On combining classifiers for speaker authentication ⋆

Leandro Rodríguez-Liñares [a] Carmen García-Mateo [b]
José Luis Alba-Castro [b]

[a] *E.T.S.E Informática, Universidade de Vigo, 32004 - Ourense (Spain)*
[b] *E.T.S.E Telecomunicación, Universidade de Vigo, 36200 - Vigo (Spain)*

Abstract

Speaker Verification and Utterance Verification are examples of techniques that can be used for Speaker Authentication purposes.

Speaker Verification consists of accepting or rejecting the claimed identity of a speaker by processing samples of his/her voice. Usually, these systems are based on HMM models that try to represent the characteristics of the speakers' vocal tracts.

Utterance Verification systems make use of a set of speaker-independent speech models to recognize a certain utterance. If the utterances consist of passwords, this can be used for identity verification purposes.

Up to now, both techniques have been used separately. This paper is focused on the problem of how to combine these two sources of information. New architectures are presented to join an utterance verification system and a speaker verification system in order to improve the performance in a speaker verification task.

Key words: Speaker authentication, Speaker verification, Utterance authentication, Gaussian mixture models, Verbal information verification, Neural networks, Classifier combination.

1 Introduction

Speaker authentication is a process by which an output hypothesis produced by a statistical classifier is verified to determine wether the input speech belongs to the claimed speaker or not [1,2].

⋆ This work has been partially supported by Spanish CICYT under the project 1FD97-0077-C02-01.

Scanning the literature, two generic approaches for designing a speaker authentication system can be found. One is based on verifying the actual content of the speech as a means of verifying the speaker identity [3–5]. This can be considered as the classical "password-based" approach and implies that the speech content must be kept secret and attached to the speaker. The core of the system is a speech recognizer tailored for this particular purpose. We call these systems "Utterance verifiers" (see figure 1).

The other is based on verifying the speaker voice by itself as a way of verifying the identity. The latter approach is by nature speech content independent [6,7], although in some practical systems is restricted to be text-dependent [8,9]. Although these systems are not speech recognizers, they usually share most of their technology with speech recognizers, i.e., most of them are based on a HMM (Hidden Markov Model) formulation with a front-end based on a cepstral analysis. We call these systems "Speaker verifiers".

As stated before, speaker verification systems make use of the physical characteristics of the speaker's vocal tract, while utterance verification systems make use of speech content. A certain degree of uncorrelation may be then assumed, making it possible to build speaker authentication systems that simultaneously exploit both levels of information in order to obtain more reliable and robust recognition.

The approach of combining information from multiple sources has been used in different application fields of classification like speech recognition [10,11] and handwriting recognition [12]. In [13] a method is proposed to combine two sets of speaker-dependent phoneme models for prompted-text speaker recognition. A new set of models is obtained using a linear combination and the combined system obtains better results than any of the individual systems. Since this paper is not focussed on combination, this topic is briefly presented and not thoroughly explored, although it is clearly stated that combining different speaker authentication systems is an interesting issue to be investigated in greater depth.

The motivation for exploring the combination issue is to improve performance. This can make possible the developing of systems with a level of performance adequate to be deployed in real-world applications. This is especially important if the aim is to transfer the technology from the research laboratories to the industry.

In [14], Kitter explores the topic of combining two or more classification systems. A common theoretical framework for classifier combination is developed. Different assumptions and approximations are made to derive commonly used classifier combination schemes such as the product rule, sum rule, min rule and max rule.

In this article, Kitter's theoretical framework was used to design our set of experiments with these combination rules. Moreover, new methods based on a neural network scheme were added. In [14], each classifier uses its own representation of the pattern to classify. This means a serious overhead compared to individual systems. In this paper, the starting point of the proposed combination procedures is to assume that the components for the combined system share most of their technology thus avoiding this overhead. Besides, the theoretical scenario is changed from the one proposed by Kitter, since in our case the classifiers share the representation of the patterns to classify. This implies that major changes in the theoretical frame must be made.

The organization of the rest of the paper is as follows. In the next section the characteristics of each individual verification system are described. Section 3 is devoted to the different combination techniques that can be applied when developing a common theoretical framework for classifier combination. In section 4, the experimental framework is described: databases, speech parametrization, and experiments definition. The results of the experiments for the individual classifiers and the combined systems are shown in section 5. Finally, section 6 summarizes the main results of the paper.

2 Architecture of the verifiers

In this paper the problem of how to combine two different speaker authentication systems is addressed. Before that, the design and development of the individual classifiers, namely "utterance verifier" and "speaker verifier" is required. In the case of the speaker verifier, systems based on GMM's (Gaussian Mixture Models) were used employing the world-model approximation [15,2]. In the utterance verifier, the passwords set was modeled by concatenation of a string of sub-word HMM's [16,3,4].

2.1 Speaker verifier

Since our system is based on passwords, the speaker authentifier as a whole may be considered as text-dependent. However, and for practical reasons, each time the password is changed the speaker verification system should not be affected. The verification process must then be, by nature, text-independent. GMM's, which are a special case of continuous HMM's [7,8] where the number of states is one were opted for. This type of model has proved to be effective in modeling the speaker identity in text-independent speaker recognition applications [7,6,2]. A Hidden Markov Model with several states models, not only the underlying sounds, but also the temporal sequencing among them. How-

ever, in text-independent speaker recognition tasks, the sequencing of sounds found in the training data does not necessarily represent the sound sequences found in the testing data. So, transition probabilities between states have little influence.

A Gaussian Mixture Model λ is characterized by

$$\lambda = \{c_\mathrm{m}, \bar{\mu}_\mathrm{m}, \mathbf{U}_\mathrm{m}\ ,\ 1 \leq m \leq M\}$$

where, for each mixture m:

- c_m, is the mixture's weighting factor.
- $\bar{\mu}_\mathrm{m}$ is the mixture's mean vector.
- $\mathbf{U}_\mathrm{m}$ is the covariance matrix.

The likelihood of a sequence of observation vectors

$$\mathbf{O} = [\ \bar{o}_1\ \bar{o}_2\ \ldots\ \bar{o}_\mathrm{T}\ ] \tag{1}$$

given the model λ will be a combination of gaussian components

$$P(\mathbf{O}/\lambda) = \prod_{\mathrm{t}=1}^{\mathrm{T}} b(\bar{o}_\mathrm{t})$$

where the probability of the observation of the vector $\bar{o}_\mathrm{t}$ is given by

$$b\left(\bar{o}_\mathrm{t}\right) = \sum_{\mathrm{m}=1}^{\mathrm{M}} c_\mathrm{m} \mathcal{N}\left(\bar{o}_\mathrm{t}; \bar{\mu}_\mathrm{m}, \mathbf{U}_\mathrm{m}\right)$$

$\mathcal{N}\left(\bar{o}_\mathrm{t}; \bar{\mu}_\mathrm{m}, \mathbf{U}_\mathrm{m}\right)$ denotes the m^{th} multivariate Gaussian probability density function with mean vector $\bar{\mu}_\mathrm{m}$ and covariance matrix $\mathbf{U}_\mathrm{m}$ and c_m is the mixture weight for the m^{th} mixture component, with the constraint that $\sum_{\mathrm{m}=1}^{\mathrm{M}} c_\mathrm{m} = 1$. M is the total number of mixture components.

In a speaker verification problem, the goal is to determine whether a person is who he or she claims to be. The most straightforward approximation would be to use the log-likelihood $\log(P(\mathbf{O}/\lambda_\mathrm{k}))$ where λ_k is the supposed speaker's model. This is what it is called *unnormalized log-likelihood score*. The speaker is accepted when the unnormalized score $S_{\mathrm{unnor}}(\mathbf{O}, k)$ is above a certain threshold $\varphi_\mathrm{k}^\mathrm{s}$ (s for Speaker):

$$S_{\mathrm{unnor}}(\mathbf{O}, k) = \log(P(\mathbf{O}/\lambda_\mathrm{k}))\ \geq\ \varphi_\mathrm{k}^\mathrm{s} \Rightarrow \text{accepted}$$

One of the most difficult problems for this task is to select the optimal threshold. The threshold must be set up in order to achieve the required balance between rejecting true claimant utterances (false rejection or type I errors) and accepting impostors utterances (false acceptance or type II errors). The unnormalized log-likelihood score usually exhibits a high sensivity to the value of the threshold. This problem is adressed in section 2.3.

