HMM, recognition, speech, disorders

Marek WIŚNIEWSKI[1], Wiesława KUNISZYK-JÓŹKOWIAK[1],
Elżbieta SMOŁKA[1], Waldemar SUSZYŃSKI[1]

IMPROVED APPROACH TO AUTOMATIC DETECTION OF SPEECH DISORDERS BASED ON THE HIDDEN MARKOV MODELS APPROACH

In the work algorithms commonly utilized in continuous speech recognition systems were applied to detection of speech disorders. The used algorithms were briefly described and the final method of speech disorders detection was presented. The article includes the results of the short test performed in order to check the effectiveness and accuracy of the method. The aim of the test was detection and classification of fricative phonemes prolongation one of the most common speech disorders in the Polish language. It is worth emphasizing that this method enables detection of a category of speech disturbance (e.g. fricative, nasal, vowels, etc… prolongation), but also provides the information about a specific phoneme being disturbed.

1. INTRODUCTION

The therapy of stuttering people is usually based on exercises that include reading and talking. The selection of proper exercises should be based on the actual kind and level of speech disturbance in patients. It consist in determination of occurrence frequency of every kind of disorders as well as measurement of their duration times. It is also very important to detect which parts of speech the patient has a problem with, especially with which phonemes disturbances are associated. Thus the application of automatic diagnosis should enable detection of disturbances and phonemes associated with.

The natural way of gaining information about phonemes in speech is utilization of the Hidden Markov Models and proper algorithms commonly used in speech recognition systems. The HMMs can be used as a simple pattern recognition method as described in [1] but the more adequate and simultaneously much more complicated method is that used in continuous speech recognition systems (CSR).

2. RECOGNITON USING HMM

The HMMs are stochastic models that gained great significance particularly in speech recognition systems [2,3,4]. The HMM is a kind of extension of the Markov chain. Markov models are characterized by the state-transition probability matrix (A) and the initial state probability matrix (π). Each state corresponds to an observed event of the modeled process. Having properly modeled matrixes one can estimate probabilities of any sequence of events. What is important, in the Markov chain each state corresponds to the directly (deterministically) observed event.

In the HMM there was introduced a (non-deterministic) process that generates observations in any state (so it can be considered as a double-embedded stochastic process with an underlying stochastic process not directly observable). This underlying process can only be probabilistically associated with another stochastic process producing the sequence of observable features [5].

2.1. HMM TYPES

A form of the HMM model can be different and depends on the type of observation space: discrete or continuous. If the observation space is discrete and finite, then the emission matrix B can be also discrete. In that case, it includes emission probabilities of every observation symbol by every state of a model directly. There is also possibility to map a continuous observation space into discrete using any

[1] marek.wisniewski@poczta.umcs.lublin.pl

clustering/quantization algorithm. But then there is a risk of a negative influence of the inherent quantization errors on the modeling quality.

For modeling a continuous output space, the best solution is a continuous output probability form. The most common are multivariate Gaussian mixture density functions because they can approximate any continuous space distribution. In such a case, the emission probability is defined as:

$$b_j(\mathbf{o}_t) = \sum_{k=1}^{M} c_{jk} N(\mathbf{o}_t; \boldsymbol{\mu}_k, \Sigma_k) \tag{1}$$

$$N(\mathbf{o}_t; \boldsymbol{\mu}_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_k|}} \mathrm{e}^{-\frac{1}{2}(\mathbf{o}_t - \boldsymbol{\mu}_k)^{\mathrm{T}} \Sigma^{-1} (\mathbf{o}_t - \boldsymbol{\mu}_k)} \tag{2}$$

where:

- j – model state index,
- $\mathbf{o}_t$ – observation vector parameters at time t,
- n – dimension of the observation vector,
- M – number of Gaussian mixtures,
- c_{ik} – coefficient value for state j and mixture k,
- $\boldsymbol{\mu}_k$ – mean vector for mixture k,
- $\boldsymbol{\Sigma}$ – covariance matrix for mixture k.

