Pattern Recognition Letters 000 (2014) 1–9

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Emotion recognition from speech signals via a probabilistic echo-state network ☆

Edmondo Trentin [a,*], Stefan Scherer [b], Friedhelm Schwenker [c]

[a] DIISM, Università di Siena, V. Roma 56, I-53100 Siena, Italy
[b] USC Institute for Creative Technologies, 12015 Waterfront Drive, 90094-2536 Playa Vista, CA, USA
[c] Institute of Neural Information Processing, Ulm University, Oberer Eselsberg, D-89069 Ulm, Germany

ARTICLE INFO

Article history:
Received 26 February 2014
Available online xxx

Keywords:
Emotion recognition
Echo state network
Sequence clustering
Semi-supervised learning

ABSTRACT

The paper presents a probabilistic echo-state network (π-ESN) for density estimation over variable-length sequences of multivariate random vectors. The π-ESN stems from the combination of the reservoir of an ESN and a parametric density model based on radial basis functions. A constrained maximum likelihood training algorithm is introduced, suitable for sequence classification. Extensions of the algorithm to unsupervised clustering and semi-supervised learning (SSL) of sequences are proposed. Experiments in emotion recognition from speech signals are conducted on the WaSeP© dataset. Compared with established techniques, the π-ESN yields the highest recognition accuracies, and shows interesting clustering and SSL capabilities.

1. Introduction

Emotional communication in human-to-human interaction is indefeasible and often crucial. It may convey information that would be missed otherwise (e.g., on the speaker's unspoken thoughts, or on the underlying context in which a dialogue is taking place), information that could grant its real meaning to what is factually spoken. Therefore, in order to render human-computer interaction (HCI) more natural and efficient, machines are sought that can recognize, understand, and express emotional states. In fact, although HCI has been taking a more and more relevant place in our everyday lives, the science of emotion modeling and recognition from audio or video signals is still in its infancy.

On the other hand, several pioneering approaches to emotion recognition from speech signals can be found in the literature. Vlasenko et al. [1] apply Gaussian mixture models (GMM) and hidden Markov models (HMM) defined at both the frame- and turn-level representations of the audio signals, while Wagner et al. [2] thoroughly analyze the behavior of HMMs and support vector machines (SVM) using Mel-cepstra [3] and energy-based features. Schwenker et al. [4] investigate the use of the SVM-GMM Supervector approach relying on PLP and ModSpec features [5]. Dellaert et al. [6] classify speech signals into 4 broad classes of emotions by applying a mixture of k-nearest neighbor [7] experts (with $k = 11$) estimated on different subsets of acoustic features. Depending on the method and on the dataset, these studies observed recognition accuracies ranging mostly between 60% and 85%.

This paper introduces and investigates a novel approach to emotion recognition and clustering from speech signals, the probabilistic echo state network (π-ESN). This article is the journal version of a workshop communication [8], and introduces new algorithms, faces new setups (unsupervised, semi-supervised), reports on a much wider and deeper experimental investigation, and offers an in-depth discussion of the key findings.

The π-ESN realizes a parametric model of the probability density function (pdf) underlying the distribution of a sample of variable-length sequences of real-valued, multivariate random vectors. The model relies on the hybridization between an echo state network (ESN) [9] and a constrained radial basis function (RBF)-like network [10].

While RBFs are feed-forward networks known to realize linear combinations of Gaussian kernels evaluated on a fixed-dimensional, real-valued vector space, ESNs are a particular subclass of the broad family of recurrent neural networks (RNN). A schematic representation of an ESN is shown in Fig. 1. The most important part of the network is its recurrent reservoir. It is a large collection of units that are loosely and randomly connected to each other. The probability of a connection holding between a generic pair of units a_i, a_j is a decreasing function of the reservoir size, and lies typically in the $(0.02, 0.1)$ range [9]. The connection weights W of the reservoir are drawn at random, as well. Additive-sigmoid transfer functions $\psi(.)$ are associated with the units. The input layer is fully connected with the reservoir via connections having weight matrix W^{in}. The reservoir, in turn, is

☆ This paper has been recommended for acceptance by Sanniti di Baja.
* Corresponding author. Tel.: +39 0577 233601; fax: +39 0577 233602.
E-mail address: trentin@dii.unisi.it (E. Trentin).

http://dx.doi.org/10.1016/j.patrec.2014.10.015
0167-8655/

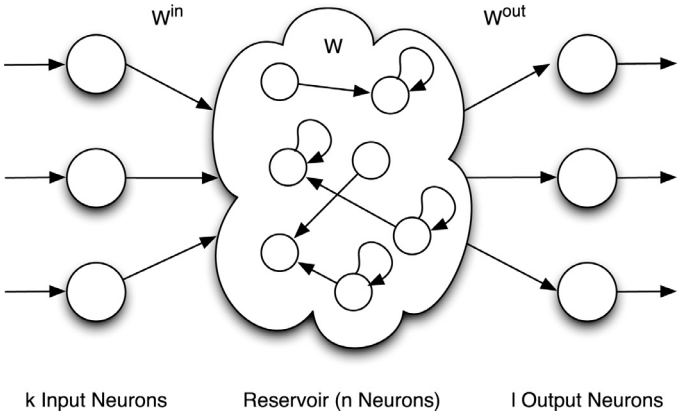

Fig. 1. Schematics of an ESN. Inputs and outputs are fully connected to the reservoir via W^{in} and W^{out}, respectively. Connections of the reservoir and their weights W are random.

fully connected (with weight matrix W^{out}) to the linear output layer. The loose connectivity of the reservoir leads to the formation of small cycles of units that are recursively connected to each other. These cycles are sensitive to certain dynamic phenomena in the signal received through the input units and from other adjacent neurons in the reservoir. Since there are feed-backward and recursive connections within the reservoir, the output at any given time t is a function of the current input pattern and of the *state* (i.e., the value yielded by the corresponding transfer function) of each of the other units in the reservoir at time t (which can be thought of as a non-linearly filtered history of all the inputs up to time $t-1$).

ESNs found application to such different tasks as classification, pattern generation, and control [9,11–13]. They offer advantages with respect to traditional RNNs, e.g. their stability toward noisy inputs [11] and the efficient weight adaptation method [13]. Moreover, ESNs possess the universal computation property, i.e. they can approximate arbitrarily well any non linear filter having bounded memory [14]. Since there is no backpropagation of partial derivatives through the reservoir (albeit there are bounds on the ESN memory [15] in the general case), ESNs do not suffer from major learning problems that affect classic RNNs [16]. This is utterly relevant to our purposes, making the ESN a viable candidate for encoding multivariate time sequences, including the modeling of typical dynamics found in the speech signals such as the prosody of emotional expressions. The expectation is corroborated by the empirical evidence reported in [8,12].

The basic idea pursued in the paper is that the recurrent reservoir of the ESN realizes an encoding of an input sequence by means of the pattern of activation of its state units. The trainable state-to-output weights and the linear output layer of the ESN are replaced by an RBF architecture. The RBF is trained in order to estimate the pdf underlying the distribution of these patterns of activation within the encoding space. Training is realized according to a constrained gradient-ascent algorithm, presented in Section 2, aimed at the maximization of the likelihood of the parameters of the model given the input sequence. Constraints are required to ensure that the estimated model satisfies the axioms of probability. The training scheme is inherently unsupervised and non-discriminative, along the line of statistical parametric pdf estimation techniques that rely on the maximum-likelihood (ML) criterion [7]. Nonetheless, it can be applied in classification tasks by using a separate π-ESN to estimate the class-conditional pdf [7] for each of the classes $\omega_1, \ldots, \omega_c$ involved in the problem, and by applying Bayes decision rule.

The algorithm has been introduced in the framework of sequence classification, assuming that class-labels of specific emotions are associated with all the acoustic observation sequences in the training set. Unfortunately, there are two major issues with this assumption: (1) the categorization of emotions into classes is intrinsically ill-defined, possibly overlapping, and even subjective; (2) emotion processing in real-world scenarios (i.e., not relying on pseudo-emotions simulated by actors) would require huge amounts of mostly unlabeled spontaneous speech data. These issues are faced in the paper by extending the π-ESN training algorithm to fit the unsupervised clustering and the semi-supervised learning (SSL) setups [17] with adaptive number of clusters/classes. This is achieved in Section 3 by exploiting the probabilistic nature of the π-ESN within a (quasi)cross-validated likelihood model selection strategy [18].

