IMPROVING THE PERFORMANCES OF HIDDEN MARKOV MODELS FOR TEXT DEPENDENT SPEAKER VERIFICATION OR IDENTIFICATION

Frédéric. Quesne, Assistant Ir. and Henri. Leich, Professeur Dr. Ir.

Service de théorie des circuits et de traitement du signal
Faculté Polytechnique de Mons - Belgique

ABSTRACT - Two algorithms are described here, improving the results of hidden Markov models, one in a text dependent speaker identification process and another in a text dependent speaker verification process. An original method is also presented for the estimation of identification success rates with small databases.

INTRODUCTION

The aim of this paper is the description of a practical and efficient system for speaker identification or verification. In order to be practical, the system has to perform with a short and user friendly learning phase: not too much repetition of the chosen sentence, which must be short.
In such a case, only algorithms with time alignment of reference and test versions should give good results. Hidden Markov models are chosen because they give better results than dynamic time warping, with a longer learning phase but with a shorter time response during tests; this is very interesting, especially for the identification.
Neural networks have been avoided because they require too much learning computation time and data.

DATABASE

A 15 speakers database is used to test the system. Each speaker has pronounced 5 times consecutively the short French utterance "l'hydravion" for the model construction and later, 10 times over one month, the same sentence for the tests.

FEATURE EXTRACTION

The speech wave is band limited to 4.8 kHz and sampled at a 10 kHz sampling frequency. After suppression of the DC component by an elementary filter and emphasis the high frequencies by, a 30 ms Hamming window is applied every 10 ms. First to fourteenth predictor coefficients are extracted from each window by the Schur-Leroux-Gueguen algorithm and are transformed into cepstrum coefficients [6], using the following recursive relationship [10]:

$$c_1 = a_1$$

$$c_n = \sum_{k=1}^{n-1}(1-k/n)a_k c_{n-k} + a_n \qquad 1 \le n \le 14$$

where a_i and c_i are respectively the i^{th} order linear predictor coefficient and the i^{th} order cepstrum coefficient.
The cepstrum representation of each speech frame is completed with the delta cepstrum coefficients, computed as the difference between cepstrum coefficients at time t+2 and time t-2:

$$\Delta c_t = c_{t+2} - c_{t-2}$$

where Δc_t and c_t are respectively the i^{th} delta-cepstrum and cepstrum coefficients.
Finally, the parameter vector also takes the signal energy shape into account with the delta energy logarithm: $\Delta \ln E$, where E is the sum of all square values of the samples over the Hamming window and is computed as:

$$(\Delta \ln E)_n = \ln E_{n+2} - \ln E_{n-2}$$
$$= \ln(E_{n+2}/E_{n-2})$$

The second form of the formula shows that $\Delta \ln E$ is independent of the recording level.

BASIC MODEL DESCRIPTION

The number of states of the hidden Markov model varies from one to five times the phoneme number, with adjunction of an initial and a final state.
The transition probability matrix allows, given some state, to stay on the same state or to jump to one of the two next states (figure 1).

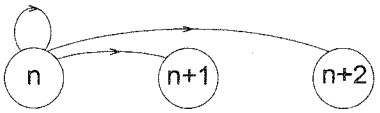


Figure 1 allowed transitions from state n

For the initial and final states, looping is forbidden.
The output probabilities are based on a continuous distribution of the parameter vectors instead of a vector quantization. Indeed, in the case of a general vector quantization for all the speakers, we loose too much information and in the case of a vector quantization for each speaker, too much data and too much computation time are required.
Two suppositions are made to estimate the output probabilities at each state of the model. After training of the hidden Markov model, each state should represent a stationary part of the speech wave. As a result, the parameter vectors associated to one state should, if the state number is sufficient, follow a Gaussian distribution. This assumption has been checked with the Kolmogorov-Smirnov method test [9] with only one state by phoneme. The examination of the covariance matrix of the parameter vectors associated to one state shows that the covariance between the i^{th} and the j^{th} element of the vectors are always negligible in comparison with the variance of the i^{th} and the variance of the j^{th} element. So, the multidimensional Gaussian distribution of the parameter vectors for one state can be decomposed into independent unidimensional Gaussian distributions. Each one is define by the mean value and the variance of the corresponding component of each vector associated to the state, after implementation of the Viterbi algorithm. Then, the output probability of a vector at a state can be estimated by the multiplication of the probability density of each of its components. This is, of course, only an image of the output probability since this value can be greater than 1.
This method of calculation implicates that the output probability of one vector at a state is independent of the previous vectors and their associated states.
The initial output probabilities (before the first implementation of the Viterbi algorithm, at the beginning of the learning phase) are based on an uniform distribution of the speech frames throughout the states of the model.
In order to limit the dynamic of the accumulated probability during the Viterbi algorithm implementation, all the probabilities are replaced by distances as follow:

$$\text{distance} = -\ln(\text{probability})$$

In the same way that the output probabilities are not really probabilities, these distances are not really distances since their value can be lower than 0. The accumulated distance, computed in this way and weighted by the number of speech frames of the test sentence, is used to discriminate the different speakers.

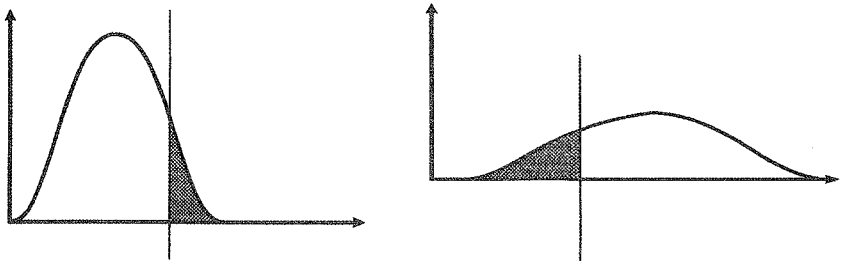
SUCCESS RATE ESTIMATION

The success rate estimation is different for verification or identification, but the methods described here for both cases suppose a Gaussian distribution of the discriminant distances calculated between models and test sentences.

Verification

In case of speaker verification, the global success rate is evaluated as the mean value of the success rates obtained with each speaker of the database (each model).
When a test sentence is compared to a model, it can be accepted or rejected if the calculated distance is respectively lower or greater than a predetermined threshold. It seems evident that two kinds of errors may occur: rejection errors, when the speaker corresponding to the model is wrongly

rejected, and acceptation errors, when impostors are accepted. Thus, there are two different success rates which both depend of the chosen threshold.
If we consider the distribution of the intra-speaker distances (distances between model and good speaker) to be Gaussian, the rejection error rate equals the surface lying under the Gaussian probability density on the right side of the threshold. In the same way, if we assume that the distribution of inter-speaker distances (distances between the model and impostors) is Gaussian, the acceptation error rate equals the surface lying under the corresponding Gaussian probability density on the left side of the threshold (see figure 2).



rejection error rate Figure 2 acceptation error rate

In order to obtain only one success rate estimation, which will be comparable with other results, the thresholds are chosen so that the acceptation and error rates are equal. This chosen threshold can be computed as:

$$\text{choosen treshold} = \frac{\sigma_1 m_2 + \sigma_2 m_1}{\sigma_1 + \sigma_2}$$

where m_1, m_2, σ_1 and σ_2 are respectively the mean value and the standard deviation of the intra and inter-speaker Gaussian probability density.

Identification

The purpose of identification process is to claim the identity of a speaker from his test utterance as the speaker to whom the model is the most compatible.
The estimation of the success rate in such a process is much more complex than for the verification process, even if we consider the distance distributions to be Gaussian. The only right way to do it is to count the number of good estimations and to divide the obtained results by the number of tests; but this method requires a big database to return an efficient value.
An intuitive method is described here to be performed with a little database. Though this method is not strict, it can give a good estimation of the success identification rate which could be comparable with other identification rates calculated in the same way.
For each test, the process returns a distance value between the test utterance and all the models of the database. Only the distance between the test utterance and the model corresponding to the good speaker and the distance between the test utterance and the nearest of the other models are retained. The subtraction of these two values gives a random variable that we suppose to follow a Gaussian distribution over all the tests. The identification success rate is then evaluated as the surface lying under this Gaussian probability density on the right side of the point 0.

SEGMENTATION

Model learning has to be based on a starting distribution of the speech vectors along the different states of the model. In the basis model described before, this distribution is uniform; but, in that case, during the successive iterations, output and transition probabilities are often converging to non optimum values, corresponding to a local minimum of the accumulated distance.
Then, it seems interesting to implement an algorithm which can deliver a distribution near the global minimum so that the Viterbi algorithm converges to the optimum distribution.