2.2 Utterance verifier

As can be observed in figure 1, the use of utterance verification techniques requires a set of previously trained speaker independent models that represent the linguistic units or sub-words that can be present in the utterances. With such a set, valid passwords can be constructed dynamically by concatenation of the required units.

Let us suppose there is a sequence of observation vectors as in equation (1) and we want to verify whether $\mathbf{O}$ corresponds to a certain word or phrase $\mathbf{W}_{\mathrm{k}}$ or not.

The word $\mathbf{W}_{\mathrm{k}}$ can be modelled as a concatenation of J sub-word units

$$\mathbf{W}_{\mathrm{k}} = \left[\; w_{1}^{(\mathrm{k})} \; w_{2}^{(\mathrm{k})} \; \ldots \; w_{\mathrm{J}}^{(\mathrm{k})} \; \right]$$

The sequence of observation vectors $\mathbf{O}$ can be thus divided using Viterbi alignment in J segments

$$\mathbf{O} = [\; \mathbf{O}_{1} \; \mathbf{O}_{2} \; \ldots \; \mathbf{O}_{\mathrm{J}} \;]$$

where $\mathbf{O}_{\mathrm{j}}$ contains i_{j} vectors and is the observation sequence corresponding to the speech sequence for sub-word $w_{\mathrm{j}}^{(\mathrm{k})}$. From now on, the sub-word $w_{\mathrm{j}}^{(\mathrm{k})}$ will be substituted for its correspondent HMM model Λ_{j} without loss of generality.

The sequence of observation vectors $\mathbf{O}$ will be accepted to correspond to a certain word $\mathbf{W}_{\mathrm{k}}$ when the unnormalized score

$$S_{\mathrm{unnor}}(\mathbf{O}, \mathbf{W}_{\mathrm{k}}) = \frac{1}{J} \sum_{\mathrm{j}=1}^{\mathrm{J}} \log \left[P(\mathbf{O}_{\mathrm{j}} / \Lambda_{\mathrm{j}}) \right]$$

is above a certain threshold $\varphi_{\mathrm{k}}^{\mathrm{u}}$ (u stands for Utterance):

$$S_{\mathrm{unnor}}(\mathbf{O}, \mathbf{W}_{\mathrm{k}}) \geq \varphi_{\mathrm{k}}^{\mathrm{u}} \Rightarrow \text{accepted}$$

As in the speaker verification case, utterance verification using $S_{\text{unnor}}(\mathbf{O}, \mathbf{W}_{\text{k}})$ exhibits high sensivity to threshold estimation.

2.3 Normalization

We can reformulate a generic verification problem from the perspective of a statistical hypothesis test. Thus, the *null hypothesis* H_0 can be defined as "a token $\mathbf{T}$ belongs to the claimed set' and the *alternative hypothesis* H_1 as "$\mathbf{T}$ does *not* belong to the claimed set". Given a test token $\mathbf{T}$, our objective is to check the null hypothesis against the alternative hypothesis:

$$P(H_0/\mathbf{T}) \;\geq\; P(H_1/\mathbf{T}) \tag{2}$$

These two probabilities are usually unknown, but, applying Bayes rule, equation (2) is equivalent to:

$$\frac{P(\mathbf{T}/H_0)P(H_0)}{P(\mathbf{T})} \geq \frac{P(\mathbf{T}/H_1)P(H_1)}{P(\mathbf{T})}$$

or to evaluate the likelihood ratio

$$\frac{P(\mathbf{T}/H_0)}{P(\mathbf{T}/H_1)} \tag{3}$$

and compare it against a decision threshold.

2.3.1 Speaker verifier

In the speaker verification problem, $P(\mathbf{T}/H_0)$ is calculated as $P(\mathbf{O}/\lambda_{\text{k}})$. To estimate $P(\mathbf{T}/H_1)$ several different choices can be found. One of them is to train a HMM model (called *anti-model* and denoted $\lambda_{\bar{\text{k}}}$) with observation segments from other speakers but the claimant. There are typically two strategies to build it up: to use a set of B speaker-dependent models called *cohort* or *background models* [17] or to train a *world model* shared by all the speakers.

In the log domain, the ratio in eq. (3) becomes

$$S_{\text{nor}}(\mathbf{O}, k) = \log P(\mathbf{O}/\lambda_{\text{k}}) - \log P(\mathbf{O}/\lambda_{\bar{\text{k}}}) \tag{4}$$

This ratio is called the *normalized log-likelihood score*. This log-likelihood ratio is compared to the threshold $\varphi^{\text{s}}_{\text{k}}$ to accept or reject the claimed speaker.

The decision is thus based on a relative score less dependent on non-speaker utterance variations such as voice quality or speaker's vocal tract variations.

2.3.2 Utterance verifier

Given the fact that the word $\mathbf{W}_\mathrm{k}$ can be modeled by concatenation of a string of sub-word models, the utterance verification can be formulated as a set of independent hypothesis tests, each of them representing a sub-word verification test [18]. The *null hypothesis* H_0 will be "$\mathbf{O}_\mathrm{j}$ corresponds to the sub-word $w_\mathrm{j}^{(\mathrm{k})}$ (the model Λ_j)" and the *alternative hypothesis* H_1 will mean that "$\mathbf{O}_\mathrm{j}$ does not correspond to the sub-word $w_\mathrm{j}^{(\mathrm{k})}$ (the model Λ_j)".

There are several possibilities to obtain the *normalized log-likelihood score*. In this case, this score is calculated as

$$S_{\mathrm{nor}}(\mathbf{O}, \mathbf{W}_\mathrm{k}) = \frac{1}{J} \sum_{\mathrm{j}=1}^{\mathrm{J}} \left[\log \left[P(\mathbf{O}_\mathrm{j}/\Lambda_\mathrm{j}) \right] - \max_{\mathrm{m} \neq \mathrm{j}} \left[\log \left[P(\mathbf{O}_\mathrm{j}/\Lambda_\mathrm{m}) \right] \right] \right] \tag{5}$$

using, for each segment $\mathbf{O}_\mathrm{j}$ the probability of the most likely model except Λ_j. This score is compared to a threshold $\varphi_\mathrm{k}^{\mathrm{u}}$.

3 Combination techniques

In [14], Kitter proposes several strategies for combining two or more classification systems. The theoretical framework is as follows: a decision should be taken if a pattern $\mathcal{Z}$ corresponds to one class m out of M possibilities $(\omega_1, \ldots \omega_\mathrm{M})$. If the number of classifiers is R and each one uses its own representation of the pattern to classify $\mathbf{O}^{(\mathrm{r})}$, the decision to be taken is

$$\text{assign } \mathcal{Z} \longrightarrow \omega_\mathrm{m} \quad \text{if} \quad P\left(\omega_\mathrm{m}/\mathbf{O}^{(1)}, \ldots, \mathbf{O}^{(\mathrm{R})}\right) \geq \zeta_\mathrm{m} \tag{6}$$

where ζ_m is a threshold valid for the class ω_m.

This theoretical framework is, in our case, slightly modified. Given a test utterance from a presumed speaker k, two normalized scores denoted $S_{\mathrm{nor}}(\mathbf{O}, k)$ (see eq. (4)) and $S_{\mathrm{nor}}(\mathbf{O}, \mathbf{W}_\mathrm{k})$ (see eq. (5)) are produced. These scores come from the combination of probabilities, and, from a statistical point of view, they are random variables. They can be combined in a bidimensional random variable:

$$\mathbf{S}_\mathrm{k} = [S_{\mathrm{nor}}(\mathbf{O}, k), S_{\mathrm{nor}}(\mathbf{O}, \mathbf{W}_\mathrm{k})]$$

Speaker authentication can be then expressed as a classification problem where we it has to be decided whether a pattern $\mathcal{Z}$ belongs to speaker k or not. In this case, a binary random variable ω_{k} can be defined:

$$\omega_{\mathrm{k}} = \begin{cases} 0 \Rightarrow \text{impostor} \\ 1 \Rightarrow \text{customer} \end{cases}$$

In fact, $\omega_{\mathrm{k}} = 0$ means that the utterance belongs to a speaker that is not speaker k.