$$\Sigma^{-1} = \begin{pmatrix} \frac{1}{\sigma^2} & 0 & 0 \\ 0 & \frac{1}{\sigma^2} & 0 \\ 0 & 0 & \frac{1}{\sigma^2} \end{pmatrix} \tag{3}$$

$$\mu_i = \frac{\sum_{n=1}^{N} x_i}{N} \tag{4}$$

$$\sigma_{ii}^2 = \frac{\sum_{n=1}^{N} (x_i - \mu_i)^2}{N-1} = \frac{\sum_{n=1}^{N} x_i^2 - \left(\left(\sum_{n=1}^{N} x_i \right)^2 \Big/ N \right)}{N-1} \tag{5}$$

If parameters of observation vectors are not correlated or their correlation is very small (for example MFCC parameters), then the covariance matrix is diagonal (3,4,5) so it perfectly reduces the computational time. If HMMs are fully continuous, then each model needs to have calculated mixture parameters individually i.e. the observation space needs to be divided into the defined number of clusters and then the mean and variance values of those clusters need to be calculated. In the case of modeling processes that require a lot of models, the training and recognition time can be too long. In order to reduce the computational time and simultaneously keep a good level of modeling accuracy one can apply the co-called semicontinuous or tied-mixture HMMs. In the semicontinuous HMM the mixture density functions are tied together across all models and only mixture weights are specific to each model.

In the case of continuous speech recognition systems, the basic speech entity is a phoneme. The articulation of a phoneme can by divided into three phases: initiation, essential, finalization and usually each model state is related to one of them [6].

The most common model type used in continuous speech recognition systems (CRS) is a left-right. If the recognition unit is a phoneme, the natural choice is a three-state left-right model.

2.2. PHONEME RECOGNITION

The recognition process with the HMM depends on the modeled process. If the process is described by only one model, then required algorithms are simple. The number of states of the model is selected on the basis: one event one state. Models are trained using the well known Forward-Backward (or Baum-Welch) algorithm. For evaluation or recognition the simple Viterbi algorithm can be used. As a result, one can obtain the best state sequence of the model for a given observation. In the case when the modeled process is more complex, more than one model is used. Then the used algorithms are more complicated, especially when continuous speech is modeled..

In the continuous speech recognition usually one phoneme is modeled by one left-right HMM. For speech recognition there is a need to create a model for every phoneme and any other sounds that appear in a human speech. Having such a database, recognition can be performed using search algorithms. In that case, a search algorithm should give the best models sequence (state sequence is not important here) that is best matched to the input sound.

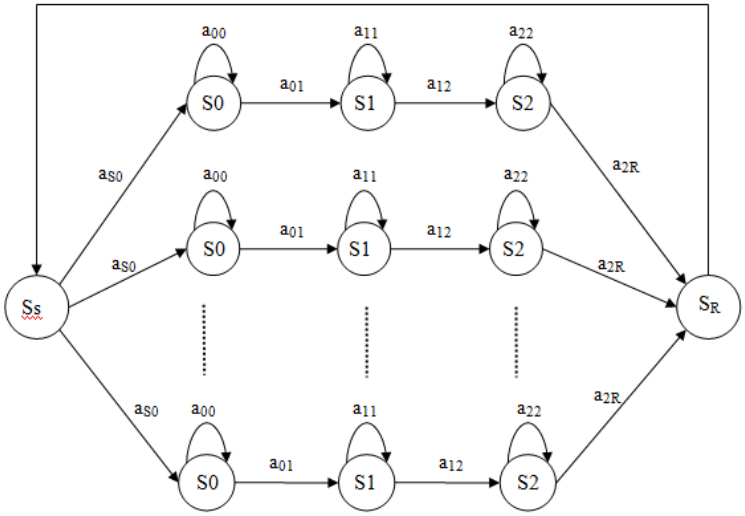


Fig 1. The model structure for the phoneme recognition.