Section 4 reports (and discusses in depth) experiments based on a corpus containing pseudo-words spoken in six different emotional prosodies, WaSeP© [19]. The behavior of the π-ESN in supervised, unsupervised, and semi-supervised tasks is analyzed and (favorably) compared w.r.t. established techniques. Final remarks are drawn in Section 5.

2. The probabilistic echo-state network

As we stated in the previous section, a separate, class-specific, and independent π-ESN is used for each emotion involved in the task. Thence, in the following we will focus on a generic π-ESN, trained over the corresponding emotion-specific training sample, with the understanding that the algorithm has to be subsequently applied to as many π-ESNs as the number of classes at hand. Albeit intrinsically unsupervised, the algorithm is foremost oriented to supervised classification tasks.

Let $\mathcal{T} = \{\mathcal{Y}_1, \ldots, \mathcal{Y}_N\}$ be a random sample of N acoustic observation sequences, identically and independently drawn (iid) from the unknown pdf $p(\mathcal{Y})$. The π-ESN is devised as a plausible parametric model of $p(\mathcal{Y})$, namely $p(\mathcal{Y}|\theta)$, determined uniquely by the value of its parameters $\theta = (\theta_1, \ldots, \theta_k)$. A parameter learning algorithm is pursued that maximizes the likelihood $p(\mathcal{T}|\theta)$ of θ given $\mathcal{T}$. Relying on the iid assumption, we can write $p(\mathcal{T}|\theta) = \prod_{i=1}^{N} p(\mathcal{Y}_i|\theta)$. Before proceeding, it is necessary to specify a well-defined form for $p(\mathcal{Y}|\theta)$, as follows. Let us assume the existence of an integer d and of two functions, $\phi : \{\mathcal{Y}\} \to \Re^d$ (where $\{\mathcal{Y}\}$ is the universe of all possible observation sequences) and $\hat{p} : \Re^d \to \Re$, s.t. $p(\mathcal{Y})$ can be decomposed as:

$$p(\mathcal{Y}) = \hat{p}(\phi(\mathcal{Y})). \tag{1}$$

It is seen that there exist (infinite) functions $\phi(.)$ and $\hat{p}(.)$ that satisfy Equation (1), the most trivial being $\phi(\mathcal{Y}) = p(\mathcal{Y}), \hat{p}(x) = x$. We call $\phi(.)$ the *encoding*, while $\hat{p}(.)$ is simply referred to as the "likelihood". Again, we assume parametric models $\phi(\mathcal{Y}|\theta_\phi)$ and $\hat{p}(\mathbf{x}|\theta_{\hat{p}})$ for the encoding and for the likelihood, respectively, and we set $\theta = (\theta_\phi, \theta_{\hat{p}})$ and $p(\mathcal{Y}|\theta) = \hat{p}(\phi(\mathcal{Y}|\theta_\phi)|\theta_{\hat{p}})$.

A hybrid two-block connectionist/statistical model is proposed for $p(\mathcal{Y}|\theta)$ as follows. The function $\phi(\mathcal{Y}|\theta_\phi)$ is realized via an ESN, suitable to map sequences $\mathcal{Y}$ into real vectors $\mathbf{x}$. Let θ_ϕ be the set of the ESN weights. A RBF-like network is then used to model $\hat{p}(\mathbf{x}|\theta_{\hat{p}})$, where $\theta_{\hat{p}}$ is the parameter vector of the RBF. In order to ensure that a pdf is obtained, constraints have to be placed on the hidden-to-output connection weights of the RBF (assuming that normalized Gaussian kernels are used).

First, let us focus on the ESN-based model for $\phi(\mathcal{Y}|\theta_\phi)$. The topology and the weight matrix W of the reservoir are generated at random. W is normalized s.t. its spectral radius is $\alpha \leq 1$ [13]. This scaling of the weight matrix is accomplished so that the maximal eigenvalue $\lambda_{\max}$ of W satisfies $|\lambda_{\max}| \leq 1$. The encoding of $\mathcal{Y} = \mathbf{y}_1, \ldots, \mathbf{y}_T$ (where T is not fixed, but sequence-specific) is accomplished as follows:

1. initialize the state $\hat{\mathbf{x}}$ of the reservoir at random
2. feed the ESN with the first L acoustic feature vectors[1] $\mathbf{y}_1, \ldots, \mathbf{y}_L$
3. save the resulting state $\mathbf{x}_0$ of the ESN at time L as the starting state

[1] This is done to minimize the influence of the random initial conditions [13].

4. set the ESN in $\mathbf{x}_0$, and sequentially feed the ESN with inputs $\mathbf{y}_1, \ldots, \mathbf{y}_T$
5. let $\mathbf{x}_t$ denote the reservoir state at time t, for $t = 1, \ldots, T$
6. let the *encoding* $\mathbf{x}$ of $\mathcal{Y}$ be the state $\mathbf{x}_T$ of the reservoir at time T, i.e. $\mathbf{x} = \mathbf{x}_T$

where the generic state $\mathbf{x}_t$ of the reservoir is the real-valued vector of the output values from its units at time t, according to the system equation $\mathbf{x}_t = \psi(W^{\text{in}}\mathbf{y}_t + W\mathbf{x}_{t-1})$. Once the encoding of $\mathcal{Y}$ is accomplished, $\mathbf{x}$ is fed to the RBF. Since the weights θ_ϕ of the ESN are not optimized, we focus on the ML estimation of the RBF parameters $\theta_{\hat{p}}$ given $\mathcal{T}$, maximizing the quantity

$$p(\mathcal{T}|\theta_{\hat{p}}) = \prod_{i=1}^{N} \hat{p}(\phi(\mathcal{Y}_i|\theta_\phi)|\theta_{\hat{p}}). \tag{2}$$

A hill-climbing algorithm for maximizing $p(\mathcal{T}|\theta_{\hat{p}})$ w.r.t. $\theta_{\hat{p}}$ is obtained in two steps. (1) *Initialization*: Start with some initial, e.g. random, assignment of values to the RBF parameters. (2) *Gradient-ascent*: Repeatedly apply a learning rule in the form $\Delta\theta_{\hat{p}} = \eta\nabla_{\theta_{\hat{p}}}\{\prod_{i=1}^{N}\hat{p}(\phi(\mathcal{Y}_i|\theta_\phi)|\theta_{\hat{p}})\}$ with $\eta \in \Re^+$. This is a batch learning rule. In practice, neural network learning may be simplified, yet even improved, with the adoption of an on-line training scheme that prescribes $\Delta\theta_{\hat{p}} = \eta\nabla_{\theta_{\hat{p}}}\{\hat{p}(\phi(\mathcal{Y}|\theta_\phi)|\theta_{\hat{p}})\}$ upon presentation of each individual training sequence $\mathcal{Y}$. The gradient is calculated w.r.t. two distinct families of adaptive parameters.

(1) Mixing parameters $c_1, \ldots, c_n$, i.e. the hidden-to-output weights of the RBF network. A constrained ML estimation process is required to ensure that $c_j \in (0, 1)$ for $j = 1, \ldots, n$, and that $\sum_{j=1}^{n} c_j = 1$. To satisfy the constraints we introduce n latent parameters $\gamma_1, \ldots, \gamma_n$, which are unconstrained, and we let

$$c_i = \frac{\varsigma(\gamma_i)}{\sum_{j=1}^{n}\varsigma(\gamma_j)}, \quad i = 1, \ldots, n \tag{3}$$

where $\varsigma(x) = 1/(1 + e^{-x})$. Each γ_i is then treated as an unknown parameter to be estimated via ML.

(2) d-dimensional mean vector μ_i and $d \times d$ covariance matrix Σ_i for each of the Gaussian kernels $K_i(\mathbf{x}) = G(\mathbf{x}; \mu_i, \Sigma_i)$, $i = 1, \ldots, n$ of the RBF, where $G(\mathbf{x}; \mu_i, \Sigma_i)$ denotes a multivariate Gaussian pdf having mean vector μ_i, covariance matrix Σ_i, and evaluated over the random vector $\mathbf{x}$. A common (yet effective) simplification is to consider diagonal covariance matrices, i.e. independence among the components of the input vector $\mathbf{x}$. This assumption leads to the following three major consequences: (i) modeling properties are not affected significantly, according to [20]; (ii) generalization capabilities of the overall model may turn out to be improved, since the number of free parameters is reduced; (iii) ith multivariate kernel K_i may be expressed in the form of a product of d univariate Gaussian pdfs as:

$$K_i(\mathbf{x}) = \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp\left\{-\frac{1}{2}\left(\frac{x_j - \mu_{ij}}{\sigma_{ij}}\right)^2\right\} \tag{4}$$

i.e., the free parameters to be estimated are the means μ_{ij} and the standard deviations σ_{ij}, for each kernel $i = 1, \ldots, n$ and for each component $j = 1, \ldots, d$ of the input space. Recapitulating, the RBF system equation can thus be expressed in the concise form $\hat{p}(\mathbf{x}|\theta_{\hat{p}}) = \sum_{i=1}^{n} c_i G(\mathbf{x}; \mu_i, \Sigma_i)$.

