TEXT DEPENDENT SPEAKER VERIFICATION USING BINARY CLASSIFIERS

Dominique Genoud, Miguel Moreira and Eddy Mayoraz

IDIAP, Dalle Molle Institute of Perceptual Artificial Intelligence
P.O. Box 592, 1920 Martigny, Switzerland
genoud@idiap.ch, miguel@idiap.ch, mayoraz@idiap.ch

ABSTRACT

This paper describes how a speaker verification task can be advantageously decomposed into a series of binary classification problems, i.e. each problem discriminating between two classes only. Each binary classifier is specific to one speaker, one anti-speaker and one word. Decision trees dealing with attributes of continuous values are used as classifiers. The set of classifiers is then pruned to eliminate the less relevant ones. Diverse pruning methods are experimented, and it is shown that when the speaker verification decision is performed with an a priori threshold, some of them give better results than a reference HMM system.

1. INTRODUCTION

Text dependent speaker verification is a very promising domain, where real applications can be designed for telephone or banking services. The importance of the text dependency lies in the fact that the input speech pronounced by the speaker can be controlled. This control is made by a speech recognizer, which performs a temporal segmentation of the input utterances, allowing a comparison of the recognized words with pre-trained speaker models. However, the scoring procedures commonly used simply sum the partial scores of all the models without using a priori knowledge, like, for example, the fact that for a given speaker some of the words he pronounces are more discriminative than others. For this information to be useful, an analysis of that discrimination has to be done automatically for each registered speaker. This paper describes a way of revealing and exploring that discriminative information by building a set of binary classifiers. Each one of these classifiers separates the data of one single word for one couple (*registered speaker, anti-speaker*). A registered speaker is a speaker on which the verification is performed, and an anti-speaker is a predetermined speaker whose speech is used in the binary classification task as the anti-class of the speaker. The outputs of the set of classifiers are merged, using diverse methods, into a single score which is finally compared to a threshold. Both a priori and a posteriori thresholds are used. Binary classifiers have been successfully used with the text independent approach by Castellano et al. [2]. The method proposed here is compared to a classical HMM approach using log likelihood ratio [8].

2. THE DATABASE

The Polycost database [5] used in these experiments is composed of 133 speakers recorded over international telephone line in several sessions. The speakers are from 13 different countries. The part extracted for the present work contains 6 sessions for a fixed subset of 104 speakers; each session is composed of four 10-digit sequences uttered in English (all the digits from 0 to 9 in different order for each sequence). The 10-digit sequences are identical for each speaker and all the sequences have been time-labeled digit by digit using a speech recognizer [5].

The set of 104 speakers is partitioned into three subsets: *CLI* with 82 speakers (48 male, 34 female), who are the clients (registered speakers); *ANSP* with 12 speakers (6m., 6f.) called anti-speakers; and *VALIMP* with the 10 remaining speakers (5m., 5f.). *WORLD* is defined as *ANSP* $\cup$ *VALIMP*.

For every particular registered speaker $s \in$ *CLI*, the training and the evaluation of the classifier require a partition of the 104 times 6 sessions into a **training set**, a **validation set** and a **testing set** done as follows. The first session (four 10-digit sequences) of speaker s constitutes the client samples of the **training set**, while the impostor samples are provided by the first sessions of the 12 anti-speakers in *ANSP*. The second session of speaker s composes the client samples of the **validation set**; the impostor samples are made of four 10-digit sequences chosen randomly among the 60 sessions of speakers in *VALIMP*. The **test set** is composed of the last 4 sessions of speaker s (16 sequences of client utterances) and 81 sessions (324 sequences of impostor accesses), each of which being chosen randomly among the last 4 sessions of each speaker in *CLI* other than s.

For all the experiments, the log energy, 13 LPCC coefficient and their derivatives are extracted from the speech signal. A 28-element vector is thus created at each 10ms. using an analysis window of 25.6ms.