It has previously been stated that $\mathbf{S}_{\mathrm{k}}$ is a bidimensional random variable. However, each time an authentication attempt takes place, what there actually is is a realization of this random variable $\mathbf{s}_{\mathrm{k}}$. Thus, the authentication process can be reformulated based on $\mathbf{s}_{\mathrm{k}}$:

$$\text{assign } \mathcal{Z} \longrightarrow \omega_{\mathrm{k}} \quad if \quad P\left(\omega_{\mathrm{k}} = 1/\mathbf{s}_{\mathrm{k}}\right) \geq \varphi_{\mathrm{k}} \tag{7}$$

where φ_{k} is a threshold calculated for speaker k. The criterion expressed in eq. (6) is simplified: the sets of sequences of parameters $\left(\mathbf{O}^{(1)}, \ldots, \mathbf{O}^{(\mathrm{R})}\right)$ are transformed into a two-element vector $\mathbf{s}_{\mathrm{k}}$.

From now on and for sake of simplicity, the index k will be eliminated. The super-indexes $^{\mathrm{s}}$ for Speaker and $^{\mathrm{u}}$ for Utterance will be used. Therefore:

$$\begin{aligned} S^{\mathrm{s}} &= S_{\mathrm{nor}}(\mathbf{O}, k) \\ S^{\mathrm{u}} &= S_{\mathrm{nor}}(\mathbf{O}, \mathbf{W}_{\mathrm{k}}) \\ \mathbf{s} &= [s^{\mathrm{s}}, s^{\mathrm{u}}] \end{aligned}$$

The rule of eq. (7) is optimal from a theoretical point of view. However, it does not lead to a practical test, since it does not say how to compute the probability $P\left(\omega_{\mathrm{k}} = 1/\mathbf{s}_{\mathrm{k}}\right)$. Thus, to implement this rule, several assumptions must be made in order to derive an approximation function $f(\cdot, \cdot)$:

$$\begin{aligned} f\left(s^{\mathrm{s}}, s^{\mathrm{u}}\right) &\approx P\left(\omega = 1/\left[s^{\mathrm{s}}, s^{\mathrm{u}}\right]\right) \\ f\left(s^{\mathrm{s}}, s^{\mathrm{u}}\right) &\geq \varphi \Rightarrow \text{accepted} \end{aligned} \tag{8}$$

There are many possible approaches to formulate the approximation function $f(\cdot, \cdot)$, and each of them will lead to a different combination rule. Two of these approaches are explained in the rest of this section, namely:

- $f(\cdot, \cdot)$ is an algebraic function.

- $f(\cdot,\cdot)$ is calculated by a learning scheme like a neural network.

3.1 Kitter rules

The function $f(\cdot,\cdot)$ can be considered to be *algebraic*. This case is studied in depth in [14], where four combination rules are proposed.

Applying Bayes' rule in eq. (8) and *assuming statistical independence* between S^{s} and S^{u}:

$$P\left(\omega/\left[s^{\mathrm{s}}, s^{\mathrm{u}}\right]\right) = \frac{1}{P(\omega)} P\left(\omega/s^{\mathrm{s}}\right) P\left(\omega/s^{\mathrm{u}}\right)$$

If the speaker and utterance verifiers perform correctly, it can be expected that the probabilities $P\left(\omega = 1/s^{\mathrm{s}}\right)$ and $P\left(\omega = 1/s^{\mathrm{u}}\right)$ will be monotonically increasing functions on s^{s} and s^{u}. It will therefore be possible to find thresholds to transform the authentication criteria expressed in eq. (7) obtaining:

$$\begin{aligned} P\left(\omega = 1/s^{\mathrm{s}}\right) P\left(\omega = 1/s^{\mathrm{u}}\right) \geq \varphi' &\Rightarrow \text{accepted} \\ s^{\mathrm{s}} \times s^{\mathrm{u}} \geq \varphi'' &\Rightarrow \text{accepted} \\ s^{\mathrm{s}} + s^{\mathrm{u}} \geq \varphi''' &\Rightarrow \text{accepted} \end{aligned}$$

The second and third criteria in this equation correspond to the product and sum rules proposed by Kitter.

(1) *Product rule.*

$$f(s^{\mathrm{s}}, s^{\mathrm{u}}) = s^{\mathrm{s}} \times s^{\mathrm{u}} \tag{9}$$

With this assumption $f(\cdot,\cdot)$ combines the scores generated by the individual classifiers by means of a product operation.

(2) *Sum rule.*

$$f(s^{\mathrm{s}}, s^{\mathrm{u}}) = s^{\mathrm{s}} + s^{\mathrm{u}} \tag{10}$$

Thus, $f(\cdot,\cdot)$ combines the scores by means of an addition operator.

(3) *Minimum rule.* Taking into account that

$$\forall\, a, b \leq 1, \quad a \times b \leq \min(a, b)$$

product rule can be approximated by the scores minimum:

$$f(s^{\mathrm{s}}, s^{\mathrm{u}}) = \min(s^{\mathrm{s}}, s^{\mathrm{u}})$$

(4) *Maximum rule.* Equivalently

$$\forall\ (a,b),\ \ \frac{1}{2}\,(a+b) \leq \max(a,b)$$

and sum rule can be approximated by the maximum:

$$f(s^{\mathrm{s}}, s^{\mathrm{u}}) = \max(s^{\mathrm{s}}, s^{\mathrm{u}})$$

3.1.1 Error sensivity

As was previously explained, s^{s} and s^{u} come from approximating two likelihood ratios of two statistical hypothesis tests. An important point here is that they have been considered to be correctly computed so far, when, actually, they are estimates. Then, $\hat{P}(\omega/s^{\mathrm{s}})$ and $\hat{P}(\omega/s^{\mathrm{u}})$ will deviate from the true values $P(\omega/s^{\mathrm{s}})$ and $P(\omega/s^{\mathrm{u}})$ by errors that will extend their influence to the scores s^{s} and s^{u}:

$$\begin{aligned}\hat{s}^{\mathrm{s}} &= s^{\mathrm{s}} + \chi^{\mathrm{s}} \\ \hat{s}^{\mathrm{u}} &= s^{\mathrm{u}} + \chi^{\mathrm{u}}\end{aligned}$$

The rest of this section is devoted to consider the effect of these estimation errors on the proposed rules.

Introducing (11) in the product rule (9)

$$\hat{s}^{\mathrm{s}}\hat{s}^{\mathrm{u}} = (s^{\mathrm{s}} + \chi^{\mathrm{s}})(s^{\mathrm{u}} + \chi^{\mathrm{u}}) = s^{\mathrm{s}}s^{\mathrm{u}}\left(1 + \frac{\chi^{\mathrm{s}}}{s^{\mathrm{s}}}\right)\left(1 + \frac{\chi^{\mathrm{u}}}{s^{\mathrm{u}}}\right)$$

A similar analysis of the sum rule (10) gives:

$$\hat{s}^{\mathrm{s}} + \hat{s}^{\mathrm{u}} = s^{\mathrm{s}} + \chi^{\mathrm{s}} + s^{\mathrm{u}} + \chi^{\mathrm{u}} = (s^{\mathrm{s}} + s^{\mathrm{u}})\left(1 + \frac{\chi^{\mathrm{s}} + \chi^{\mathrm{u}}}{s^{\mathrm{s}} + s^{\mathrm{u}}}\right)$$

Taking into account that $\chi^{\mathrm{s}} \ll s^{\mathrm{s}}$ and $\chi^{\mathrm{u}} \ll s^{\mathrm{u}}$ second order terms can be eliminated. Comparing these equations it can be seen that

$$\left[1 + \frac{\chi^{\mathrm{s}} + \chi^{\mathrm{u}}}{s^{\mathrm{s}} + s^{\mathrm{u}}}\right] \leq \left[1 + \frac{\chi^{\mathrm{s}}}{s^{\mathrm{s}}} + \frac{\chi^{\mathrm{u}}}{s^{\mathrm{u}}}\right]$$

So, the addition rule will be less sensitive than the product rule to errors in the computation of the estimates.