An explicit form for $\Delta\theta_{\hat{p}}$ is now obtained by calculating the partial derivatives of $\hat{p}(\phi(\mathcal{Y}|\theta_\phi)|\theta_{\hat{p}})$ w.r.t. the two families of free parameters in the model. For a generic mixing parameter $c_i, i = 1, \ldots, n$, from Equation (3) and since $p(\mathcal{Y}|\theta) = \sum_{k=1}^{n} c_k K_k(\mathbf{x})$ we have

$$\begin{aligned}\frac{\partial\hat{p}(\phi(\mathcal{Y}|\theta_\phi)|\theta_{\hat{p}})}{\partial\gamma_i} &= \sum_{j=1}^{n}\frac{\partial p(\mathcal{Y}|\theta)}{\partial c_j}\frac{\partial c_j}{\partial\gamma_i} \\ &= \sum_{j=1}^{n} K_j(\mathbf{x})\frac{\partial}{\partial\gamma_i}\left(\frac{\varsigma(\gamma_j)}{\sum_{k=1}^{n}\varsigma(\gamma_k)}\right) \\ &= K_i(\mathbf{x})\frac{\varsigma'(\gamma_i)}{\sum_k\varsigma(\gamma_k)} - \sum_j K_j(\mathbf{x})\frac{\varsigma(\gamma_j)\varsigma'(\gamma_i)}{[\sum_k\varsigma(\gamma_k)]^2} \\ &= \frac{\varsigma'(\gamma_i)}{\sum_k\varsigma(\gamma_k)}\{K_i(\mathbf{x}) - p(\mathcal{Y}|\theta)\}.\end{aligned} \tag{5}$$

As for the means μ_{ij} and the standard deviations σ_{ij} we proceed as follows. Let θ_{ij} denote the free parameter, i.e. μ_{ij} or σ_{ij}, to be estimated. It is seen that $\frac{\partial\hat{p}(\phi(\mathcal{Y}|\theta_\phi)|\theta_{\hat{p}})}{\partial\theta_{ij}} = c_i\frac{\partial K_i(\mathbf{x})}{\partial\theta_{ij}}$, where the calculation of $\frac{\partial K_i(\mathbf{x})}{\partial\theta_{ij}}$ is accomplished as follows. First, let us observe that for any real-valued, differentiable function $f(.)$ this property holds true: $\frac{\partial f(.)}{\partial x} = f(.)\frac{\partial\log[f(.)]}{\partial x}$. Thence, from Equation (4) we can write

$$\begin{aligned}\frac{\partial K_i(\mathbf{x})}{\partial\theta_{ij}} &= K_i(\mathbf{x})\frac{\partial\log K_i(\mathbf{x})}{\partial\theta_{ij}} \\ &= K_i(\mathbf{x})\frac{\partial}{\partial\theta_{ij}}\sum_{k=1}^{d}\left\{-\frac{1}{2}\left[\log(2\pi\sigma_{ik}^2) + \left(\frac{x_k - \mu_{ik}}{\sigma_{ik}}\right)^2\right]\right\}.\end{aligned} \tag{6}$$

For the means, i.e. $\theta_{ij} = \mu_{ij}$, Equation (6) yields $\frac{\partial K_i(\mathbf{x})}{\partial\mu_{ij}} = K_i(\mathbf{x})\frac{x_j - \mu_{ij}}{\sigma_{ij}^2}$. For the covariances, i.e. $\theta_{ij} = \sigma_{ij}$, Equation (6) takes the form $\frac{\partial K_i(\mathbf{x})}{\partial\sigma_{ij}} = K_i(\mathbf{x})\frac{\partial}{\partial\sigma_{ij}}\{-\frac{1}{2}\log(2\pi\sigma_{ij}^2) - \frac{1}{2}(\frac{x_j - \mu_{ij}}{\sigma_{ij}})^2\}$, that is $\frac{\partial K_i(\mathbf{x})}{\partial\sigma_{ij}} = \frac{K_i(\mathbf{x})}{\sigma_{ij}}\{(\frac{x_j - \mu_{ij}}{\sigma_{ij}})^2 - 1\}$ which completes the calculation of $\Delta\theta_{\hat{p}}$. The training algorithm is guaranteed to increase the likelihood up to a (possibly local) maximum. Due to the universal approximation properties of both ESNs [14] and RBFs [10] it is seen that, under the same mild conditions assumed therein, the π-ESN is a universal model of any continuous and bounded pdf of multivariate sequences. In classification tasks, at test time the output of the ith π-ESN is used as an estimate of the ith class-conditional pdf in the right-hand side of Bayes theorem [7] where it is combined with the class-prior probabilities, s.t. Bayes decision rule can be applied.

3. Extension of the π-ESN to unsupervised and semi-supervised frameworks

The probabilistic nature of the π-ESN can be exploited in an interesting, yet rather straightforward manner to fully unsupervised and semi-supervised frameworks by means of the cross-validated likelihood criterion [18]. These extensions are suitable to generic problems of clustering or SSL of sequential data.

In the unsupervised (clustering) case we rely on the π-ESN to devise an algorithm for partitioning a collection $\mathcal{U} = \{\mathcal{Y}_1, \ldots, \mathcal{Y}_v\}$ of unlabeled acoustic observation sequences into k clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$, where a maximum cross-validated likelihood estimation of k is realized. In other words, the algorithm discovers spontaneously the optimal amount of "emotional clusters" according to a well-defined probabilistic criterion. Algorithm 1 hands out the pseudo-code (the notation is inherited from the previous section). It is seen that the algorithm can be readily simplified to the case of a pre-defined number k of clusters, if desired. While in the previous section each class ω_i was modeled relying on a mixture of RBF kernels, in this setup (as in k-means) each Gaussian kernel corresponds to an individual cluster, and clusters do not necessarily correspond to "classes" (a class could be made up of several clusters; or, certain unsupervised tasks could not even involve explicit classes at all).

Please cite this article as: E. Trentin et al., Emotion recognition from speech signals via a probabilistic echo-state network, Pattern Recognition Letters (2014), http://dx.doi.org/10.1016/j.patrec.2014.10.015

Algorithm 1: Unsupervised clustering via π-ESN.

Data: unsupervised dataset of sequences $\mathcal{U} = \{\mathcal{Y}_1, \ldots, \mathcal{Y}_v\}$
Result: the optimal number of clusters k and the corresponding clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$
split $\mathcal{U}$ into $\mathcal{T}$ and $\mathcal{V}$ s.t. $\mathcal{T} \cup \mathcal{V} = \mathcal{U}$, $\mathcal{T} \cap \mathcal{V} = \varnothing$;
$\mathcal{L} \leftarrow 0$;
$\mathcal{L}_{\max} \leftarrow 0$;
$k \leftarrow 1$;
while $\mathcal{L} \geq \mathcal{L}_{\max}$ **do**
  initialize a π-ESN with k kernels;
  train the π-ESN on $\mathcal{T}$;
  $\mathcal{L} \leftarrow p(\mathcal{V}|\pi\text{-ESN})$;
  if $\mathcal{L} \geq \mathcal{L}_{\max}$ **then**
    $\mathcal{L}_{\max} \leftarrow \mathcal{L}$;
    $k_{\max} \leftarrow k$;
    $k \leftarrow k + 1$;
  end
end
initialize a π-ESN with $k_{\max}$ kernels;
train the π-ESN on $\mathcal{U}$;
for $i \leftarrow 1$ **to** $k_{\max}$ **do**
  $\mathcal{C}_i \leftarrow \varnothing$;
end
for $j \leftarrow 1$ **to** v **do**
  $\mathbf{x} \leftarrow$ the ESN encoding of $\mathcal{Y}_j$;
  $p_{\max} = 0$;
  for $i \leftarrow 1$ **to** $k_{\max}$ **do**
    $p \leftarrow K_i(\mathbf{x})$;
    if $p > p_{\max}$ **then**
      $c \leftarrow i$;
      $p_{\max} \leftarrow p$;
    end
  end
  $\mathcal{C}_c \leftarrow \mathcal{C}_c \cup \{\mathcal{Y}_j\}$;
end
return $k_{\max}, \mathcal{C}_1, \ldots, \mathcal{C}_{k_{\max}}$