In order to obtain a good starting distribution, a method described by John S. BRIDLE and Nigel C. SEDGWICK in 1977 [4] can be used. The aim of this method is to decompose the speech wave into N segments (which will be in correspondence with the N states of the hidden Markov model) so that the quadratic error obtained by replacing the speech vectors by the mean vector of each segment is minimum.
In our case, this quadratic error is replaced by:

$$-\sum_{k=i+1}^{j} \ln(P_{\vec{m},\vec{\sigma}}(\vec{x}_k))$$

where $P_{\vec{m},\vec{\sigma}}(\vec{x}_k)$ is the output probability density of the k^{th} vector from the segment to which it is associated.
Let us notice (i, j) a segment containing L vectors from index i+1 to j. The problem is to find the indices of the junctions between all consecutive segments, to get the optimal segmentation: [(0, i_1) (i_1, i_2) ..... (i_{N-2}, i_{N-1}) (i_{N-1}, L)]. The quadratic error over one segment (i, j) is:

$$f(i,j) = -\sum_{k=i+1}^{j} \ln(P_{\vec{m},\vec{\sigma}}(\vec{x}_k))$$

where $P_{\vec{m},\vec{\sigma}}(\vec{x})$ is the Gaussian probability density defined by:

$$\text{mean vector} = \vec{m} = \frac{1}{j-i}\sum_{k=i+1}^{j} \vec{x}_k$$

$$\text{variance} = \frac{1}{j-i}\sum_{k=i+1}^{j} (\vec{x}_k - \vec{m})^2$$

The quadratic error over several consecutive segments is calculated as:

$$G[i_0, i_1, \ldots, i_M] = \sum_{k=1}^{M} f(i_{k-1}, i_k)$$

or recursively as:

$$G[i_0, i_1, \ldots, i_M] = G[i_0, i_1, \ldots, i_{M-1}] + f(i_{M-1}, i_M) \qquad (1)$$

We are interested by the sequence 0, i_1, i_2, ..., i_{M-1}, L which minimises $G[0, i_1, \ldots, i_{M-1}, L]$. Let us notice $F(n,l)$ the minimum value of $G[0, i_1, \ldots, i_{n-1}, l]$ (segmentation of the l first vectors into n segments). Using (1), a recursion can be made by the obvious relationship:

$$F(n,l) = \min_i [F(n-1,i) + f(i,l)] \qquad (2)$$

which allows to compute F(n,l) if the values of F(n-1,i) are known for all i. At the beginning, we can easily know the values of F(1,l) for all l and then compute the values of F(2,l), F(3,l),... for all l using (2), until F(N,L). A backtracking then allows to find the optimal boundaries between the segments.
This method requires a lot of computation time. Nevertheless, if we consider that a segment will contain a minimum of x and a maximum of y speech frames, the computation time can be reduced in a large proportion using the recursive relationship:

$$F(n,l) = \min_{l-y \le i \le l-x} [F(n-1,i) + f(i,l)]$$

The choice of x and y depends on the sentence length and the number of states.
Moreover, this method has not to deliver exactly the optimum distribution since the purpose is only to start the model learning by the Viterbi algorithm near the global optimum. So, we may reduce the computation time by considering the speech frames as inseparable groups of two or three frames.

TOTAL NORMALISATION

The accumulated distance delivered by the Viterbi algorithm, as described in the basis model, is normalised with the number of frames of the speech wave taking into account the differences of length between test and learning utterances.
However, it does not take into account the differences of rhythm (the beginning may be longer and the end shorter for instance).

The idea of total normalisation is to perform a perfect alignment of the distance accumulated during each state of the Markov model with the mean number of speech frames associated to this state at the end of the learning phase, so that each state contributes in the same proportion (than in the learning phase) to the discriminant distance, which has the same signification for all possible rhythms of the sentence. The normalisation is performed as follows:

$$D_{discr} = \sum_{i=1}^{N} d_i \frac{m_i}{n_i}$$

where D_{discr} is the discriminant distance, d_i is the distance accumulated by state i, n_i is the number of speech frames in state i and m_i is the mean number of frames in the i^{th} state at the end of the learning phase.
A backtracking after the Viterbi algorithm implementation is needed to find all n_i. As a result, this method requires a little bit more computation time than the basis method.
An other great advantage of this method is to avoid acceptations of impostor utterances containing only one phoneme, such as "hiiiii" in place of "l'hydravion".

