A HMM/SVM Hybrid Method for Speaker Verification

Răstoceanu Florin and Diana Militaru
Ministry of National Defense
Military Equipment and Technologies Research Agency
Bucharest, Romania

Abstract—In this paper, we describe an application of speaker verification using Romanian vowels as speaker’s models in case of a small Romanian language database. Vowels models are obtained with continuous HMMs using re-training of the vowels models for every speaker. Afterwards the models are classified with the powerful technique named SVM.

Keywords – speaker verification; Romanian language; SVM; HMM; PLP

I. INTRODUCTION

In the recent years the technology for speaker verification or call authentication has received an increasing amount of attention. There is a great interesting in developing and performance increasing of automatic speaker verification applications because of the advantages offered comparing to other biometrical methods. The most important aspects are that a speech processing application with SVM (Support Vector Machines) has a low cost implementation and is also very easy to use.

In speaker verification, a person makes an identity claim (e.g., by entering a number or presenting a smart card). Because of that speaker verification is a decision-making process that for a given speech sample and a claimed identity, the verification system gives a value for acceptance or rejection. The process consists of a measurement of the input speech sample with pre-stored templates, and a comparison based on the obtained measurement and a pre-defined threshold.

In this paper we propose a method to use the capabilities of two important technologies in speech processing: HMM (Hidden Markov Models) and SVM.

II. HIDDEN MARKOV MODELS (HMM)

The HMMs finite automata are a given number of states; passing from one state to another is made instantaneously at equally spaced time moments. The hidden Markov model incorporates the knowledge about feature constellation corresponding to each of the distinct phonetic units to be recognized. At every pass from one state to another, the system generates observations, two processes taking place: the transparent one represented by the observations string (features sequence), and the hidden one, which cannot be observed, represented by the state string.

In speech recognition, the left-right model (or the Bakis model) is considered the best choice. For each symbol, such a model is constructed; a word string is obtained by connecting corresponding HMMs together in sequence. For very limited tasks there are preferred phonetic models based on monophones (which are phonemes without context), because the phonemes are easy generalizable and of course also trainable. Monophones constitute the foundation of any training method but in real speech the words are not simple strings of independent phonemes, because each phoneme is affected through the immediately neighboring phonemes by co-articulation.

The basic parameters of a HMM model are:

- \(N\) – The number of states \(S = \{s_1, s_2, ..., s_N\}\); a state to a certain time is denominated as \(q_t\), \(q_t \in S\);
- \(M\) – The number of distinct symbols observable in each state. The observations are \(O = \{o_1, o_2, ..., o_M\}\), one element \(o_t\) from \(V\) is a symbol observed at moment \(t\);
- \(A\) – The transition matrix containing the probabilities \(a_{ij}\) of the transition from state \(i\) in state \(j\);
- \(B\) – Matrix of observed symbols in each state of the model, \(b_j(k)\), represents the probability to observe a symbol \(v_k\) in state \(j\);
- \(\Pi\) – The matrix of initial probabilities

In a compact mode a discrete HMM can be symbolized as \(\lambda = (\Pi, A, B)\).

In our experiments we used continuous models in which each observation probability distribution is represented by a mixture of Gaussian densities. The essential problem is to estimate the means and variances of a HMM in which each state output distribution is a single component Gaussian:

\[
b_j(o_t) = \frac{1}{\sqrt{(2\pi)^{M}|\Sigma|}} \exp\left(-\frac{1}{2} (o_t - \mu_j)^T \Sigma_j^{-1} (o_t - \mu_j) \right) \tag{1}\]

If \(\xi_j(t)\) is the probability to be in the state \(j\) at the time \(t\), then the mean and variance are:

\[
\hat{\mu}_j = \frac{\sum_{t=1}^{T} \xi_j(t) o_t}{\sum_{t=1}^{T} \xi_j(t)} \tag{2}
\]
Based on HMMs the statistical strategies have many advantages, among them being recalled: rich mathematical framework, powerful training and decoding methods, good sequence handling capabilities, flexible topology for statistical phonology and syntax. The disadvantages lie in the poor discrimination between the models and in the unrealistic assumptions that must be made to construct the HMMs theory, namely the independence of the successive feature frames (input vectors) and the first order Markov process.