First, the algorithm partitions $\mathcal{U}$ into a training set $\mathcal{T}$ and a validation set $\mathcal{V}$. Next, the optimal k is searched for, iterating over increasing values of k as long as the likelihood on $\mathcal{V}$ of the resulting π-ESN with k kernels increases. Note that while $\lim_{k \to |\mathcal{T}|} p(\mathcal{T}|\pi\text{-ESN}) \to \infty$ ($|\mathcal{T}|$ Gaussian kernels, one per each encoding $\mathbf{x}_j, j = 1, \ldots, |\mathcal{T}|$, having null covariance and the mean equal to $\mathbf{x}_j$; i.e., the kernels reduce to Dirac's deltas), $p(\mathcal{V}|\pi\text{-ESN})$ is bounded and it increases (for $k = 1, 2, \ldots$) up to a (possibly local) maximum for $k = k_{\max}$. Roughly speaking, the quantity $p(\mathcal{V}|\pi\text{-ESN})$ is a measure of the generalization capability of the π-ESN, and it is expected to worsen once the π-ESN begins (for $k > k_{\max}$) to overfit the data. Note that the computation of the likelihoods, e.g. $p(\mathcal{V}|\pi\text{-ESN})$, takes place via Equation (2). In the second part of the algorithm, the $k_{\max}$ clusters are finally created by assuming (in a k-means fashion [7]) that ith cluster is identified by the mean vector and the covariance matrix of the corresponding (ith) RBF kernel. A generic sequence $\mathcal{Y} \in \mathcal{U}$ having encoding $\mathbf{x} = \phi(\mathcal{Y})$ is assigned to cluster $\mathcal{C}_i$ if $K_i(\phi(\mathcal{Y})) > K_j(\phi(\mathcal{Y}))$ for all $j = 1, \ldots, k_{\max}$, $j \neq i$. In practice, a maximum probability clustering technique is obtained, while traditional algorithms (such as the k-means) use minimum distance criteria. As it happens with unsupervised clustering algorithms, whose outcome (the data partitioning) may or may not be in strict relation with the presence of classes of an underlying classification problem, the unsupervised π-ESN is expected to partition the speech signals into well-separated clusters, having high internal cohesion; to which extent this clustering reflects emotion-related properties of the speech signals is the matter of empirical evaluation.

Algorithm 2: Semi-supervised training of π-ESNs.

Data: $c, k, T, \mathcal{S}, \mathcal{U}, \theta_{\text{in}}, \theta_{\text{out}}$
Result: new c, the class-specific π-ESNs $\varphi_1, \ldots, \varphi_c$
for $i \leftarrow 1$ **to** c **do**
  $\mathcal{S}_i \leftarrow \{\mathcal{Y} \mid \mathcal{Y} \in \mathcal{S}, \mathcal{Y} \in \omega_i\}$;
end
for $t \leftarrow 1$ **to** T **do**
  if $k > |\mathcal{U}|$ **then**
    $k \leftarrow |\mathcal{U}|$;
  end
  if $k = 0$ **then**
    break;
  end
  $\mathcal{V} \leftarrow$ a random subsample $\{\mathcal{Y}_1, \ldots, \mathcal{Y}_k\}$ of $\mathcal{U}$;
  let $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{V}$ and $\upsilon \leftarrow \sum_{i=1}^{c} |\mathcal{S}_i|$;
  for $i \leftarrow 1$ **to** c **do**
    $P_i \leftarrow |\mathcal{S}_i|/\upsilon$;
    Regular-Training$(\varphi_i, \mathcal{S}_i)$;
  end
  $\mathcal{X} \leftarrow \varnothing$;
  for $j \leftarrow 1$ **to** k **do**
    let $max \leftarrow 0$ and $winner \leftarrow 0$;
    for $i \leftarrow 1$ **to** c **do**
      $y_i \leftarrow \frac{P_i \varphi_i(\mathcal{Y}_j)}{\sum_{l=1}^{c} P_l \varphi_l(\mathcal{Y}_j)}$;
      if $y_i > max$ **then**
        let $winner \leftarrow i$ and $max \leftarrow y_i$;
      end
    end
    if $y_{\text{winner}} > \theta_{\text{in}}$ **then**
      $\mathcal{S}_{\text{winner}} \leftarrow \mathcal{S}_{\text{winner}} \cup \{\mathcal{Y}_j\}$;
    else
      $\mathcal{X} = \mathcal{X} \cup \{\mathcal{Y}_j\}$;
    end
  end
  if $\mathcal{X} \neq \varnothing$ **then**
    let $c \leftarrow c + 1$ and $\mathcal{S}_c \leftarrow \mathcal{X}$;
    for $i \leftarrow 1$ **to** c **do**
      $P_i \leftarrow |\mathcal{S}_i|/(\upsilon + |\mathcal{S}_c|)$;
    end
    Regular-Training$(\varphi_c, \mathcal{S}_c)$;
  end
  for $i \leftarrow 1$ **to** c **do**
    $\mathcal{X} \leftarrow \mathcal{S}_i$;
    for $j \leftarrow 1$ **to** $|\mathcal{X}|$ **do**
      $X_j \leftarrow j$th item in $\mathcal{X}$;
      $y_i \leftarrow \frac{P_i \varphi_i(X_j)}{\sum_{l=1}^{c} P_l \varphi_l(X_j)}$;
      if $y_i < \theta_{\text{out}}$ **then**
        $\mathcal{S}_i \leftarrow \mathcal{S}_i \setminus \{X_j\}$;
        $\mathcal{U} \leftarrow \mathcal{U} \cup \{X_j\}$;
      end
    end
  end
end
return $c, \varphi_1, \ldots, \varphi_c$

The idea of exploiting the probabilistic output of π-ESNs along with a statistical criterion (the maximum a-posteriori criterion, in this case) to account for the distribution of unlabeled data can be contextualized to a semi-supervised learning setup. The pseudo-code is shown in Algorithm 2. The basic idea goes as follows. Let us assume we start with c classes of emotions $\omega_1, \ldots, \omega_c$ and a collection

Please cite this article as: E. Trentin et al., Emotion recognition from speech signals via a probabilistic echo-state network, Pattern Recognition Letters (2014), http://dx.doi.org/10.1016/j.patrec.2014.10.015

of available acoustic sequences $\mathcal{T} = \mathcal{S} \cup \mathcal{U}$, $\mathcal{S} \cap \mathcal{U} = \varnothing$, such that the class labels are known for the sequences in $\mathcal{S}$ (supervised subset), while $\mathcal{U}$ is unlabeled (unsupervised subset). At first, we train c class-specific π-ESNs, say[2] $\varphi_1, \ldots, \varphi_c$, using the supervised training algorithm presented in the previous section. The generic ith model φ_i is trained using all and only the fraction of data in $\mathcal{S}$ that belongs to class ω_i. Our aim is then twofold: (i) exploit the information underlying $\mathcal{U}$ in order to improve $\varphi_1, \ldots, \varphi_c$; (ii) if not all the sequences in $\mathcal{U}$ can be explained in a probabilistically sound way by $\varphi_1, \ldots, \varphi_c$, then a more suitable, increased value is determined for c, new emotional "classes" are implicitly generated accordingly, along with the corresponding π-ESNs. This is reminiscent of the cross-validated likelihood strategy we used in Algorithm 1, where the number of clusters was increased iteratively as long as the resulting model yielded a higher validation likelihood. In Algorithm 2, at each iteration we increase c and we generate and train a new π-ESN. Moreover, the previous π-ESNs are re-trained over refined class-specific datasets. To this end, only a sub-sample $\mathcal{V} = \{\mathcal{Y}_1, \ldots, \mathcal{Y}_k\}$ of k sequences drawn at random from $\mathcal{U}$ is used at each iteration. There are several reasons for this sub-sampling. Its main rationale is computational, since in real-world scenarios $|\mathcal{U}|$ may be huge, preventing us from using the whole $\mathcal{U}$ at each iteration. Furthermore, working on $\mathcal{V}$ instead of $\mathcal{U}$ is intended as representative of situations where an initial unsupervised dataset $\mathcal{V}$ is collected from a real-life source $\mathcal{U}$ (e.g., the Web) and is used for developing a first version of the classifier, which will be improved at a later step once another set $\mathcal{V}$ of data has been collected from $\mathcal{U}$, and so on. Moreover, drawing $\mathcal{V}$ from $\mathcal{U}$ reminds us of the random initialization of "experts" in mixtures or multiple models, where the expert (i.e., the π-ESN at hand) begins to specialize on a given, random region of its input domain (in fact, using the whole $\mathcal{U}$ would result in a useless model spread across all classes in a roughly uniform manner). Finally, although the in-depth investigation of the topic is beyond the scope of the paper, the random sub-(re)sampling of $\mathcal{U}$ with partial re-insertion (see the final part of Algorithm 2) is expected, to put it in qualitative terms, to increase the statistical robustness of the model in the spirit of bootstrapping with replacement [21].