3.1.2 Practical implementation

First, s^{s} and s^{u} were transformed into scores that contain the distances to their respective thresholds:

$$s^{\mathrm{s}\prime} = s^{\mathrm{s}} - \varphi^{\mathrm{s}}$$
$$s^{\mathrm{u}\prime} = s^{\mathrm{u}} - \varphi^{\mathrm{u}}$$

$s^{\mathrm{s}\prime}$ and $s^{\mathrm{u}\prime}$ will be lesser or greater than zero depending on whether the likelihoods are lesser or greater than their correspondent thresholds, respectively.

Afterwards, and for smoothing purposes, a sigmoid function was applied:

$$\widetilde{s^{\mathrm{s}}} = \frac{1}{1 + \mathrm{e}^{-a s^{\mathrm{s}\prime}}}$$
$$\widetilde{s^{\mathrm{u}}} = \frac{1}{1 + \mathrm{e}^{-a s^{\mathrm{u}\prime}}}$$

$\widetilde{s^{\mathrm{s}}}$ and $\widetilde{s^{\mathrm{u}}}$ will tend to 0 or 1 depending on whether they exceed their thresholds or not. The steepness of the sigmoid is controlled by a. Previous tests led us to use a value of $a = 5$ for this parameter.

These newly calculated scores can be combined into a new one using Kitter rules:

- *Maximum rule*:

$$V^{\max} = \max(\widetilde{s^{\mathrm{s}}}, \widetilde{s^{\mathrm{u}}}) - 0.5$$
$$V^{\max} > 0 \Rightarrow \text{accepted} \tag{11}$$

- *Sum rule*:

$$V^{\mathrm{sum}} = \widetilde{s^{\mathrm{s}}} + \widetilde{s^{\mathrm{u}}} - 1$$
$$V^{\mathrm{sum}} > 0 \Rightarrow \text{accepted} \tag{12}$$

- *Minimum rule*:

$$V^{\min} = \min(\widetilde{s^{\mathrm{s}}}, \widetilde{s^{\mathrm{u}}}) - 0.5$$
$$V^{\min} > 0 \Rightarrow \text{accepted} \tag{13}$$

- *Product rule*:

$$V^{\mathrm{prod}} = \widetilde{s^{\mathrm{s}}} * \widetilde{s^{\mathrm{u}}} - 0.5$$
$$V^{\mathrm{prod}} > 0 \Rightarrow \text{accepted} \tag{14}$$

From a practical point of view, the acceptation criteria expressed by Kitter rules in eqs. (11), (12), (13) and (14) will divide the area of the $\widetilde{s^{\mathrm{s}}}\widetilde{s^{\mathrm{u}}}$ plane

defined by ($0 \leq \widetilde{s^{\mathrm{s}}} \leq 1, 0 \leq \widetilde{s^{\mathrm{u}}} \leq 1$) into two areas: one where the verification attempt will be accepted and another where it will be rejected. These areas can be observed in figure 2 where the rejection region is placed towards the origin.

3.2 Neural Networks

An important feature of algorithms like neural networks is their capability to learn the parametric distribution directly from the experimental data. Instead of proposing a formulation for $f(\cdot, \cdot)$, this function is estimated from the training material. In this paper, results are presented using simple neural networks like the ones shown in figure 3:

- A perceptron (figure 3.a).
- A three-layer neural network with two neurons in the hidden layer (figure 3.b).
- A three-layer neural network with three neurons in the hidden layer (figure 3.c).

These two last schemes will be referred to as NN2 and NN3 in the rest of this paper.

The training of neural networks consists of a learning algorithm that enables them to find an optimally calculated linear boundary in the decision space. In a speaker authentication task, the thresholds are therefore substituted by the network itself that will decide whether a pair of values corresponds to a customer authentication attempt or not.

Since speaker authentication is a classical two-class classification problem, a cross-entropy error function [19,20] was used to train the neural networks. The data is divided into two classes: class C_1 corresponds to the data in which the presumed and actual identities are the same (customer tests) and C_2 is the opposite (impostor tests). The output y of the neural networks represents the a posteriori probability $P(C_1/\mathbf{s})$ for class C_1 while the a posteriori probability of class C_2 will be given by $P(C_2/\mathbf{s}) = 1 - y$, where $\mathbf{s} = [s^{\mathrm{u}}, s^{\mathrm{s}}]$. This can be achieved if a target coding scheme is considered for which $t = 1$ when the inputs s^{u} and s^{s} belong to class C_1 (a customer test) and $t = 0$ when the inputs belong to C_2 (an impostor test). These two probabilities can be combined into a single expression so that the probability of observing either target value is

$$P(t/\mathbf{s}) = y^t(1-y)^{1-t}$$

With this interpretation, the likelihood of observing the training data set is then given by

$$\prod_{\mathrm{n}=1}^{\mathrm{N}} (y_{\mathrm{n}})^{(t_{\mathrm{n}})}(1-y_{\mathrm{n}})^{(1-t_{\mathrm{n}})}$$

where N is the total number of tokens. For numerical reasons, it is convenient to minimize the negative logarithm of this likelihood. This leads to the cross-entropy error function:

$$E = -\frac{1}{N}\sum_{\mathrm{n}=1}^{\mathrm{N}}\{t_{\mathrm{n}} ln(y_{\mathrm{n}}) + (1-t_{\mathrm{n}}) ln(1-y_{\mathrm{n}})\}$$

In this equation, the classification errors in customer ($t_{\mathrm{n}} = 1$) and impostor ($t_{\mathrm{n}} = 0$) tests are taken into account by left and right part of each term of the summation, respectively. A modification in the training process was introduced due to the unbalanced numbers of customer and impostor training tokens. If N is the total number of training files, N_{cus} belong to customers and N_{imp} belong to impostors. The previous equation is modified to take into account the fact that $N_{\mathrm{cus}} \ll N_{\mathrm{imp}}$ (see sec. 4):

$$E = -\frac{1}{N}\sum_{\mathrm{n}=1}^{\mathrm{N}}\left[\frac{N_{\mathrm{imp}}}{N} t_{\mathrm{n}} ln(y_{\mathrm{n}}) + \frac{N_{\mathrm{cus}}}{N}(1-t_{\mathrm{n}}) ln(1-y_{\mathrm{n}})\right]$$

In order to minimize this error function, the well-known back-propagation algorithm for neural networks was used. After some trials, a learning rate of $\lambda = 0.25$ was used and the iterative algorithm was stopped when $(E_{\mathrm{l}-1} - E_{\mathrm{l}}) \leq \eta$ with $\eta = 10^{-6}$ in all cases.

4 Experimental framework

4.1 The database

Presently, many speaker and speech recognition systems are migrating from the laboratory demos to many services and products. One of the main problems in this migration is the mismatch between training and testing conditions in laboratory and training and testing conditions in the real world. This mismatch can be reduced if real-world databases are used in the laboratory. Nevertheless, there are not so many public databases available specifically collected for speaker recognition purposes under real world conditions. We

can mention COST250 PolyCost database [21] and our own database called "TelVoice" [22], as two examples of such databases.

All the experiments presented in this paper were conducted using TelVoice. This database was recorded during 1996, 1997 and 1998 and consists of telephone speech recorded using a PC computer equipped with a Sound Blaster sound card and a hardware interface connected to the parallel port. This interface was designed and built up in our research laboratory.

The number of speakers is 59 (39 male and 20 female) with 10 phone calls each. All the fields of the database were recorded in Spanish. The time between recordings is variable across speakers, ranging from three days to more than one year. This can be seen as an inconvenience, but this fact introduces realism into the database.

TelVoice has been carefully designed to obtain much as meaningful material as possible. In each session, there are 10 items varying from isolated digits, strings of digits, connected digits, phrases, and free speech.

If the speech content is going to be used in the recognition process, the use of a set of previously trained speaker-independent models is required to represent the linguistic units that can appear in the test material. In this case, a set of 25 phoneme-like models trained using a telephone database was used. The models are left-to-right 3 state HMM's with 16 mixtures per state.