III. SUPPORT VECTOR MACHINES (SVM)

The foundations of Support Vector Machines are based on Structural Risk Minimization (SRM) principle, which minimizes an upper bound on the generalization error, as opposed to Empirical Risk Minimization (ERM), which minimizes the error on the training data. This technique can be successfully used in pattern recognition and information retrieval tasks. The main idea in training a SVM system is to find a hyperplane as a decision boundary between two classes of objects.

A. Linear Case

Consider the problem of separating the set of $N$ training vectors ${\{(x_1, y_1), \ldots, (x_n, y_n)\}}$, $x \in \mathbb{R}^m$, belonging to two different classes $y_i \in \{-1, 1\}$. The goal is to find the linear decision function $D(x)$ and the separating plane $H$.

\[
H : \langle w, b \rangle + b = 0 \tag{4}
\]

\[
D(x) = \text{sign}(w \cdot x + b) \tag{5}
\]

where $b$ is the distance of the hyperplane from the origin and $w$ is the normal to the decision region.

Let the “margin” of the SVM be defined as the distance from the separating hyperplane to the closest two classes. The SVM training finds the optimal separating hyperplane. The optimal hyperplane is the one with the maximum margin. The margin is equal to $2/|w|$. Once the hyperplane is obtained, all the training examples satisfy the following inequalities.

\[
x_i \cdot w + b \geq +1 \quad \text{for } y_i = +1 \tag{6}
\]

\[
x_i \cdot w + b \geq -1 \quad \text{for } y_i = -1 \tag{7}
\]

We can summarize the above procedure to the following [1]:

\[
\text{Minimize } L(w) = \frac{1}{2} |w|^2 \tag{8}
\]

Subject to $y_i (x_i \cdot w + b) \geq 1, \quad i = 1, 2, \ldots, N$

B. Non-linear Case

Real-world classification problems typically involve data that can only be separated using a nonlinear decision surface. Optimization on the input data in this case involves the use of a kernel-based transformation who transform data in a higher dimensional space (feature space) in which data are linear separable.

\[
k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \tag{9}
\]

IV. HYBRID SPEAKER VERIFICATION METHOD

One significant drawback in SVMs is that, they are inherently static classifiers — they do not implicitly model temporal evolution of data. HMMs have the advantage of being able to handle dynamic data with certain assumptions about stationarity and independence. Taking advantage of the relative strengths of these two classification paradigms we have developed a hybrid SVM/HMM system.

The hybrid HMM/SVM method is composed from three stages.

In first stage we use HMM to obtain a several models from each Romanian vowels. From every speaker are created a set of such models. The models are obtained in two steps (figure 3). First, the training data is uniformly segmented, all of the phone models are initialized to be identical and have the same means and variances equal to the global speech mean and...
variance. In the training with global initialization, there are obtained 25 multiple mixture component Gaussian distributions of monophones started with this initial data.

![Diagram](image1)

Figure 3. The statistic method used to obtained the phoneme (vowels) supervector. HMM-1, HMM-2, HMM-3 are the models used to form the phoneme (vowels) supervectors

After that it is performed a re-training with global initialization [2]. The initial set of models is extracted apart from the multiple mixtures for each speaker, but the means and variances are the same for every phoneme. The parameters extraction consists in repeated alignments for observation segmentations and parameters recalculation depending on convergence factor and the maxim number of estimation cycles.

Afterwards the initial set of models are calculated, the data are training in the same way like in the first stage.

For these experiments there are used three types of HMMs (figure 3):

- HMM-1, obtained only with 2 parameters re-estimation of the initial parameters;
- HMM-2, obtained from the HMM-1 with silence models fixed and followed by another 2 parameters re-estimation;
- HMM-3, obtained from HMM-2 with the training data re-aligned and followed by another 2 parameters re-estimation.

In the second stage we create the phoneme (vowels) supervectors [3], corresponding to each speaker, from the HMM models. The phoneme (vowels) supervectors are the results of HMMs based on the left-right model with 5 states and consist in means and variances (figure 4) or only means. The parametrization used in this paper is PLP (perceptual linear parameterization) with delta dynamics parameters and log-energy.