The core of the algorithm is the application of the reject option. Once trained, the π-ESNs are applied to the next unlabeled sequence $\mathcal{Y}_j \in \mathcal{V}$ obtaining the outputs $\varphi_1(\mathcal{Y}_j), \ldots, \varphi_c(\mathcal{Y}_j)$, which are used in Bayes theorem to obtain an estimate of the corresponding class-posterior probabilities (variables $y_1, \ldots, y_c$ in Algorithm 2). Let φ_i be the *winner* π-ESN according to Bayes decision rule, that is $y_i = \max_{m=1,\ldots,c} y_m$. If the highest class-posterior probability is s.t. $y_i \leq \theta$ (the rejection threshold), then it is seen that $\mathcal{Y}_j$ is unlikely to be drawn from any of the probability distributions modeled by the π-ESNs, and $\mathcal{Y}_j$ is re-inserted in $\mathcal{U}$. If, on the other hand, $y_i > \theta$ (the ith probability distribution explains $\mathcal{Y}_j$ well, and the high output yielded by φ_i expresses confidence in the Bayesian decision) then $\mathcal{Y}_j$ is assigned to the training set for φ_i. In practice, two distinct thresholds are used: θ_{in} fixes the Bayesian confidence required of φ_i in order to let it take responsibility over $\mathcal{Y}_j$, and θ_{out} sets the bar for letting go of $\mathcal{Y}_j$ and re-inserting it in $\mathcal{U}$. Statistics on the distribution of the estimated class-posterior probability values $y_1, \ldots, y_c$ on the training set may be used to find significant values for θ_{in} and θ_{out} (possibly, varying with training time).

Let us spell out the pseudo-code in more detail. First, in addition to c, $\mathcal{S}$, and $\mathcal{U}$, Algorithm 2 takes in input the maximum number of allowed iterations T (which implicitly entails an upperbound on the maximum value for c), the number k of sequences to be inserted in $\mathcal{V}$, and the real-valued rejection thresholds θ_{in} and θ_{out}. First, the algorithm splits $\mathcal{S}$ into c class-specific sub-samples $\mathcal{S}_1, \ldots, \mathcal{S}_c$. The body of the algorithm is iterated until all the sequences in $\mathcal{U}$ have been probabilistically accounted for ($\mathcal{U}$ is emptied, and $k = 0$), unless T is reached (causing a sub-optimal termination where the whole information in $\mathcal{U}$ has not been exploited). Next, $\mathcal{V}$ is drawn at random from $\mathcal{U}$. For $i = 1, \ldots, c$, the class-priors P_i are computed using the usual frequentist approach and φ_i is trained on $\mathcal{S}_i$ using the learning rules outlined in Section 2 (referred to as routine *Regular-Training*$(\varphi_i, \mathcal{S}_i)$ in the pseudo-code). The set $\mathcal{X}$ of unlabeled sequences that is possibly used as the training set for the next, newly created π-ESN is initialized as $\mathcal{X} = \varnothing$. In the following loop, the class-posteriors of the sequences in $\mathcal{V}$ are computed via Bayes theorem (relying on the π-ESNs trained so far, and on the estimated class-priors). Sequences whose class-posteriors y_i are below the rejection threshold θ_{in} (for all the classes) are set into $\mathcal{X}$. The remaining sequences, instead, are assigned (one by one) to the training set $\mathcal{S}_{\text{winner}}$ of the corresponding *winner* class (the class with the highest class-posterior probability). Finally, if $\mathcal{X} \neq \varnothing$, c is increased and a new model is generated. This is accomplished by creating the new training set $\mathcal{S}_c = \mathcal{X}$ for the new "class" c, re-estimating the class-priors, and training the new π-ESN φ_c as usual. Eventually, for $i = 1, \ldots, c$ and for all the sequences in $\mathcal{S}_i$ (in the pseudo-code the loop is on the items of $\mathcal{X}$, having set $\mathcal{X} = \mathcal{S}_i$, since the latter undergoes modifications within the body of the loop itself), the posterior probability of ith class is estimated via Bayes, relying on φ_i. The sequences for which the class-posterior turns out to rest below θ_{out} are removed from $\mathcal{S}_i$ and set back into $\mathcal{U}$ (even if they were originally part of the labeled portion of the dataset). Roughly speaking, the reject option is used again (within a maximum a-posteriori strategy) as a test of probabilistic suitability of the π-ESNs to individual sequences (and, of confidence of the π-ESNs). Note that, to avoid lengthy writing in the pseudo-code, we do not check for the case of emptied training sets, i.e. if $\mathcal{S}_i$ turns up being $\varnothing$ (which would require removal of the ith model and of $\mathcal{S}_i$, and $c \leftarrow c - 1$), but the necessary modifications to the code are fairly straightforward, if sought. At last, the whole procedure is re-iterated over the new, refined values of c, $\mathcal{S}_1, \ldots, \mathcal{S}_c$, and $\mathcal{U}$.

The algorithm may be used either for clustering the data into an adaptive number of variable shape clusters (easing the search for emotional "classes"), or for regular sequence classification as follows. Once training is completed, classification of the sequences in a test set into the newly discovered c states of nature is accomplished, as usual, via the π-ESNs and Bayes decision rule. If the application at hand requires assignment of the sequences to the original classes ω_i, only the corresponding π-ESNs φ_i are used (class-priors $P(\omega_i)$ require re-estimation to account for the implicit removal of the newly generated states of nature). In so doing, although some of the trained π-ESNs are not actively involved in the classification task eventually (which may look like a waste of computational resources), a significant benefit is expected on the resulting classifier from their affecting the training of their active siblings by adaptively re-balancing and re-shaping the corresponding training sets. Furthermore, in the very spirit of Algorithm 2, the scenario may suggest application of the reject option at test time as well, e.g. relying on the rejection threshold θ_{in}, farther reducing the amount of misclassifications. False rejections may be accounted for by adopting some specific performance evaluation criteria, in particular the equal error rate [22].

It is worth stressing two characteristics of Algorithm 2: (1) while in the unsupervised case new clusters are assumed to have a Gaussian distribution in the space of the encodings of the training sequences, in this case the π-ESNs may model complex distributions of the data (including non-convex, non-connected regions); (2) during the execution of the algorithm, a fraction of the original labeled sequences (as defined by $\mathcal{S}$) are moved to $\mathcal{U}$, i.e. they are treated as unlabeled data. In other words, the knowledge of the human experts who defined the minimal set of classes and the corresponding labeling of $\mathcal{S}$ is possibly questioned by the machine. This may play a significant role in those classification tasks, such as emotion recognition, that involve a fuzzy, overlapping, and partially subjective notion of the classes.

[2] Technically, $\varphi_i(.)$ is the function computed by ith π-ESN.

Please cite this article as: E. Trentin et al., Emotion recognition from speech signals via a probabilistic echo-state network, Pattern Recognition Letters (2014), http://dx.doi.org/10.1016/j.patrec.2014.10.015

Table 1
Static classifiers: Average recognition accuracy of female speech sequences from the WaSeP© dataset.

Method	Average accuracy (%)
Nearest neighbor	33.90
Multilayer perceptron	39.32
AdaBoost	45.87
SVM	48.01
π-ESN	86.39

4. Experimental evaluation

The experiments are accomplished on the pseudo-words of the "Corpus of spoken words for studies of auditory speech and emotional prosody processing" (WaSeP©) [19]. This subset of WaSeP© consists of 222 phonetically balanced words, repeatedly uttered by actors (male and female) in six different emotional prosodies: neutral, joy, sadness, anger, fear, and disgust. The average duration of the signals ranges from 0.75 s ("neutral" prosody) to 1.70 s ("disgust"). The outcome of a perception test conducted on these data involving 74 listeners was an average emotion recognition accuracy of 78.53% [23]. It was also observed that the most confused emotion is "disgust". A set of 21 RASTA-PLP acoustic features was extracted [5].