4.2 Speech analysis

Some decisions about recording conditions and speech parameterization have been taken. The voice was sampled at 8 kHz and off-line filtered to remove the 50 Hz electric-supply noise. Energy and 12 mel-cepstra coefficients were computed using a Hamming window with frame length of 20 msec. and a frame period of 10 msec. Liftering (by a factor of 22) was also used. First derivatives of the energy and the mel-cepstra were appended to the parameters of each frame. This makes a total of 26 parameters per vector. Parameterization details can be seen in table 1.

A VAD (Voice Activity Detector) was used to process all the speech material of the database with the purpose of eliminating the effect of the background noise in the recognition process. The VAD is similar to the one described in [23] with some additional modifications to improve its efficiency in noisy environments and to work in a MBE (Multi-Band Excitation) speech coder [24].

4.3 Authentication experiments

Two operating scenarios for a speaker authentication system are differentiated:

- Assessment in real-world conditions. In this case, the values of the authentication thresholds must be estimated in advance or *a priori*, that is, before the accessing attempt. In this case, our goal was to find such thresholds where the system behaves close to the EER (Equal Error Rate) point.
- Assessment in research environment. Assessing the authentication systems regardless of the thresholds setting problem is the aim. In this case, the ROC (Receiver Operating Characteristic) curve is used as a figure of merit, and the EER is drawn from this curve.

The experiments presented in this paper were conducted with a subset of TelVoice consisting of those speakers with a minimum of 6 recording sessions. This subset consists of 46 speakers (32 males and 14 females).

4.3.1 Training material

For the speaker verification system, training material consists of:

- Digits 0 to 9 in ascending order (approximately 12 sec.).
- Two fixed phrases: *El templo oscuro y triste quedaba a pocos pasos del palacio de la familia real* and *Las primeras palabras que brotaron de sus labios fueron para darme las gracias* (7 sec. each).

Sessions no. 1 and no. 2 were used as training material. This makes a total of 52 sec. of data per speaker before VAD.

4.3.2 Testing material

As testing material, pronunciations of the Spanish National Identity Card number were taken. This consists of eight digits (5 sec. before VAD). The speakers were asked to pronounce it naturally (digit by digit, grouping digits or as a whole, as they usually do) but consistently across sessions. In each session, the speakers uttered their number four times, so there are 4 tokens/session $*$ 4 test sessions/speaker = 16 test tokens/speaker.

The experiments thus proposed are *text-independent*, as the speech contents of the test and the training tokens differ. This allows this system to be deployed in real-world applications where passwords can be modified.

4.3.3 Experimental protocol

Both in threshold estimation and in testing, when a speaker is used as a client, all 4 tokens per session are used. Whenever a speaker performs as an impostor, only one token per session is used. The motivation behind this election is to alleviate the unbalance between impostor and client tests in the experimental setup.

Thresholds can be speaker dependent or universal. When the size of the database is small, using speaker dependent thresholds may have lower statistical significance than using a global one. In this article results are presented using global thresholds.

When using a set of speakers for estimating background models, there is a potential bias if material from these speakers is used in authentication experiments. In a fair test, this material should belong to a set of speakers independent of the set to be tested upon. To avoid this source of problems, the set of speakers was divided into two parts (A and B), each including 23 speakers (16 male and 7 female). Then, world models were estimated with training material from A while recognition experiments were performed using test tokens from B and viceversa. To avoid problems caused by a possible biased division of the database, the database was again split into two other sets (X and Y) with 23 speakers each and this procedure was repeated.

The details of the experimental protocol is explained in the following. Subindexes will be used to denote the distinct sessions of the database. For example, X_{12} corresponds to the first and second sessions belonging to the speakers included in set X.

4.3.3.1 A priori authentication The training and testing procedures in the case when a priori calculated thresholds are used are presented in an algorithm-like notation:

foreach Set **in** ('A','B','X','Y')
  if (Set == 'A') **then** WorldSet = 'B';
  if (Set == 'B') **then** WorldSet = 'A';
  if (Set == 'X') **then** WorldSet = 'Y';
  if (Set == 'Y') **then** WorldSet = 'X';
  Train SpeakersGMMs **using** Set_{12};
  Train WorldGMM **using** WorldSet_{12};
  foreach TrainSession **in** (3,4,5,6)
    Calculate Thresholds **using**
      $\text{Set}_{\text{TrainSession}}$, SpeakersGMMs, WorldGMM;
    foreach TestSession **in** (3,4,5,6) **and** TestSession≠TrainSession

Obtain %FA **and** %FR **using**
$\text{Set}_{\text{TestSession}}$, SpeakersGMMs, WorldGMM, Thresholds;

As stated before, tokens from one session were used for a priori threshold estimation corresponding to the EER points. These thresholds were used to perform a priori authentication experiments on the rest of the testing material of the database. The performance of the systems thus measured is given by two rates, namely false acceptance (%FA) and false rejection (%FR) percentage rates.

In this case, the number of experimental tests is:

$$4\ sets\ \times\ 23\ \frac{speakers}{set}\ \times 4\ \frac{training\ session}{speaker}\ \times\ 3\ \frac{testing\ sessions}{training\ session}\ \times \\ \times\ 4\ \frac{tests}{testing\ session}\ =\ 4416\ clients\ tests$$

$$4\ sets\ \times\ 23\ \frac{speakers}{set}\ \times 4\ \frac{training\ session}{speaker}\ \times\ 3\ \frac{testing\ sessions}{training\ session}\ \times \\ \times\ 22\ \frac{impostors}{testing\ session}\ \times 1\ \frac{tests}{impostor}\ =\ 24288\ impostors\ tests$$

4.3.3.2 A posteriori authentication The training and testing procedures with a posteriori threshold estimation are the following:

foreach Set **in** ('A','B','X','Y')
  if (Set == 'A') **then** WorldSet = 'B';
  if (Set == 'B') **then** WorldSet = 'A';
  if (Set == 'X') **then** WorldSet = 'Y';
  if (Set == 'Y') **then** WorldSet = 'X';
  Train SpeakersGMMs **using** Set_{12};
  Train WorldGMM **using** WorldSet_{12};
  Calculate Thresholds **using**
    Set_{3456}, SpeakersGMMs, WorldGMM;
  Obtain %EER **using**
    Set_{3456}, SpeakersGMMs, WorldGMM, Thresholds;

To obtain the %EER figures, a methodology close to the one proposed to perform experiments on PolyCost database [25] has been used.

The total number of experimental tests is:

$$4\ sets\ \times\ 23\ \frac{speakers}{set}\ \times 4\ \frac{sessions}{speaker}\ \times\ 4\ \frac{tests}{session}\ =\ 1472\ clients\ tests$$

$$4\ sets\ \times\ 23\ \frac{speakers}{set}\ \times 22\ \frac{impostors}{speaker}\ 4\ \frac{sessions}{impostors}\ \times\ 1\ \frac{tests}{impostor}\ =$$
$$=\ 8096\ impostors\ tests$$

5 Experimental results

In this section, experimental results obtained with the verification systems described in previous sections are presented. All the tests were performed using the experimental framework presented in section 4 based on the database TelVoice.

Experimental results obtained with the combination criteria described in section 3 are also presented, both for a posteriori and a priori estimated thresholds. When a priori thresholds are used, their values correspond to those calculated for the speaker and utterance verification systems. When a posteriori thresholds are used, these are calculated specifically for the considered combination criterion.

5.1 Speaker verifier

The first aspect investigated is the relationship between the number of mixtures and performance. The performance was evaluated for a GMM-based speaker verification system with a number of mixtures ranging from 1 to 96. The number of mixtures of the world-model is 92 or 184. All the models were trained using the iterative Expectation-Maximization (EM) algorithm [26,27] on initial estimates obtained using *k-means* vector quantization [28]. In both cases 20 iterations were generally sufficient for convergence and the maximum number of iterations was limited to 60. Covariance-tied models were used across all the experiments to avoid problems caused by scarcity of data.

- *A priori speaker verification*: the performance of the speaker verifier system thus obtained can be seen in table 2. As can be observed, there is little variation in performance over 16 mixtures.
- *A posteriori speaker verification*: in table 2 EER's, which were calculated when the size of the models varies, can be observed. DET (Detection Error Tradeoff) curves [29] can also be observed in figure 4. It can be observed that the performance increases with the number of Gaussians. However, there is little improvement in performance from 32 to 96 mixtures. The

reason could be the saturation of the model capability due to the scarcity of training data.