RESULTS AND CONCLUSIONS.

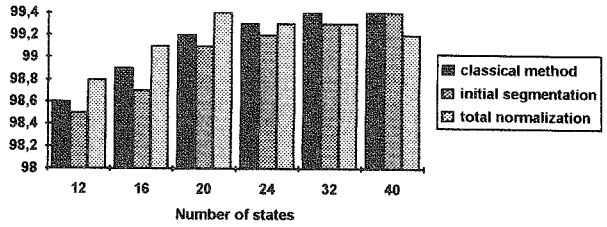


Figure 3 verification success rate

It can be seen from the results of figure 3 that an initial segmentation by the Bridle and Sedgwick algorithm doesn't provide better results for verification. Indeed, if a model corresponds to a local optimum, the distances are greater than for a global optimum, and it seems that distances with good speaker or impostor utterances are amplified in the same proportion, so that (after adaptation of the threshold) the results have not to suffer from it.

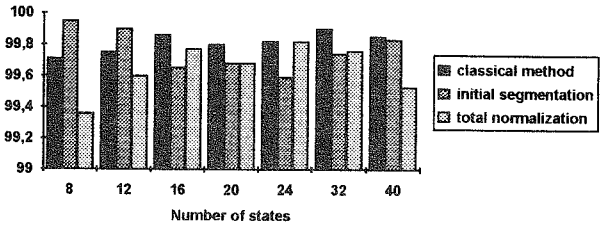


Figure 4 identification success rate

On the other hand, for an identification process, where a sentence is compared to all models and associated to the nearest one, a model corresponding to a global optimum is privileged in comparison with a model corresponding to a local optimum (the distances obtained with different test sentences are smaller in average). In such a case, an adequate initial segmentation improves the results in good proportion (figure 4). This improvement occurs only for a reduced number of states (when there is a significant number of speech frames associated with each state). The success rate is greater for an

eight state model with initial segmentation than for all number of states without initial segmentation (great gain on a memorisation and computation time point of view).
We can see on figure 4 that a total normalisation doesn't provide efficient results in a speaker identification process. On the other hand, in a verification process, where a comparison is done with a threshold, the better results obtained with a total normalisation (figure 3) underline the importance of making a correspondence between the rhythm of the test utterance and the rhythm of reference utterances which have given the threshold. This improvement is only observed for a reduced number of states in the model because the total normalisation algorithm is disturbed by the empty states that appears when the number of states grows. The maximum success rate is thus obtained with a fewer number of states with total normalisation than with the classical method.

REFERENCES

[1] Aaron E. Rosenberg, Chin-Hui Lee, Frank K. Soong, "Sub-Word Unit Talker Verification Using Hidden Markov Models", ICASSP 1990, p.269-272.

[2] Aaron E. Rosenberg, Chin-Hui Lee, Sedat Gokcen, "Connected Word Talker Verification Using Whole Word Hidden Markov Models", ICASSP 1991, p.381-384.

[3] Michael Savic and Sunil K. Gupta, "Variable Parameter Speaker Verification System Based on Hidden Markov Modeling", ICASSP 1990, p.281-284.

[4] John S. Bridle and Nigel C. Sedgwick, "A Method for Segmenting Acoustic Patterns with Applications to Automatic Speech Recognition", ICASSP 1977, p.656-659.

[5] Naftali Z. Tishby, "On the Application of Mixture AR Hidden Markov Models to Text Independent Speaker Recognition", IEEE Transactions on Signal Processing, vol.39, n°3, March 1991, p.563-570

[6] Ronald W. Schafer and Laurence R. Rabiner, "Digital Representations of Speech Signals", Proc. of the IEEE, vol.63, n°4, April 1975, p.662-677.

[7] F.K. Song and A.E. Rosenberg, "On the Use of Instantaneous and Transisional Spectral Information in Speaker Recognition", ICASSP 1986, p.877-880.

[8] Sadaoki Furui, "Speaker-Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum", IEEE Transactions on Acoustics, Speech and Signal Processing, vol.ASSP-34, n°1, February 1986, p.52-59.

[9] L. Lebart, A. Morineau and J.-P. Fénelon (1982), "Traitement des données statistiques", dunod.

[10] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification", J. Acoust. Soc. Am., Vol. 55, N° 6, June 1974, p.1304-1312.