3. THE CLASSIFICATION

3.1. HMM Reference System

Two types of HMM [8] are created for each digit. First, a **world model** is trained on the *WORLD* set, where 58 occurrences of each digit uttered by the 22 different people were extracted. The parameters of the model are estimated by a classic training – Viterbi algorithm and Baum-Welch re-estimation [1]. Second, a **speaker model**, which uses the world model as bootstrap model, is re-estimated with the speaker data. All models have the same HMM left-right one-mixture-per-state structure. Each model has one state per phoneme and one state per phoneme transition [4]. When the tests are performed, for each digit uttered by the speaker, the log likelihood ratio (LLR):

$$LLR_{\mathrm{sw}} = \log(L_{\mathrm{s}}) - \log(L_{\mathrm{w}})$$

is computed, with L_{s}, L_{w} being the likelihood of the speaker and world models respectively. The LLR_{sw} scores of each digit, for a given test utterance, is summed and then compared to a threshold.

3.2. Decomposition into Binary Classifiers

For each speaker s in *CLI* an $M \times N$ matrix D^s of classifiers is built (see Fig. 1). Each row of the matrix is associated with one word (this application being on digits, $M = 10$), and each column with one anti-speaker in *ANSP* ($N = 12$ in this case). The total amount of classifiers for one registered speaker s is 120.

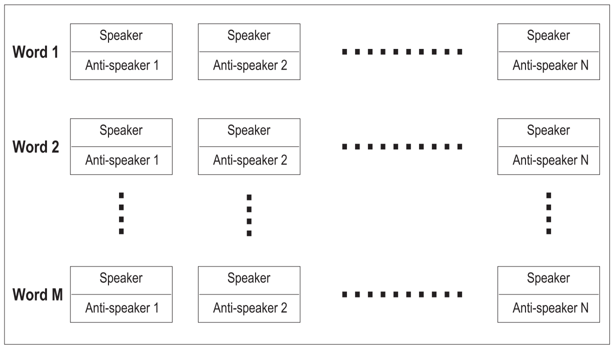

Figure 1: Matrix D^s of classifiers for speaker s

When the membership of a new utterance x to a registered speaker s is to be tested, it is passed to each of the $M \cdot N$ classifiers in D^s. Each classifier returns a value in the range $[0, 1]$ expressing its confidence for x to have been uttered by speaker s. The outputs of the $M \cdot N$ classifiers are then merged through a linear combination and compared to a threshold t :

$$\sum_{i=1}^{M} \sum_{j=1}^{N} \omega_{ij} D^s_{ij}(x) \geq t.$$

The choice of the weights ω will be discussed in Section 3.4. The selection of the threshold t will be explained in Section 4.

The goal of each classifier is to discriminate between the data coming from the speaker and the data coming from a well chosen anti-speaker, and thus to solve a 2-class classification problem.

3.3. Decision Tree Classifier

Among the classical learning algorithms available in Machine Learning for the resolution of classification problems [6], a decision tree-based algorithm has been chosen to implement the classifiers and to test the procedure proposed here. The particular algorithm used here is the well-known C4.5, a decision tree method adapted to use continuous attribute values and developed by Quinlan [7]. In spite of its relative simplicity, the algorithm presents several attractive features: it is accurate and capable of making a good data separation, requires little training time, and the size of the trained models for each speaker is considerably low (500-1000 bytes).

Decision tree learning methods use the training data to recursively build a decision tree. The root node is associated with the whole training data space, and this space is then partitioned in subregions in a recursive way, where each subdivision is associated with the test of an attribute. At each node, C4.5 selects the attribute providing the best information gain, i.e. the one that best represents the required output classification, based on an **entropy measure**. The same procedure is then applied iteratively to the sub-nodes created by the decision. This process continues until all the examples are correctly classified or all the attributes have been used. When the tree is complete, each leaf node corresponds to a class. C4.5 is an extended decision tree algorithm that allows the use of continuous attribute values and that is able to deal with missing values of the attributes. In addition, the tree is pruned after construction by replacing a whole subtree by a leaf node. A pruning is made when the expected error rate in a subtree is greater than in the equivalent single leaf. C4.5 is extensively detailed in [7].

Each classifier in D^s is trained using as input the vectors issued from the registered speaker and from one of the 12 anti-speakers on one of the 10 digits. The C4.5 algorithm separates the training data T into two classes: the speaker data (class 1) and the anti-speaker data (class 0). Each of the 28 elements of the input vectors are considered as a continuous attribute.

An important outcome of the use of this kind of algorithm is the automatic selection of the input coefficients that best separate the couple (speaker, anti-speaker). An entropy criterion is computed to decide the quantity of useful information contained in each LPCC, ΔLPCC coefficients, and in the energy, Δenergy coefficients. The selected coefficients may be different for each classifier, i.e. for each pair speaker/anti-speaker and for each word (e.g. digit).