Figure 1 presents the model scheme used in continuous speech recognition. Of all models in the database, one structure is created. All models are concatenated by the additional, virtual states s_S and s_R. The structure determines permited transitions between all models and their states.

In the work the properly modified Viterbi algorithm was used for connected phoneme recognition.

2.3. VITERBI ALGORITHM FOR CONNECTED PHONEME RECOGNITION

The basic Viterbi algorithm allows to find the best state sequence for one model. It can also be useful for recognition of isolated words as well as for training.

In the case of the connected word recognition, the extended version of the algorithm is employed. Besides the state sequence finding the best model sequence is also required.

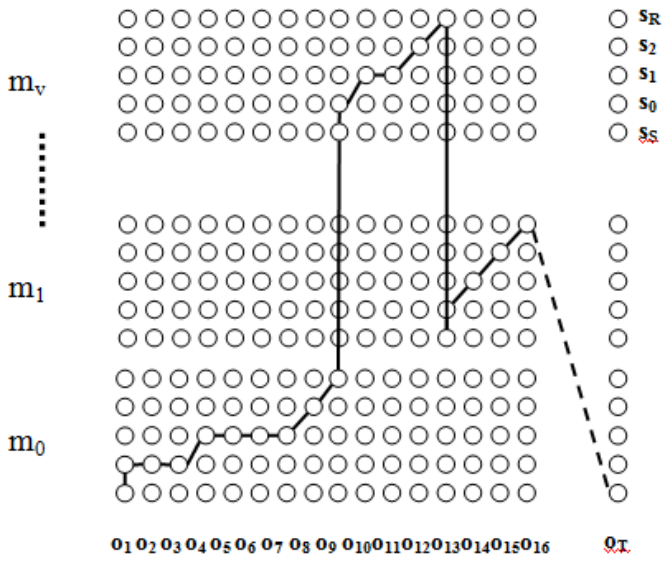


Fig. 2. The algorithm for phoneme decoding.

In figure 2 the recognition issue is presented. The problem is to find the best path across the grid points. The vertical axis represents the set of models denoted by $M=\{m_1, m_2,\ldots,m_V\}$ and the horizontal axis represents the input observation sequence denoted by $O=\{o_1, o_2,\ldots,o_T\}$. Each grid point represents a state of a suitable model at each time step . In our case, every model has five states where the three are emitting (s_1, s_2, s_3) and the two are virtual or non-emitting (s_S, s_R). Virtual states are necessary to properly construct a search algorithm and they not consume any time step so they are not associated with any observation chunk o_t.

The first step is probabilities calculation for all grid points (i.e. probability of being in a state of a model at the time t) considering some constraints (related to the permitted transitions between states). Let signify a state as a $s(\theta,v)$ – where θ is a state number of the model m_v, and the set of its predecessor states by $\Xi(s(\theta,v))$.

The probability of being in a state θ of the model m_v at the time t can be achieved from equation (6).

$$\delta(\mathrm{s}(\theta,\mathrm{v}),\mathrm{t}) = \mathrm{b}(\mathrm{s}(\theta,\mathrm{v}),\mathrm{t}) + \max_{(i,k)\in\Xi(s(\theta,v))}\left\{d(\mathrm{s}(i,\mathrm{k}),\mathrm{s}(\theta,\mathrm{v})) + \delta(\mathrm{s}(i,\mathrm{k}),\mathrm{t}\text{-}1)\right\} \tag{6}$$

where:

$b(s(\theta,v),t)$ - emission probability for the observation chunk o_t from the state s_θ of the model m_v,

$\delta(s(i,k),t-1)$ - probability of being in a state i of the model m_k at the time $t-1$.