5.2 Utterance Verifier

The results for the this system in both a priori and a posteriori verification tasks are included in table 2.

Compared with the Speaker Verifier, the overall performance of this system is significantly lower. This can be caused by the mismatch between training and testing conditions. Several techniques can be used to reduce this mismatch and obtain an improvement in performance, but this topic is out of the scope of this article, since the emphasis has been put on the advantage of the combination of verification systems.

5.3 Kitter rules

In the case where speaker or utterance verification systems were used, ROC curves were calculated when the value of the threshold varies. In this case, there are two likelihoods (speaker and utterance) and the values of their thresholds can be placed in pairs in the XY plane. Then, false acceptance and false rejection percentages can be represented as two surfaces in a three dimensional space, and the %EER point will correspond to the minimum value on a line defined by the intersection of these surfaces. For example, figure 5 represents the average of the surfaces obtained in an a posteriori speaker authentication task using the maximum rule. The number of mixtures of the Speakers' and the World models is 64 and 92, respectively.

- *A priori speaker authentication*: in this case, thresholds calculated a priori for the speaker and utterance verifiers were used. By examining the set of formulas of eqs. (11) to (14), it can be seen that

  $$V^{\max} \geq V^{\text{sum}} \geq V^{\min} \geq V^{\text{prod}}$$

  Then, it can be expected that false acceptance rate will decrease and false rejection rate will increase from left to right. This predicted behavior corresponds to what actually happens, as it can be seen in figure 6.
- *A posteriori speaker authentication*: the results obtained in this task can be observed in figure 6. It can be seen that the best performance of these four combination criteria is obtained using the sum rule. This is consistent with its higher insensivity to errors in the estimation of the likelihoods, as presented in section 3.1.

- *A priori speaker authentication*: in table 3 and in figure 6 the results obtained with the neural networks in this task are included. It can be observed that the neural networks outperform the systems that make use of the Kitter rules. It can also be observed that there is some improvement when the complexity of the networks is increased from a perceptron to the NN2 case. However, there is no significant improvement when a NN3 network is used, due to the scarcity of training data. There is also an unbalanced behavior due to the scarcity of customers' tests in training data.

  In figure 7 it can be seen how decision frontiers are placed in these schemes. This figure represents the neural networks where the number of mixtures for the speaker's and the world GMMs is 64 and 92, respectively.
- *A posteriori speaker authentication*: in table 3 and in figure 6 it can be seen that these architectures outperform Kitter rules criteria. As before, better results are obtained when using a NN2 than when using a perceptron, but there is no improvement in using a NN3 instead of a NN2 network.

6 Conclusions

In this paper, speaker authentication systems based on the combination of several speaker classifiers have been presented. After describing two different approaches for building up a speaker authentication system, a theoretical framework is derived in order to define the combination procedure. The design of the individual classifiers: the "utterance verifier" and the "speaker verifier" respectively is carried out taking into account practical implementation issues such as complexity, speaker enrollment time, required resources, etc. A common "test hypothesis formulation" is used to derive the verification formulation for each individual classifier. A likelihood ratio is formed where the null and alternate hypotheses are modeled differently depending on the type of verifier. In the case of the speaker verifier, null hypothesis is modeled using a speaker GMM and alternate hypothesis is modeled using the world-model approximation. In the utterance verifier, null hypothesis is approximated by the log-probability per frame given that the utterance can be modeled by concatenation of a string of sub-word HMM's; alternate hypothesis is modeled using the closest HMM's to the actual string of competitors sub-word HMM's. Combination rules as product rule, sum rule, minimum rule, maximum rule are used. Neural networks have also been used as a combination strategy.

The performance of each individual verifier has been estimated using a realistic database such as TelVoice, and defining a set of comprehensive experiments. In the case of the speaker verifier, the relationship between the number of Gaus-

sian mixtures and performance has been investigated. The main conclusion drawn from the results is that with 16-mixtures GMMs a good compromise between performance, complexity and adequate use of the training material is achieved. For speaker authentication purposes, the utterance verifier performs worse than the speaker verifier. Moreover, there is little room for improvement in this case, apart maybe from the use of better trained HMM models. Nevertheless, experimental results show that there is always an improvement in performance when the two verifiers are combined. This is one of the main conclusions of this paper and the reason why new lines for further work appear. Regarding the best way of combining, experiments show that the neural network approximation outperforms the others due to its ability to learn the optimal operation point from the data.

References

[1] J. P. Campbell, Speaker Recognition: A tutorial, *Proceedings of the IEEE* (1997) 1437–1462.

[2] S. Furui, Speaker Recognition, in: K. Ponting, ed., *Computational Models of Speech Pattern Processing* (NATO ASI Series, Springer Verlag, 1999) 132–142.

[3] Q. Li, B. H. Juang, Q. Zhou and C. H. Lee, Verbal Information Verification, *Proceedings of the EuroSpeech* **2** (1997) 839–842.

[4] T. Matsui and S. Furui, Speaker Adaptation of Tied-Mixture-Based Phoneme Models for Text-Prompted Speaker Recognition, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* **1** (1994) 125–128.

[5] R. A. Sukkar, A. R. Setlur, C. H. Lee, and J. Jacob, Verifying and Correcting Recognition String Hypotheses using Discriminative Utterance Verification, *Speech Communication* **22** (1997) 333–342.

[6] T. Matsui and S. Furui, Comparison of Text-Independent Speaker Recognition Methods Using VQ-Distortion and Discrete/Continuous HMM's, *IEEE Transactions on Speech and Audio Processing* **2** (1994) 456–459.

[7] D. A. Reynolds, Speaker Identification and Verification using Gaussian Mixture Speaker Models, *Speech Communication* **17** (1995) 91–108.

[8] A. E. Rosenberg, C. H. Lee and F. K. Soong, Sub-Word Unit Talker Verification Using Hidden Markov Models, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (1990) 269–272.

[9] F. K. Soong and A. E. Rosenberg, On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition, *IEEE Transactions on Acoustics, Speech and Signal Processing* **36** (1988) 871–879.

[10] S. Tibrewala and H. Hermansky, Sub-Band Based Recognition of Noisy Speech, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* **2** (1997) 1255–1258.

[11] C. García Mateo, W. Reichl and S. Ortmanns, On Combining Confidence Measures in HMM-Based Speech Recognizers, *Proceedings of 1999 IEEE Workshop on Automatic Speech Recognition and Understanding* (1999).

[12] L. Xu, A. Krzyzak and C. Y. Suen, Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition, *IEEE Trans. on Systems, Man and Cibernetics* **22** (1992) 418–435.

[13] T. Matsui and S. Furui, Concatenated Phoneme Models for Text-Variable Speaker Recognition, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* **2** (1993) 391–394.

[14] J. Kitter, M. Hatef, R. Duin and J. Matas, On Combining Classifiers, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **20** (1998) 226–239.

[15] D. A. Reynolds, Robust Text-Independent Speaker Identification using Gaussian Mixture Speaker Models, *IEEE Transactions on Speech and Audio Processing* **3** (1995) 72–83.

[16] C. García Mateo and C. Lee, A Study on Subword Modelling for Utterance Verification in Mexican Spanish, *Proceedings of 1997 IEEE Workshop on Automatic Speech Recognition and Understanding* (1997).

[17] D. A. Reynolds, Comparison of Background Normalization Methods for Text-Independent Speaker Verification, *Proceedings of the EuroSpeech* **2** (1997) 963–966.

[18] R. A. Sukkar and C. H. Lee, Vocabulary Independent Discriminative Utterance Verification for Nonkeyword Rejection in Subword Based Speech Recognition, *IEEE Transactions on Speech and Audio Processing* **2** (1994) 420–429.

[19] C. M. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, 1995) 230–236.

[20] G. Hinton, Connectionist Learning Procedures, *Technical report CMU-87-115* (Carnegie Mellon University, 1987).

[21] D. Petrovska, J. Hennebert, H. Melin, and D. Genoud, POLYCOST: A Telephone-Speech Database for Speaker Recognition, *Proceedings of the Workshop on Speaker Recognition and its Commercial and Forensic Applications (RLA2C)* (Avignon, France, 1998) 211–214.