3.4. Fusion of the Partial Decisions

The output of each classifier provides a local speaker/non-speaker decision. There are several possibilities to recombine these partial decisions. The first and simplest approach consists in taking the mean (or the sum) of all the scores issued from the classifiers in D^s, in other words to choose $\omega_{ij} = \frac{1}{MN}$ for any i, j. This can be compared to the sum of the LLR scores in the HMM algorithm. The difference is that each classifier concentrates only on the separation of one speaker from the other, while in the case of an HMM system using a speaker model and a world model, the separation is done between one speaker and all the others.

Among the $M \cdot N$ classifiers of a speaker, some will give a more accurate output than others. To increase the accuracy of the overall speaker verification task, some classifiers can be pruned ($\omega_{ij} = 0$) and an adequate weighting can advantageously differentiate the remaining ones.

The scores of the classifiers of D^s on the **validation set** will be used to determine ω. Distinct weighting techniques were experimented. They are described below.

3.4.1. Zero-Error Pruning

This method selects only the classifiers with no classification error. The weighting matrix ω will be a binary matrix, with value 1 for the classifiers with no error on the validation set, and 0 for the others.

3.4.2. Distribution Distance Pruning

In this case the distance between the distributions of the speaker and of the impostor is used:

$$Dist_{\text{sp/im}} = (\mu_{\text{sp}} - \mu_{\text{im}}) - (\sigma_{\text{sp}} + \sigma_{\text{im}}) \,,$$

where μ_{sp}, μ_{im} are the means over the validation set of the output values of one binary classifier (i, j) for the speaker and for the impostor respectively, while σ_{sp}, σ_{im} are the standard deviations. For a binary classifier (i, j), if $Dist_{\text{sp/im}}$ is negative, the classifier is pruned, i.e. $\omega_{ij} = 0$. Otherwise, three different approaches were experimented, also based on the values of $Dist_{\text{sp/im}}$.

1. Using a *binary mask*, where all the classifiers with a non-negative distance are used with the same weight $\omega_{ij} = 1$;
2. Using a *normalized continuous mask*, such that all the weights of the classifiers with a non-negative distance are proportional to $Dist_{\text{sp/im}}$ and sum to 1;
3. Using *pruning at a fixed percentage error*. In this case, the weights used in 2) are sorted in descending order, and they are then summed till the sum becomes greater than a threshold Ω. The remaining small weights are set to 0, and finally, the weights are renormalized so that $\sum \omega_{ij} = 1$. This pruning method amplifies the contribution of the best classifiers and suppresses the poorests.

3.4.3. Speaker Distribution Pruning

The mean μ_{sp} and the standard deviation σ_{sp} of the speaker distribution scores are computed. The classifiers with an impostor scoring above $\mu_{\text{sp}} - \sigma_{\text{sp}}$ or more than one speaker scoring below $\mu_{\text{sp}} - \sigma_{\text{sp}}$ are suppressed. All the others are maintained with equal weight 1.

4. RESULTS OF THE EXPERIMENTS

The results obtained with this pairwise coupling method are compared with a classical state-of-the-art HMM-based algorithm. The classifiers are trained on the **training set** (as defined in Sect. 2) and the tests are performed on the **test set**. Two types of results are given here. The first set of results concentrates on the intrinsic performances of the algorithm, using a speaker dependent *a posteriori EER* (Equal Error Rate) threshold. The second set of results give the *living* algorithm performances, using a speaker dependent *a priori EER* threshold computed on the **validation set**. The results are given for the HMM reference system, for the mean of the $M \cdot N$ classifiers (*BP*), using the zero-error pruning criterion (*BP-ze*), using the distribution distance pruning with a binary mask (*BP-dddisc*), the same with a continuous mask (*BP-ddcont*), with a continuous mask with a rejection threshold $\Omega \in [0.5, 0.9]$ (*BP-ddcont-*Ω), and finally the speaker distribution pruning (*BP-spdisc*).

Table 1 shows the results when using an a posteriori EER threshold on the **test set**. It can be observed that the HMM reference system give the best results and that *BP-disc* is also considerably robust. The pruning seems to deteriorate the intrinsic quality of the system, in general.