Section 4.1 presents a first experimental round[3] involving only gender-dependent (namely, female) signals. An adequate supervised π-ESN model (architecture and training parameters) is selected which suits the nature of the present data. It is compared w.r.t. statistical pattern recognizers and neural networks. Since these paradigms do not behave like dynamic systems, they are referred to as static classifiers. Section 4.2 discusses a second experimental round involving the complete collection of pseudo-words in the dataset WaSeP©. The model selected in the previous round is applied, and compared to dynamic classifiers (namely, recurrent neural networks and hidden Markov models). Then, the extensions of π-ESN to clustering and SSL are evaluated in Section 4.3, relying on the same data, and Section 4.4 provides a discussion of critical findings.

4.1. Model selection and comparison with static classifiers

In this first round, a subset of WaSeP© consisting of 1386 variable-length sequences extracted from the female speech signals was considered (231 sequences per each class). Evaluation of the classifiers was accomplished relying on the recognition rate averaged over a 10-fold crossvalidation procedure. Each fold was defined by splitting the overall dataset, at random, into a training set (1254 sequences) and a test set (132 sequences). The folds were created such that (i) the 10 test sets did not overlap with each other, and (ii) a uniform prior distribution of individual classes was granted (namely, 22 sequences per class in each test set).

Table 1 reports the average recognition accuracies obtained with π-ESN, and with the traditional techniques. All the static classifiers were trained at the acoustic frame level (i.e., one feature vector at a time). Classification on test set was accomplished by averaging over the class-specific scores yielded by the static classifier along the whole observation sequence. Six class-specific π-ESNs were trained independently over the sequences of the corresponding class. The π-ESNs were initialized as follows[4]. Fed by the 21 input units, the reservoir consisted of 100 state neurons (with transfer function tanh). Its random topology comprised a 10% fraction of all the possible unit-to-unit connections. The RBF-like network features $n = 3$ kernels, having mean, covariances, and mixing parameters initialized as follows. The components of the mean vectors were initialized at random (uniformly) over the range $\mathcal{I} = (-0.5, 0.5)$; the components of the diagonal covariance matrices were initialized to a fixed value, namely $\sqrt{|\mathcal{I}|/n}$; the mixing parameters were initialized at random over the interval (0.0, 1.0) such that they sum to 1. Training of the π-ESNs was accomplished for 20 epochs, using different, quantity-specific learning rates for the mixing parameters ($\eta_\gamma = 1.0e-06$), the means of the Gaussians ($\eta_\mu = 1.0e-10$), and the corresponding variances ($\eta_\sigma = 1.0e-11$). Results confirm the approach is effective. A concise discussion is given in Section 4.4. Table 2 shows the confusion matrix yielded by the π-ESN. Each entry in the matrix is expressed in terms of fraction (%) of test sequences (averaged over the 10 folds). It is seen that "disgust" is the most confusable class. This is in line with the aforementioned results of the human perception test. In this case, it is confused with "neutral" most of the times.

[3] The results reported in Section 4.1 were originally presented in [8].

[4] Unless otherwise stated, in the following the architectures and learning parameters are selected relying on the cross-validated likelihood criterion.

Table 2
π-ESN: Average confusion matrix (%) on female speech signals.

	Neutral	Joy	Sadness	Anger	Fear	Disgust
neutral	99.55	0.45	0.00	0.00	0.00	0.00
joy	2.27	85.00	0.91	0.00	11.36	0.45
sadness	2.27	0.00	97.27	0.45	0.00	0.00
anger	0.00	5.00	1.36	85.91	1.36	6.36
fear	0.00	2.73	7.27	0.91	89.09	0.00
disgust	67.27	4.09	2.73	0.45	9.09	16.36

Table 3
Dynamic classifiers: Average recognition accuracy ($\pm$ standard deviation) of sequences from the whole WaSeP© dataset.

Method	Average accuracy (%)
RNN	66.80 ± 5.44
HMM	79.11 ± 3.91
π-ESN (average rnd. initialization)	87.70 ± 8.62
π-ESN (best rnd. initialization)	96.69 ± 4.67

4.2. Comparison with dynamic classifiers on the whole WaSeP© corpus

In the second experimental round we evaluated the π-ESN on the complete collection of pseudo-words from the dataset WaSeP© (female plus male speech), and we compared the results with sequence-oriented recognizers, namely the traditional RNN by Elman [24], and a continuous-density HMM [25]. The dataset consists of 4714 sequences (1868 female and 2845 male signals) whose length ranges heavily between 33 and 324. The number of sequences per class ranges between 736 (*fear*) and 887 (*sadness*). Again, 10-fold cross-validation was used as in the previous experiment. Individual folds were defined by splitting the data into a training sample of 4243 sequences, and a test sample of 471 sequences. The average recognition accuracies ($\pm$ standard deviations over the 10-fold) are reported in Table 3.

Elman RNN was trained via backpropagation through time [26]. An architecture with 24 hidden (and, context) units was selected, with sigmoid outputs having 0/1 Widrow-Hoff-like targets [7]. If $\mathbf{x}_1, \ldots, \mathbf{x}_T$ denotes a generic input sequence, the most effective classification strategy was to interpret the ith RNN output y_{it} at time t as an estimate of the posterior probability for ith emotional class ω_i, i.e. $y_{it} \approx P(\omega_i|\mathbf{x}_1, \ldots, \mathbf{x}_t)$, and to apply the usual maximum-a-posteriori decision rule at time T.

The HMM was a left-to-right topology with mixtures of eight Gaussian emission probability densities. It was initialized via segmental k-means, trained via forward-backward, and tested via Viterbi [25]. HMMs are known to realize reliable acoustic models of the speech signals in speech recognition, hence they are expected to suit the present scenario well. In fact, in the study by Lee et al. [27] the HMMs

Please cite this article as: E. Trentin et al., Emotion recognition from speech signals via a probabilistic echo-state network, Pattern Recognition Letters (2014), http://dx.doi.org/10.1016/j.patrec.2014.10.015

Table 4
Best π-ESN: Average confusion matrix (%) on the whole dataset.

	Neutral	Joy	Sadness	Anger	Fear	Disgust
Neutral	89.33	0.00	0.00	0.00	9.90	0.77
Joy	0.00	98.85	0.90	0.13	0.13	0.00
Sadness	0.00	2.30	97.32	0.00	0.26	0.13
Anger	4.46	0.13	0.77	94.52	0.00	0.13
Fear	0.00	0.38	0.00	0.00	99.62	0.00
Disgust	0.00	0.00	0.77	1.02	0.00	98.21

reached an appreciable accuracy of about 76% (with four emotional classes). We used six emotion-specific HMMs. Model selection led to four-state HMM topologies. The average recognition rate we obtained (79%) is basically in line with the evidence observed in [27], as well as with the human perception tests.

The same π-ESN architecture and training parameters selected in the previous section were used. The third row of Table 3 reports the average accuracy for six different random initializations of the RBF parameters (and the 10-folds) using automatic tuning of the learning rates, while the last row shows the 10-fold average accuracy yielded by the best π-ESN in the lot. As usual, random initializations affect the gradient ascent. Nonetheless, the results are surprisingly high, and the relative improvement over the HMMs (and, over the humans) is dramatic. These findings require a discussion, which is handed out in Section 4.4. Table 4 reports the 10-fold average confusion matrix for the best π-ESN over the complete dataset WaSeP©.