As the virtual state s_S does not emitt and has only one predecessor s_R (figure 1), it can be merged with the state s_R, (so s_S and s_R have the same probability value). The probability in the state s_R at the time t is achieved from equation (7) and has to be calculated after determination of all other states probabilities at that time.

$$\delta(\mathrm{s}_S,\mathrm{t}) = \delta(\mathrm{s}_\mathrm{R},\mathrm{t}) = \max_{(i,k)\in\Xi(s(\theta,v))}\left\{\delta(\mathrm{s}(i,\mathrm{k}),\mathrm{t})\right\} \tag{7}$$

As soon as all grid points are calculated the best state sequence and the best model sequence can be read using a backtracking procedure. From the grid points at time $t=T$ one must select that with the greatest probability. Next going back across the grid, the best path is found. The sequence of model names read from that path is the recognition result.

3. RECOGNITION PROCEDURE

3.1. AUDIO SAMPLE PARAMETRIZATION

The parameters of audio recordings were as follows: samples frequency: 22050Hz, the amplitude resolution: 16 bits.

The acoustic signal were converted to the common set of parameters: Mel Frequency Cepstral Coefficients (MFCC). As an additional parameter, signal energy was used. The process of determining the signal parameters was as follows:

- pre-emphasis filtering x`(n)=x(n-1) – 0.97x(n),
- division into frames of 512 samples' length,
- calculation of frames energy,
- Hamming window,
- Fast Fourier Transform calculation,
- transition of the frequency values to the mel scale according to the formula [7,8]: Fmel=2595*log(1+F/700),
- frequency filtering by 20 triangular filters,
- calculation of 19 MFCC parameters,
- MFCC parameters calculation using the formula [9]:

$$MFCC_n = \sum_{k=1}^{K} (\log S_k) \cos\left[n(k-0.5)\frac{\pi}{K} \right], \qquad for \quad n = 1...N \tag{9}$$

where:

N – required number of MFCC parameters,
S_k – power spectrum coefficients,
K – number of filters.

Power spectrum coefficients S_k values for each filter were determined according to:

$$S_k = \sum_{j=0}^{J} P_j A_{k,j} \tag{10}$$

where:

J - subsequent frequency ranges from the FFT analysis,
P_j - average power of an input signal for j frequency,
$A_{k,j}$ - k-filter coefficient.

3.2. SEMICONTINUOUS CODEBOOK

For the codebook generation a proper utterance was chosen, which covered the entire acoustic space used in the tests. In our case, the tests were performed using the recordings from two persons, so for the codebook generation the accessible recordings from them were utilized. Another issue was the choice of the best codebook size. The best way of selecting the size is the experimental one. During some tests, it appeared that a 512-element codebook was proper. Smaller codebooks gave poorer results (smaller "resolution"), however, the larger ones were not tested because of computational expenses.

The codebook was generated using the "k-means" algorithm. The selected audio sample (length over 4 min 30 sec) parameterized and 512 centriods and variances were calculated. As a distance measure, the weighted Euclidean distance was used:

$$d_{x,y} = \sqrt{\sum_{i=1}^{N} (x_i - y_i)\Lambda(x_i - y_i)} \tag{11}$$

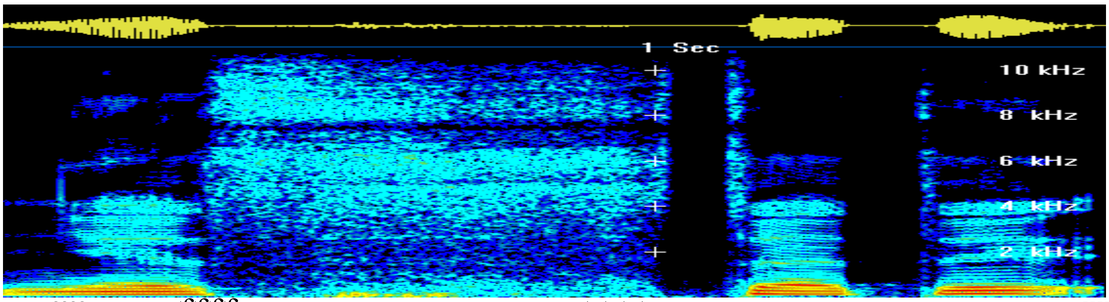
where: $d_{x,y}$ – Euclidean distance between N - dimensional vectors X and Y, Λ - diagonal matrix of scaling factors.

