-time speech enhancement framework in noisy and reverberated acoustic scenarios, Cognit. Comput. (2012) 1–13.
[31] V. Stouten, Robust automatic speech recognition in time-varying environments (Ph.D. dissertation), K. U. Leuven, Leuven, the Netherlands, 2006.
[32] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, A. Acero, Robust speech recognition using a cepstral minimum-mean-square-error-motivated noise suppressor, Speech Lang. Process. 16 (July (5)) (2008) 1061–1070.
[33] E. Principi, S. Cifani, R. Rotili, S. Squartini, F. Piazza, Comparative evaluation of single-channel MMSE-based noise reduction schemes for speech recognition, J. Electr. Comput. Eng. (2010), Article ID 962103.
[34] R.D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, M. Allerhand, Complex sounds and auditory images, in: Y. Cazals, L. Demany, K. Horner (Eds.), Auditory Physiology and Perception, vol. 83, Pergamon Press, Oxford, UK, 1992, pp. 429–446.
[35] R. Solera-Ure na, J. Padrell-Sendra, D. Martín-Iglesias, A. Gallardo-Antolín, C. Peláez-Moreno, F. Díaz-de María, SVMs for automatic speech recognition: a survey, in: Y. Stylianou, M. Faundez-Zanuy, A. Esposito (Eds.), Progress in Nonlinear Speech Processing, ser. Lecture Notes in Computer Science, vol. 4391, Springer, Berlin, Heidelberg, 2007, pp. 190–216.
[36] C. Bishop, Pattern Recognition and Machine Learning, Springer Science + Business Media, LLC, New York, 2006.
[37] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 27.
[38] H.-M. Zhou, G.-B. Huang. Official ELM toolbox. Accessed 20/12/13. [Online]. Available: url: 〈http://www.ntu.edu.sg/home/egbhuang/〉.
[39] E. Cambria, A. Hussain, C. Havasi, C. Eckl. Common sense computing: From the society of mind to digital intuition and beyond. In: LNCS, Springer, vol. 5707, (2009) pp. 252–259.
[40] E. Principi, S. Squartini, F. Piazza, Power Normalized Cepstral Coefficients based supervectors and i-vectors for small vocabulary speech recognition, in: Proceedings of the International Joint Conference on Neural Networks, Beijing, China, July 6-11, 2014, pp. 3562–3568.
[41] P. Gastaldo, R. Zunino, E. Cambria, S. Decherchi, Combining ELM with Random Projections, IEEE Intelligent Systems 28 (6) (2013) 46–48.
[42] E. Cambria, P. Gastaldo, F. Bisio, R. Zunino, An ELM-based model for affective analogical reasoning, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.01.064.
[43] E. Cambria, A. Hussain, Sentic album: Content-, concept-, and context-based online personal photo management system, Cognitive Computation 4 (4) (2012) 477–496.

Emanuele Principi was born in Senigallia (Ancona), Italy, on January 1978. He received the M.S. degree in electronic engineering (with honors) from Università Politecnica delle Marche (Italy) in 2004. He received his Ph.D. degree in 2009 in the same university under the supervision of Prof. Francesco Piazza. In November 2006 he joined the 3MediaLabs research group coordinated by Prof. Francesco Piazza at Università Politecnica delle Marche where he collaborated to several regional and European projects on audio signal processing, developing distributed speaker recognition solutions. Dr. Principi is the author and coauthor of several international scientific peer-reviewed articles in the area of speech enhancement for robust speech recognition, and keyword spotting systems. His current research interests are in the area of digital signal processing, including speech enhancement for automatic speech and speaker recognition systems, acoustic event classification, pattern recognition, and real-time digital signal processing on embedded platforms.

Stefano Squartini (Senior Member IEEE and Member AES/ISCA) was born in Ancona, Italy, on March 1976. He got the Italian Laurea with honors in electronic engineering from University of Ancona (now Polytechnic University of Marche, UnivPM), Italy, in 2002. He obtained his Ph.D. at the same university (November 2005). He got a funded Visiting Research Fellowship at Department of Computing Science at University of Stirling (August–October 2003). Moreover he was a Visiting Scholar at Electrical and Computer Engineering Department at University of Illinois at Chicago, (March–September 2004). He worked also as a post-doctoral researcher at UnivPM from June 2006 to November 2007, when he joined the DII (Department of Information Engineering) as an Assistant Professor in Circuit Theory. In 2012 he was a guest lecturer at TUM (Munich University of Technology) in Munich/Germany. His current research interests are in the area of digital signal processing and computational intelligence, with focus on speech/audio processing, cognitive systems and smart grids. Dr. Squartini is one of the founding members of the research group 3MediaLabs, and has actively participated to various (funded) regional, national and European projects on multimedia Digital Signal Processing and Smart Home Energy Management. He is co-founder and CEO of the UnivPM Spin-off DowSee, an engineering company developing environmentally sustainable ICT solutions for the rational use and saving of energy in smart grids. He is author and coauthor of many international scientific peer-reviewed articles (more than 100), and Associate Editor for the IEEE Transactions on Neural Networks and Learning Systems (since 2010) and member of the Cognitive Computation Editorial Board (starting from 2011). He is a regular reviewer for several (IEEE, Springer, Elsevier) Journals, Books and Conference Proceedings and in the recent past he organized several Special Sessions at international conferences with peer-reviewing and Special Issues of ISI journals. He joined the Organizing and the Technical Programme Committees of more than 30 International Conferences and Workshops in the recent past. He is member of the European Network of Excellence EUCOGIII and of the Executive Board of the SIREN (Italian Society of Neural Networks), and responsible for his University participation in the Texas Instruments European University Program. He is also member of the Texas Instrument Expert Advisory Panel.