4.3. Evaluation of the unsupervised and semi-supervised extensions of the π-ESN

Using the notation introduced in Section 3, we set $\mathcal{U}$ as the unlabeled collection of all the sequences of acoustic features corresponding to the pseudo-words of WaSeP©. The dataset was split into a training set $\mathcal{T}$ and a test set $\mathcal{V}$ (now playing the role of validation set) over 10-folds as in Section 4.2. Algorithm 1 was applied first, obtaining an average $k_{\max} = 21 \pm 3$. Observing the trend of $p(\mathcal{V}|\pi\text{-ESN})$ (the cross-validated likelihood) and $p(\mathcal{T}|\pi\text{-ESN})$ as functions of k, it turned out that for $k > k_{\max}$ the model overfitted the training data. The cluster size ranged from 73 to 384. Note that the value of $k_{\max}$ is in the order of the overall number of kernels (18) involved in the optimal set of π-ESNs used in the supervised setup. It is also (roughly) in the order of the number of emotions on the so-called "wheel of emotions" by Plutchik [28], although there is clearly not enough evidence yet to interpret the obtained clusters as characteristic of specific classes of emotions. Direct evaluation of the result of the clustering process may be achieved via a probabilistic adaptation pDB of the Davies-Bouldin index (DB) [29], which was shown by Günter and Bunke [30] to be robust, size-independent, and consistent with other prominent indexes. Let us assume that each cluster $\mathcal{C}_i$ is associated with a probabilistic model $p_i(\mathcal{Y})$ of the data (e.g., in the present setup we have $p_i(\mathcal{Y}) = K_i(\mathbf{x})$ where $\mathbf{x} = \phi(\mathcal{Y})$), and let ξ_i denote the centroid of $\mathcal{C}_i$ (here we set $\xi_i = \mu_i$). The pDB is defined as $\text{pDB} = \sum_{i=1}^{k_{\max}} \max_{j \neq i}(\delta_{ij})$, having let $\delta_{ij} = (\zeta_i + \zeta_j)/(\zeta_{ij} + \zeta_{ji})$ where: for a generic i, ζ_i is the average absolute likelihood difference $E[|p_i(\mathcal{Y}_l) - p_i(\xi_i)|]$, averaged over the encodings $\mathbf{x}_l = \phi(\mathcal{Y}_l)$ of all the sequences $\mathcal{Y}_l \in \mathcal{C}_i$, and ζ_{ij} (for generic i, j) is given by $\zeta_{ij} = |p_i(\xi_i) - p_i(\xi_j)|$ (it basically expresses the absolute difference between the likelihoods of model j and of model i given the latter). Clearly $\text{pDB} \in (0, +\infty)$, and small pDB values are expected of good partitions of the data (i.e., with high intra-cluster coherence and high inter-cluster separation). We obtained an average pDB of 0.31 ± 0.07, that is consistent with values of DB considered small [30]. The result was compared with an established sequence clustering technique, the segmental k-means algorithm [25], using $k_{\max}$ HMMs with Gaussian emission probabilities as the cluster prototypes ξ_i, and the usual Viterbi alignment on the trellis for computing the likelihood $p_i(\mathcal{Y}_l)$ of lth input sequence in the ith cluster. Exploiting the generative capabilities of HMMs [25], quantities $p_i(\xi_j)$ for pDB are computed generating an ML sequence $\hat{\mathcal{Y}}_j$ of the mean vectors of the Gaussian emission probabilities according to the left-to-right topology of jth HMM, and evaluating the likelihood of ith HMM given $\hat{\mathcal{Y}}_j$ via Viterbi alignment. The average pDB turned out to be 0.71 ± 0.13 (roughly doubling the pDB for the π-ESN). Finally, we surveyed the possible correlation between the clusters and the class labels of the corresponding sequences for one of the 10-fold partitioning of the dataset. We observed that in six clusters (out of 21) a fraction of at least 80% of the data belonged to a single class. A fraction of 70% was observed in other five clusters. Only in two cases the presence of all six classes was observed, and in one of them the relative frequency of the classes was close to uniform. A significant fraction of 69% of the data assigned to the largest cluster (341 sequences overall) belonged either to the *disgust* or *neutral* classes. The overlap between these classes appears to be related, to some extent, to the results reported in Table 2. Section 4.4 elaborates briefly on some of these results.

As for the SSL version of the π-ESN, we tested Algorithm 2 using the complete dataset WaSeP©, split into training and test sets according to the same 10-fold crossvalidation setup used in the supervised case (section 4.2). The maximum a-posteriori criterion used in Algorithm 2 may lead to sound partitions of the data only if the π-ESNs $\varphi_1, \ldots, \varphi_c$ used therein yield reliable estimates of the class-conditional pdfs they are expected to model. This is guaranteed by the classification results reported in Section 4.2. Quantitative evaluations of Algorithm 2 were achieved as follows. Each training set (in the 10-fold) was partitioned at random (by sampling uniformly over the different classes) into a labeled subset $\mathcal{S}$ (20% of the sequences) and an unlabeled subset $\mathcal{U}$ (the remaining 80%), and we let $k = \lfloor |\mathcal{U}|/20 \rfloor$. Conservative, model-specific values of the rejection thresholds for the generic π-ESN φ_i were statistically estimated, at each iteration, as $\theta_{\text{in}} = E[y_i] - \frac{1}{8}\sigma[y_i]$ and $\theta_{\text{out}} = E[y_i] - \frac{1}{2}\sigma[y_i]$ (where y_i is defined as in Algorithm 2), respectively. Setting the other arguments of Algorithm 2 to $c = 6$ and $T = 50$, using the same π-ESNs architecture and hyperparameters as in Section 4.2 (with automatic tuning of the learning rates), and applying 50 epochs of *Regular-Training*, resulted in termination after 23 ± 9 iterations on average, yielding 18 ± 2 new models ("classes", or clusters), s.t. $c = 24 \pm 2$. For each fold, the pDB index was computed for the resulting partition of the training set $\mathcal{S} \cup \mathcal{U}$. The quantities $p_i(\xi_j)$ involved in the calculation of pDB were computed as $p_i(\xi_j) = K_{im}(\mu_{im})$, where $K_{im}(\mu_{im})$ denotes the mth Gaussian component of the mixture model in φ_i evaluated over the corresponding mean vector, and m is the index of the largest quantity $c_{ih}K_{ih}(\mu_{ih})$ in the mixture, where c_{ih} is the hth mixing parameter in the ith mixture (thus, an ML centroid is used). The average pDB turned out to be 0.66 ± 0.03, which indicates coherent yet well-separated clusters. Moreover, the pDB compares favorably with the segmental k-means, and lies in the same fine range of the previous techniques. Next, the emotion recognition performance was evaluated. First, the average test accuracy yielded by π-ESNs trained as in Section 2 (with automatic tuning of the learning rates, and 50 training epochs) on the sole labeled subset $\mathcal{S}$ of the fold-specific training data was $81.95 \pm 5.83\%$. The gap between this result and the accuracies we observed in Section 4.2 is due to the limited amount of labeled data used herein. In spite of that, the resulting accuracy is still higher than the recognition rates yielded by traditional classifiers trained on the whole training set $\mathcal{S} \cup \mathcal{U}$. What is really relevant to observe is the effectiveness of the proposed SSL strategy: when applying the models φ_i returned by Algorithm 2 the average accuracy raised to $87.51 \pm 9.58\%$, which compares impressively with the fully supervised π-ESNs having random initialization (third row of Table 3). All the more, this accuracy represents a substantial average improvement (30.86% relative error rate reduction, that is 6.80% relative accuracy gain) of the classifier performance over its initial, supervised baseline.

Please cite this article as: E. Trentin et al., Emotion recognition from speech signals via a probabilistic echo-state network, Pattern Recognition Letters (2014), http://dx.doi.org/10.1016/j.patrec.2014.10.015

4.4. Discussion

Besides the nice general behavior of the different variants of the π-ESN, there are at least four key points in the experimental results deserving a concise discussion.

(1) The absolute recognition accuracy achieved by the π-ESN on the whole dataset WaSeP© appears to be even too high, especially if compared with the established approaches. This is due to the concomitant of several reasons. First of all, clearly the π-ESN is a good fit to the nature of the sequences in WaSeP©. Since the speech signals are pseudo-words, segmentation of the acoustic observation sequences into their phonetic units (as in standard HMMs) is not necessary, if not even misleading. On the contrary, the acoustic characteristics of diverse emotions are roughly stretched over the whole sequence. We argue that the squeezing (the encoding) of sequences onto fixed-dimensional static patterns realized by the reservoir of the π-ESN happens to capture global prosodic features that are statistically representative of the emotion underlying the entire sequence, possibly dropping finer (yet, heavier) information that may mislead other dynamic classifiers. Finally, it shall not be neglected that the sequences in WaSeP© turned out accidentally to be particularly π-ESN-friendly, as well.