[22] L. Rodríguez Liñares, Estudio y Mejora de Sistemas de Reconocimiento de Locutores mediante el Uso de Información Verbal y Acústica en un Nuevo Marco Experimental, *Ph. D. Thesis* (Dpto. de Tecnoloxías das Comunicacións, Universidade de Vigo, Spain, 1999).

[23] R. Tucker, Voice Activity Detection using a Periodicity Measure, *IEE Proceedings* **139** (1992) 377–380.

[24] C. García Mateo and D. Docampo Amoedo, Modeling Techniques for Speech Coding: a Selected Survey, in: A. Figueiras Vidal, ed., *Digital Signal Processing in Telecommunicacions* (Springer Verlag, 1996).

[25] H. Melin and J. Lindberg, Guidelines for Experiments on the PolyCost Database, *Proceedings of the COST250 Workshop on the Application of Speaker Recognition Technologies in Telephony* (Vigo, Spain, 1996) 59–69.

[26] A. P. Dempster, N. M. Laird and D. B. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society* **39** (1977) 1–38.

[27] L. R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of the IEEE* **77** (1989) 257–285.

[28] Y. Linde, A. Buzo and R. M. Gray, An algorithm for vector quantization, *IEEE Transactions on Communication* **28** (1980) 84–95.

[29] A. Martin, G. Doddington, T. Kamm, M. Ordowski and M. Przybocki, The DET Curve in Assessment of Detection Task Performance, *Proceedings of the EuroSpeech* **4** (1997).

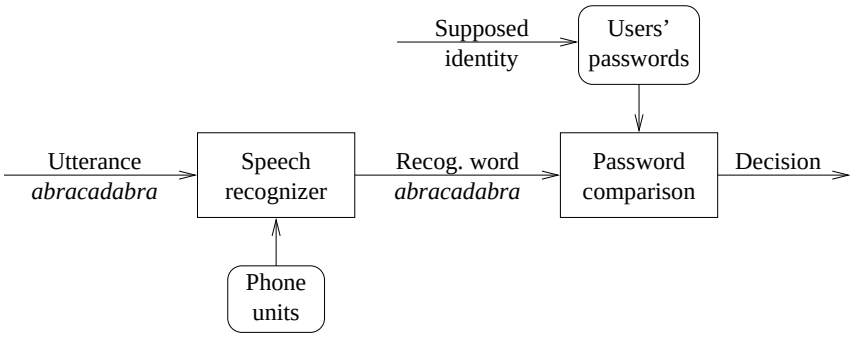

Fig. 1. Architecture of an utterance verification system

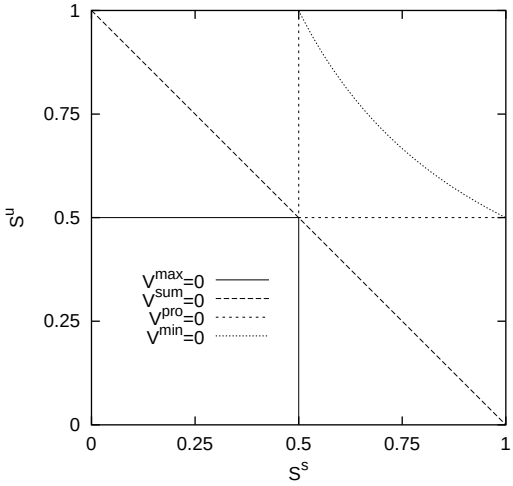

Fig. 2. Kitter rules decision boundaries

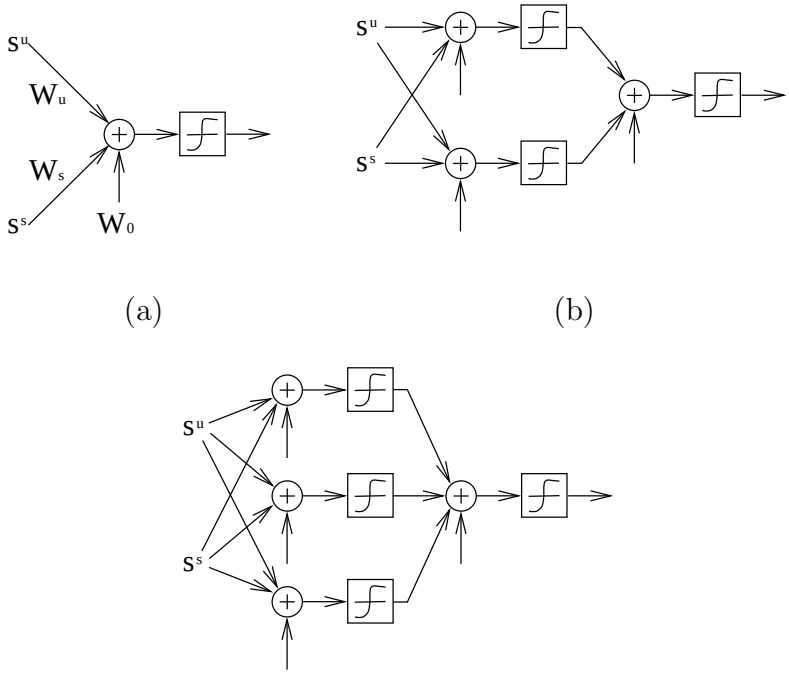

Fig. 3. Neural networks schemes: perceptron (a) and three-layer neural network with two (b) and three (c) neurons in the hidden layer