Erik Cambria Erik Cambria received his B.Eng. and M. Eng. with honours in Electronic Engineering from the University of Genoa in 2005 and 2008, respectively. In 2012, he was awarded his Ph.D. in Computing Science and Mathematics, following the completion of a Cooperative Awards in Science and Engineering (CASE) project born from the collaboration between the University of Stirling, the MIT Media Lab, and Sitekit Solutions Ltd., which included internships at HP Labs India, the Chinese Academy of Sciences, and Microsoft Research Asia. From August 2011 to May 2014, Erik was a research scientist at the National University of Singapore (Cognitive Science Programme) and an associate researcher at the Massachusetts Institute of Technology (Synthetic Intelligence Project). Today, Erik is an assistant professor at Nanyang Technological University (School of Computer Engineering), where he teaches natural language processing and data mining. His research interests include concept-level sentiment analysis, affective common-sense reasoning, noetic natural language processing, and intention awareness. Erik is editorial board co-chair of Cognitive Computation, associate editor of Knowledge-Based Systems, and guest editor of many other top-tier AI journals, including three issues of IEEE Intelligent Systems, two issues of IEEE CIM, and one issue of Neural Networks. He is also involved in several international conferences as workshop series organizer, e.g., ICDM SENTIRE since 2011, program chair, e.g., ELM since 2012, PC member, e.g., UAI in 2014, tutorial organizer, e.g., WWW in 2014, special track chair, e.g., AAAI FLAIRS in 2015, and keynote speaker, e.g., CICLing in 2015.

Francesco Piazza was born in Jesi, Italy, on February 1957. He got the Italian Laurea with honors in Electronic Engineering from University of Ancona, Italy, in 1981. From 2000, he is a full professor of Electrical Science at the Università Politecnica delle Marche (UNIVPM), Ancona, Italy. Among other academic services, at this university he has been the head of both the Electronic Engineering course (3+2 years) and the DEIT Ph.D. course. He has been supervisor of many Ph.D. students, two of them awarded for the best Italian dissertation on Artificial Neural Networks topics. Before the academic career, he worked at the Olivetti OSAI as software engineer and was co-founder and CEO of TECMAR Sc.r.l. a small high tech SME working on DSP algorithms and software. During the academic career, he was also co-founder of Leaff Engineering S.r.l., Sensible Logic S.r.l. and DowSee S.r.l. three SME spin-off of UNIVPM, working respectively on DSP and multimedia, Semantic web and metadata and sustainable ICT solutions for energy saving. At UNIVPM he founded and leads the DSP Research Group (3Medialabs) and its related laboratories A3lab and Semedia. Together with his collaborators and students, Professor Piazza has given many contributions in the area of digital signal processing in particular on multichannel blind and non-blind adaptive DSP algorithms and circuits, artificial neural networks for signal processing, speech and audio processing. In his work, he has got 2 patents and published more than 300 research papers in technical books, peer-reviewed journals and conference proceedings. He is member of IEEE (Circuits & Systems, Signal Processing and Computer Societies), ACM (Association for Computing Machinery), AES (Audio Engineering Society) and SAE (Society of Automotive Engineers). He has been member of many technical program committees of international conferences and workshops. He has been member of IEEE CAS, Blind Signal Processing and other Technical Committees, member of the management committee of the EU actions COST-277 and COST-2102 and of several European research projects. He is reviewer for many IEEE, Springer and Elsevier technical journals and conferences. His research work has been supported by several national and international organizations (Ministero dell'Istruzione, dell'Università e della Ricerca, Consiglio Nazionale delle Ricerche, ENEA, the European Commission and others) and private companies (Roland, Korg, Radvision, Atmel, Google, Indesit, Ferretti Group, Faital, Texas Instruments and others).