(2) The result achieved is significantly higher than the average accuracy yielded by humans on similar tasks. To the end of finding a rationale behind this phenomenon, the following argument appears to be sound. While the system was trained on data having identical nature to the test data (including the criteria applied for assigning specific class labels), humans do not undergo any data-specific training. They assign test utterances to an emotional class according to generic, prior knowledge of their (subjective) concept of specific emotions (from the humans' point of view, no hard distinctions can even be made between certain emotions). Furthermore, the audio recordings in the dataset are not real-world utterances (and do not express real emotions), since they were performed by actors. It is likely that this introduces a significant bias, such that humans cannot easily recognize the emotion from the acted expression. On the other hand, the machine learns from the training sample how actors tend to give a certain interpretation of a specific emotion (e.g., affecting some signature features) that, later, can be easily recognized in the test data. It is worth mentioning that in the aforementioned study by Lee et al. [27] an accuracy of 76.1% via HMMs is reported, and compared with a 68.3% accuracy observed in a human perception test on the same signals. In summary, in this setup the machines are positively biased by the dry coherence between training and test conditions.

(3) A significant (yet, counterintuitive) relative improvement in terms of the π-ESN accuracy is observed when moving from a speaker-dependent task (Section 4.1) to a speaker-independent setup (Section 4.2). This phenomenon is puzzling if observed from the speech community viewpoint, since speaker-dependent (or, gender-dependent) speech recognizers are much simpler to develop, and more accurate (as long as the user is the same speaker or belongs to the same gender, respectively) [31]. We argue that the main rationale still revolves around the argument we put forward in point 1, i.e. the π-ESN encoding of the acoustic sequences results to be disburdened of local phonetic features, and capable of expressing global and relevant emotion-related attributes of prosodic nature. Therefore, using the whole WaSeP© collection instead of the female signals only means roughly doubling the amount of emotion-related information available for training the π-ESNs (increasing the robustness of the estimated probabilistic quantities to a significant extent) without introducing any appreciable phonetic-related complexity. Finally, it is worth stressing that the first experimental round involved only 22 test sequences per fold. This is a fairly significant 4.6% fluctuation of the corresponding fold-specific accuracy as a consequence (for instance) of a single misclassification.

(4) There is a certain correlation between the clusters discovered by the unsupervised version of the π-ESN and the classes of emotions as expressed by the labels. Albeit sought, the phenomenon could not be taken for granted. On the contrary, it may even appear to be astonishing: why on earth should a clustering algorithm partition a bunch of acoustic data on the basis of the corresponding emotion instead of (say) their phonetic attributes, or the characteristics of the speaker's voice (e.g., *female* and *male* clusters)? Singularly enough, elaborating on this unsupervised-natured issue ends up providing us with the confirmation of the arguments we outlined to explain the previous, supervised-natured phenomena. Once again, the prosody-related features encapsulated within the π-ESN encodings of the speech signals seem to be utterly representative of the overall mood (i.e., emotional state) of the utterances. Thence, Algorithm 1 discovers prosodic clusters which are clearly related to specific (not necessarily pre-defined) classes of emotions.

5. Conclusions

The three variants of π-ESN were effective on WaSeP©, proving the viability of the paradigm for emotion modeling and recognition from speech signals. SSL of π-ESNs for emotion recognition and clustering over large-scale collections of partially labeled, spontaneous speech corpora is a promising research direction. Applications of π-ESNs are expected in other domains involving multivariate sequential data, as well.

References

[1] B. Vlasenko, B. Schuller, A. Wendemuth, G. Rigoll, Frame vs. turn-level: Emotion recognition from speech considering static and dynamic processing, in: Proceedings of the 2nd International Conference on Affective Computing and Intelligent Interaction (ACII'07), Springer, 2007, pp. 139–147.
[2] J. Wagner, T. Vogt, E. André, A systematic comparison of different HMM designs for emotion recognition from acted and spontaneous speech, in: Proceedings of the 2nd International Conference on Affective Computing and Intelligent Interaction (ACII'07), Springer, 2007, pp. 114–125.
[3] S.B. Davis, P. Mermelstein, Comparison of parametric representations of monosyllabic word recognition in continuously spoken sentences, IEEE Trans. Acoust. Speech Signal Process. 28 (4) (1980) 357–366.
[4] F. Schwenker, S. Scherer, Y. Magdi, G. Palm, The gmm-svm supervector approach for the recognition of the emotional status from speech, in: ICANN'09 Part I, LNCS, vol. 5768, Springer, 2009, pp. 894–903.
[5] H. Hermansky, N. Morgan, A. Bayya, P. Kohn, Rasta-plp speech analysis technique, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-92, vol. 1, IEEE Service Center, Piscataway, NJ, USA, 1992, pp. 121–124.
[6] F. Dellaert, T. Polzin, A. Waibel, Recognizing emotion in speech, in: Proceedings of ICSLP, 1996, pp. 1970–1973.
[7] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[8] E. Trentin, S. Scherer, F. Schwenker, Maximum echo-state likelihood networks for emotion recognition, in: Proceedings of Artificial Neural Networks in Pattern Recognition (ANNPR'10), Springer, 2010, pp. 61–72.
[9] H. Jaeger, H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science 304 (2004) 78–80.
[10] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, 1995.
[11] S. Scherer, M. Oubbati, F. Schwenker, G. Palm, Real-time emotion recognition from speech using echo state networks, in: Proceedings of ANNPR, Springer, 2008, pp. 205–216.
[12] S. Scherer, F. Schwenker, W.N. Campbell, G. Palm, Multimodal laughter detection in natural discourses, in: Proceedings of 3rd International Workshop on Human-Centered Robotic Systems (HCRS'09), Springer-Verlag, Berlin, Germany, 2009.
[13] H. Jaeger, Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach, Technical Report 159, Fraunhofer-Gesellschaft, St. Augustin Germany, 2002.
[14] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Comput. 14 (11) (2002) 2531–2560.

Please cite this article as: E. Trentin et al., Emotion recognition from speech signals via a probabilistic echo-state network, Pattern Recognition Letters (2014), http://dx.doi.org/10.1016/j.patrec.2014.10.015

[15] H. Jaeger, Short term memory in echo state networks, GMD-Report 152, GMD—German National Research Institute for Computer Science, 2002.
[16] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Netw. 5 (2) (1994) 157–166.
[17] F. Schwenker, E. Trentin, Pattern classification and clustering: A review of partially supervised learning approaches. Pattern Recogn. Lett. 37 (2014) 4–14.
[18] R. Rust, D. Schmittlein, A bayesian cross-validated likelihood method for comparing alternative specifications of quantitative models. Market. Sci. 4 (1) (1985) 20–40.
[19] B. Wendt, H. Scheich, The "magdeburger prosodie korpus"—A spoken language corpus for fmri-studies, in: Speech Prosody 2002, SProSIG, Aix-en-Provence, France, 2002.
[20] G. McLachlan, K. Basford (Eds.), Mixture Models: Inference and Applications to Clustering, Marcel Dekker, New York, USA, 1988.
[21] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, New York, 1993.
[22] K.P. Li, J.E. Porter, Normalizations and selection of speech segments for speaker recognition scoring, in: Proceedings of ICASSP'88, IEEE Service Center, Piscataway, NJ, USA, 1988, p. 595.
[23] K.R. Scherer, T. Johnstone, G. Klasmeyer, Vocal expression of emotion, in: R.J. Davidson, K.R. Scherer, H.H. Goldsmith (Eds.), Handbook of Affective Sciences, Affective Science, Oxford University Press, 2003, pp. 433–456.
[24] J.L. Elman, Finding structure in time. Cogn. Sci. 14 (2) (1990) 179–211.
[25] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE 77 (2) (1989) 257–286.
[26] P.J. Werbos, Generalization of backpropagation with application to a recurrent gas market model. Neural Netw. 1 (4) (1988) 339–356.
[27] C.M. Lee, S. Yildirim, M. Bulut, A. Kazemzadeh, C. Busso, Z. Deng, S. Lee, S.S. Narayanan, Emotion recognition based on phoneme classes, in: Proceedings of ICSLP'04, ISCA, Jeju Island, Korea, 2004.
[28] R. Plutchik, The nature of emotions. Amer. Scient. 89 (4) (2001) 344–354.
[29] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. 1 (2) (1979) 224–227.
[30] S. Günter, H. Bunke, Validation indices for graph clustering, Pattern Recogn. Lett. 24 (8) (2003) 1107–1113.
[31] X.D. Huang, Y. Ariki, M. Jack, Hidden Markov Models for Speech Recognition, Edinburgh University Press, Edinburgh, 1990.

Please cite this article as: E. Trentin et al., Emotion recognition from speech signals via a probabilistic echo-state network, Pattern Recognition Letters (2014), http://dx.doi.org/10.1016/j.patrec.2014.10.015