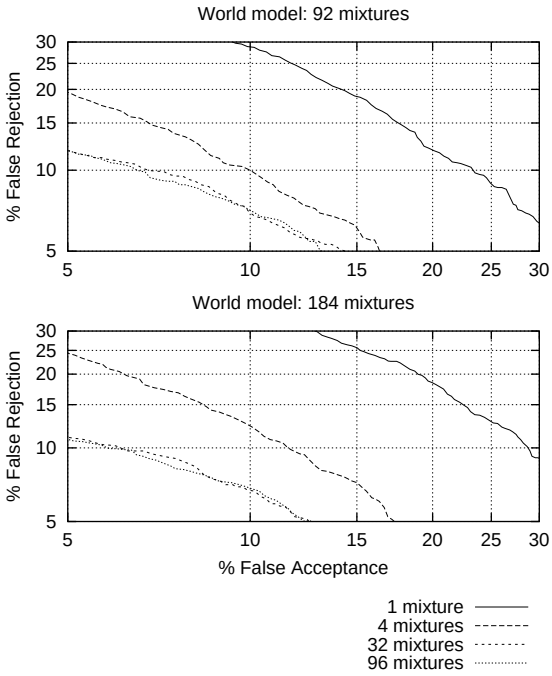

Fig. 4. A posteriori speaker verification performance

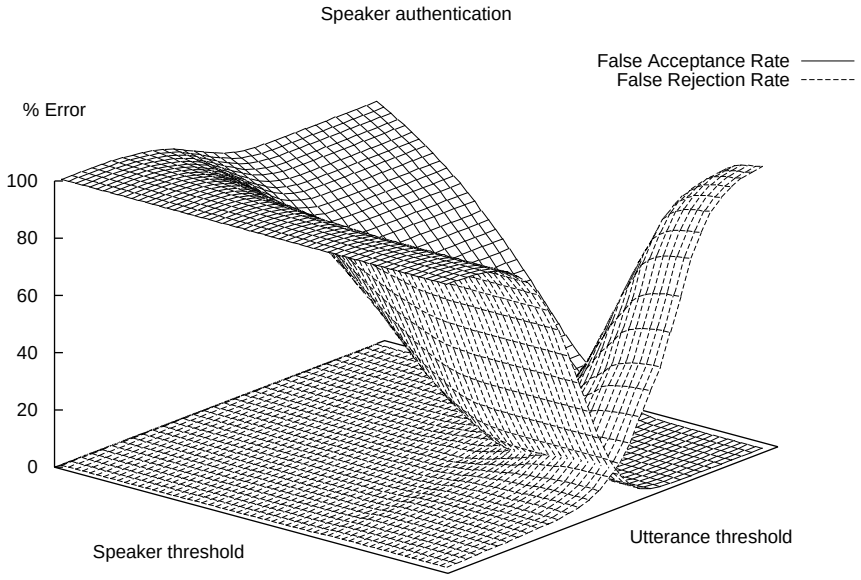

Fig. 5. A posteriori speaker authentication: ROC surfaces

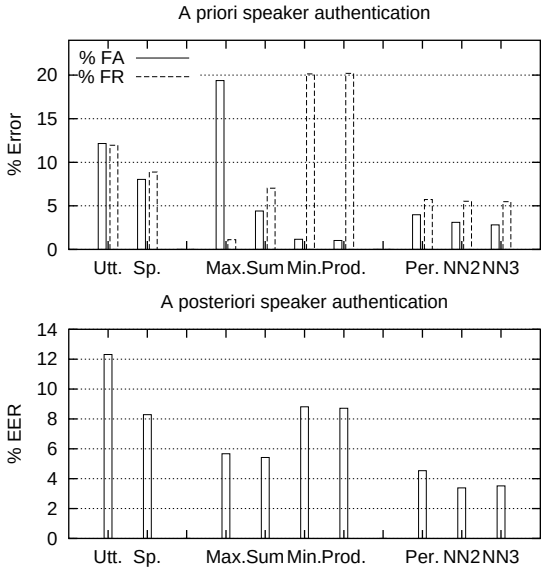

Fig. 6. Performance of the proposed speaker authentication systems (utterance and speaker verifiers, Kitter rules combination and neural network schemes), both in an a priori and a posteriori speaker authentication tasks

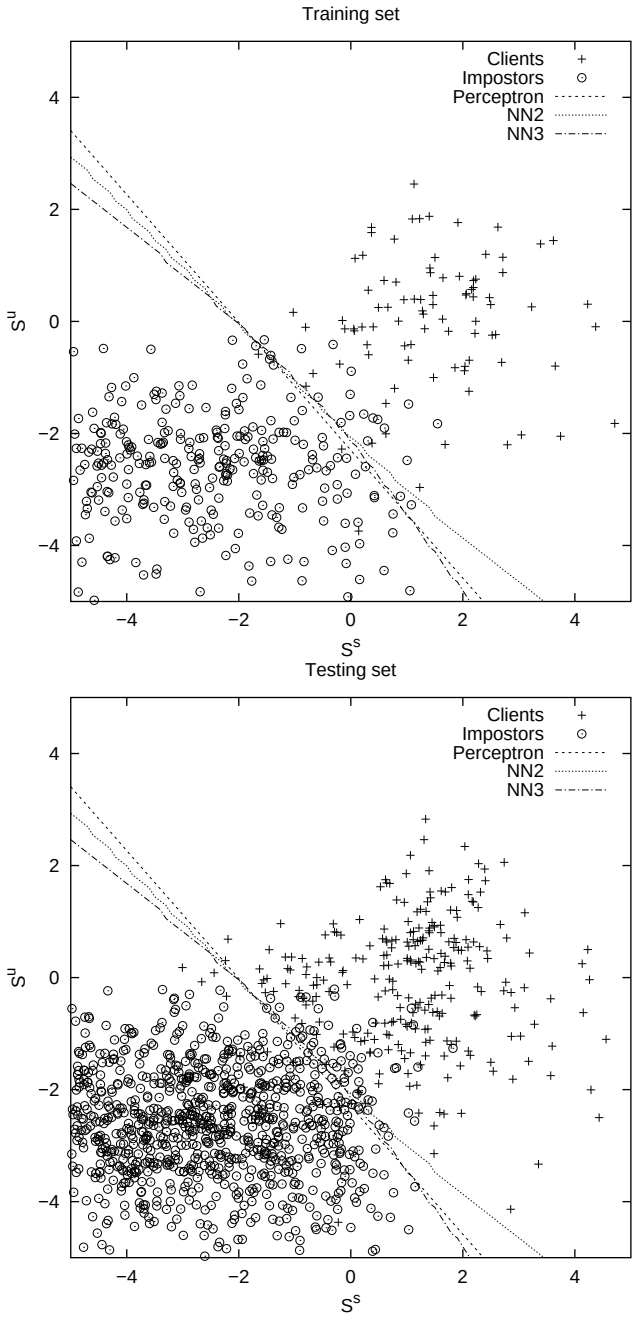

Fig. 7. Neural networks decision boundaries

Table 1
TelVoice - Parameterization details for the front-end block

Parameterization:	MFCC's + Energy + Δ-MFCC's + Δ-Energy
No. of frames for Δ calculation:	5
No. of vector coefficients:	$(12+1)*2=26$
Window type and length:	Hamming 20 msec.
Frame period:	10 msec.
No. of mel filters:	20
Preemphasis:	$k=0$
Liftering:	22
Passband:	300-3400 Hz

Table 2
Speaker and utterance verification performance

Speaker verification				
No. Mixtures		A post.	A priori	
World	Speaker	%EER	%FA	%FR
92	1	16.55	16.26	17.46
	4	10.00	9.93	10.30
	8	8.90	8.74	9.35
	16	9.15	9.17	9.60
	32	8.62	8.35	9.04
	64	8.29	8.03	8.88
	96	8.35	8.37	8.54
184	1	19.38	19.25	19.70
	4	10.80	10.87	11.16
	8	8.92	8.65	9.47
	16	8.97	8.70	9.51
	32	8.25	7.93	8.70
	64	7.86	7.91	8.40
	96	8.00	7.78	8.56
Utterance		12.31	12.15	11.96

Table 3
Speaker Authentication using Neural Network Combination

		Perceptron			NN2			NN3		
Mixtures		A post.	A priori		A post.	A priori		A post.	A priori	
W.	Sp.	%EER	%FA	%FR	%EER	%FA	%FR	%EER	%FA	%FR
	1	6.44	6.39	7.20	5.89	4.59	8.33	5.93	4.21	8.88
	4	5.12	4.89	6.36	4.02	3.96	5.53	3.90	3.75	5.77
	8	4.72	4.21	5.93	3.56	3.40	5.25	3.61	3.10	5.66
92	16	4.76	4.17	5.89	3.44	3.33	4.98	3.38	3.15	5.57
	32	4.59	4.08	5.77	3.35	3.14	5.64	3.48	3.14	5.41
	64	4.54	3.98	5.73	3.38	3.11	5.53	3.52	2.82	5.48
	96	4.62	4.06	5.89	3.48	3.03	5.75	3.66	2.90	6.09
	1	7.17	7.09	7.90	6.48	5.46	8.76	6.06	4.85	9.28
	4	5.16	5.01	6.34	4.59	4.22	6.09	4.13	4.21	6.32
	8	4.48	4.31	5.75	3.31	3.40	4.82	3.45	3.06	5.55
184	16	4.46	3.99	5.80	3.04	3.17	4.62	3.09	2.77	5.66
	32	4.55	3.88	5.80	3.10	2.97	5.37	3.19	2.99	5.21
	64	4.42	3.83	5.96	3.26	3.12	5.10	3.44	2.85	5.32
	96	4.60	3.95	5.82	3.37	3.15	5.39	3.47	2.85	5.77

Vitae

Leandro Rodríguez Liñares received the MSc and PhD in Telecommunications engineering from the University of Vigo (Spain) in 1994 and 1999, respectively, where he graduated cum laude. His research interests are focussed on speaker and speech recognition, speech sinthesis and dialogue systems. He is associate professor of parallel processing and logic programming at the University of Vigo, Spain.

Carmen García Mateo received the MSc and PhD in Telecommunications engineering from the Polythenic University of Madrid (Spain) in 1987 and 1993, respectively, where she graduated cum laude. Her research interests include speech and speaker recognition, dialogue systems man-machine applications, and speech coding. She is associate professor of discrete signal processing at the University of Vigo, Spain. She is member of IEEE and ISCA.

José L. Alba received the MSc and PhD in Telecommunications engineering from the University of Santiago and University of Vigo (Spain) in 1990 and 1997, respectively, where he graduated cum laude. His research interests include neural networks for classification applications, image segmentation, statistical pattern recognition, automatic speech and speaker recognition and biometrics. He is associate professor of discrete signal processing and image processing at the University of Vigo, Spain. He is member of IEEE.