![Diagram](image2)

Figure 4. Method to obtain phonemes supervectors

In the third stage we use SVM to classify the supervectors according to the speakers (figure 5). In the training process from each speaker were created 7 SVM models corresponding with the 7 vowels from Romanian language. For creating each model are used the supervectors extracted from negative speakers (the other nine speakers) and a collection of feature extracted from true speaker’s vowels. In the verification process were also used a collection of features extracted from negative and positive speaker’s vowels. Each of the feature vectors in the test vowels is matched with a speaker template, according to the given speaker identify, to produce a score. The equal error rate (EER) is used to measure the performance in all our evaluations.

![Diagram](image3)

Figure 5. Using SVM for speaker verification

V. EXPERIMENTAL RESULTS

The evaluation is carried out on a small database with 10 speakers (2 female and 8 male). The database was recorded in a laboratory and a number of 50 different sentences were spoken by each speaker. The HMM models are obtain with ProtoLOGOS speech recognition system and for SVM have been used MATLAB v7.1 and LIBSVM, a library for support
vector machines classification and regression, developed by National Taiwan University.

Experiments were carried out to compare the method performances using different types of kernel functions, HMM models and vowels.

In the first stage the comparison are made against the three types of HMM models and kernel functions. For this purpose are used the HMM models with means and variances and HMM models with only means. As a kernel functions are used RBF and Polynomial (grade 3 and 4). The results are presented in figure 6, tables I and II.

![Figure 6. Mean EER (for all speakers and vowels) for different kernels and HMM models](image)

<table>
<thead>
<tr>
<th>Methods</th>
<th>Mean EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pol3</td>
<td>12.90</td>
</tr>
<tr>
<td>Pol4</td>
<td>13.12</td>
</tr>
<tr>
<td>RBF</td>
<td>13.52</td>
</tr>
<tr>
<td>Pol3 (only means)</td>
<td>13.78</td>
</tr>
<tr>
<td>Pol4 (only means)</td>
<td>13.42</td>
</tr>
<tr>
<td>RBF (only means)</td>
<td>14.00</td>
</tr>
</tbody>
</table>

We can observe from this that the kernel function with the best results is Polynomial (grade 3), but good results are obtain too, with Polynomial (grade 4) and RBF kernel. Using only means from HMM models to form SVM supervectors do not improve the performance. The HMM model who obtain the best results is HMM-1. One explanation possible is that HMM-1 offers a proper number of parameter re-estimations (after the fixing the silent model) of those models extracted from the initial training.

![Table I. Mean EER (for all speakers and vowels) for different kernels and HMM models](image)

Using the results mentioned above, in the second stage, the comparisons are made using only Polynomial (grade 3) kernel and HMM-2 model. In this stage the experiments show what are the vowel with the best results (figure 7), and using this vowel, what are the results obtained by each speaker (figure 8).

![Figure 7. Mean EER (for all speakers) for Romanian vowels obtained with Polynomial (grade 3) kernel and HMM-2 model (means and variances)](image)

<table>
<thead>
<tr>
<th>HMM models</th>
<th>Mean EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM-1</td>
<td>13.33</td>
</tr>
<tr>
<td>HMM-2</td>
<td>14.03</td>
</tr>
<tr>
<td>HMM-3</td>
<td>14.03</td>
</tr>
</tbody>
</table>

The best results are obtained with vowel “a” followed by vowels “o”, “e” and “i”. The vowels “ă”, “u” and “î” are less discriminative obtaining a very big EER (figure 7).

![Figure 8. EER for speakers obtained with Polynomial (grade 3) kernel and HMM-2 model for vowel “a”](image)

VI. CONCLUSIONS

In this paper we tried to do a HMM/SVM hybrid model for speaker recognition with ProtoLOGOS system, LIBSVM and a database, recorded in a laboratory, with 10 speakers and 500 sentences. For that purpose, for HMMs we used global initialization re-training, PLP parameterization with delta parameters and log-energy, means and variances, or only means. For SVM we used RBF and polynomial kernels (grade 3 and 4).

The differences between the error rates both for speakers and for vowels are big. For vowels, one supposition is that the database is small and some vowels are better training than others.

Certainly, all these results can be improved using a bigger professional database.

REFERENCES