Table 2 shows the performances obtained by the different classifiers when their output scores are compared to an a priori threshold. This threshold is computed at the EER using the **validation set**. In this case it can be observed that the HMM performance degradation becomes very high (20 times larger error rate). This can be explained by the problem of speaker-impostor distribution separability. On the contrary, some of the pruned systems are more robust, even if their degradation is also considerable comparing to

Methods	FA%	FR%	EER%
	(6642 tests)	(1312 tests)	
HMM	0.24	0.26	**0.25**
BP	0.97	1.09	1.03
BP-ze	1.18	1.32	1.26
BP-dddisc	0.93	0.78	**0.85**
BP-spdisc	1.16	1.41	1.28
BP-ddcont	0.96	1.17	1.06
BP-ddcont-0.9	1.05	1.40	1.23
BP-ddcont-0.5	1.34	1.78	1.57

Table 1: Classifiers performances with a posteriori thresholds

the results of Table 1. It should be noted that the only system working close to the EER is the unpruned set of binary classifiers. For the pruned set, the lack of information on the output distribution scores could explain this unbalanced results.

Methods	FA%	FR%	TER%
	(6642 tests)	(1312 tests)	
HMM	1.52	9.22	**5.12**
BP	17.92	0.15	9.04
BP-ze	6.20	4.60	5.40
BP-spdisc	9.39	0.62	**5.01**
BP-dddisc	8.23	0.78	**4.50**
BP-ddcont	13.01	0.93	6.98
BP-ddcont-0.9	12.89	1.25	7.07
BP-ddcont-0.5	10.12	3.20	6.66

Table 2: Classifiers performances with a priori thresholds, TER is the Total Error Rate : (FA+FR)/2

5. CONCLUSIONS

The experiments described in this paper show that even if intrinsically the state-of-the-art HMM speaker verification algorithms are very efficient, when this kind of algorithms are used with a priori thresholds (which is always the case in real speaker verification applications), a simple decomposition into binary classifiers can be more robust. An explanation for this is the distance between speaker and impostor distributions. Indeed, this distance is larger in the case of pruned binary classifiers.

A good effect of the pruning techniques is the suppression of the classifiers that give a negative contribution to the speaker discrimination. This is not the case in HMM modelization, since the contribution of all the speakers is used for the world model.

Some improvements can still be added to the current system of binary classifiers. On the one hand, different learning algorithms can be used, like MLPs, oblique decision trees [9], or other well-suited 2-class separators. On the other hand, further attention can be given to the choice of the anti-speaker set, given its important role, which is to describe in the acoustic parameter space the whole non-speaker area.

Another issue that can be very interesting in the speaker verification domain is the input parameter selection. A work conducted by Charlet and Jouvet [3] has shown that not all the LPC parameters have a contribution to the speaker discrimination. The decision tree algorithm used here automatically performs such a selection for each pair speaker/anti-speaker and for each word (digit) pronounced.

6. REFERENCES

[1] Cambridge University Speech Group, Entropic Research Laboratories Inc., Cambridge. *HTK Hidden Markov Model Toolkit*, December 1993.

[2] P. Castelano, S. Slomka, and S. Sridharan. Telephone based speaker recognition using multiple binary classifier and gaussian mixture models. In *Proceedings ICASSP 97*, volume 2, pages 1075–1078, 1997.

[3] D. Charlet and D. Jouvet. Optimisation du paramétrage acoustique pour la vérification du locuteur. In *Acte JEP 96*, pages 399–402, 1996.

[4] D. Genoud, F. Bimbot, G.Gravier, and G. Chollet. Combining methods to improve speaker verification decision. In *Proceedings ICSLP*, pages 1756–1759, 1996.

[5] D. Genoud, J. Hennebert, and H. Melin. Polycost database. In *COST250 Minutes*, 1996. POLYCOST v1.0 homepage: `http://circwww.epfl.ch/polycost/`

[6] Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[7] J. Ross Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann, 1993.

[8] A.E. Rosenberg, C.H. Lee, and S. Gokoen. Connected word talker verification using whole word hidden markov model. In *Proceedings ICASSP 91*, volume 1, pages 381–384, 1991.

[9] P. E. Utgoff and C. E. Brodley. An incremental method for finding multivariate splits for decision trees. In *Proceedings of the Seventh International Conference on Machine Learning*, pages 58–65, Los Altos, CA, 1990. Morgan Kaufmann.