3.3. RECOGNITON EXAMPLES

According to the studies [10] phoneme duration times in the Polish fluent speech range from 100 ms to 200 ms. The duration times between 200 ms and 300 ms can be interpreted in various ways. They can indicate prolongation but not in every case.

Longer times, however, are interpreted unambiguously as prolongation.

The detection idea of phoneme prolongation was to generate a phoneme transcription from examined utterances and then check whether there are some long sequences of phonemes (lasting over 200 ms). Such detected sequences could be treated as prolongation. Unfortunately, the tests showed, that it is impossible to recognize suitably long sequences of the same phoneme. Such sequences always included several different phonemes (but usually in the same category) so there is no simple way to check the length and make the decision. One can consider two reasons – the first: too small training material and the other, more important, the nature of fricative prolongation.



a) ɲɲɲɲɲffffɪɪɪɪaaaaʧʧʧʧeeeessssssssssssssssssssssssssssssssʦʦʦʦsssssssssssssssssssssssssssssppppttttppppllllеeeeeeeeevv vvppppppssss ʃʃʃʃ ɪɪɪɪɪɪeeeeeee

b) ______ffff__________ʧʧʧʧ____ssssssssssssssssssssssssssssssʦʦʦʦssssssssssssssssssssssssssss___________________ ______vvvv______ssss ʃʃʃʃ _________________

c) ____________________________ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss_____________________ ___________________________________

Fig. 3. The recognition example of the Polish word “niessstety” [ɲ iessstei]; a) spectrogram [11], b) transcription obtained by the author’s ‘Hmm’ application , c) transcription after deleting everything except fricatives d) the final result – prolonged phoneme ‘s’.

As a solution of that problem, the new evaluation way of the recognition results were applied as shown in figure 3. If, a long sequence of models representing only fricatives appeared in the transcription of a tested sample, it would be considered as prolongation. To determine which phoneme was prolonged, duration of all phonemes was examined. The phoneme most frequently appearing in the sequence was considered as the prolonged one.

Figure 2 shows the recognition example of the word “nie’sss’tety” and its spectrogram. The prolongation, as read from figure 2d, started at 300 ms and was about 700 ms long. The prolonged phoneme was ‘s’. The recognized prolongation start time and the duration time exactly matched to the spectrogram and it was also properly determined as phoneme ‘s’.

In the tests some utterances were analyzed. The set consisted of 20 utterances which included the total number of 25 prolonged phonemes: f-1, sz-2, s-20, w-1, z-1. They were not used in the model parameters estimation.

The recognition database consists of 37 context-independent models [12]. Their parameters were estimated using about 500 short utterances recorded from two persons.

For the recognition evaluation, the two common percentage parameters were used (12,13):

$$\text{Correctness:} \quad C = \frac{H}{N} * 100\% \tag{12}$$

$$\text{Accuracy:} \quad A = \frac{H - I}{N} * 100\% \tag{13}$$

where:

H - number of correctly detected prolongations,
N - total number of prolongations,
I - incorrectly detected prolongations.

Of 25 prolongations 20 were properly recognized, 5 were not recognized and, what is the most important, none was wrongly recognized. So the correctness was exactly 80% which is a very good result.

The Errors in recognition were due to the inaccurate transcription. There can be distinguished two main reasons for that. The first were the context-independent monophone models used in the experiment – they are less effective than the context-dependent ones [5]. The other is the nature of fricative prolongations – they often include strange sounds that lead to errors. When in the transcription, where prolongation was expected, phonemes other than fricatives appeared, it was classified as an error. Also the error was when the recognized phoneme of a long sequence of fricatives did not match the actually prolonged phoneme.

Another issue are words that naturally have long sequences of fricatives, for example the Polish words: “szczygieł” [ʃ ʧ i ʃ ew], “szczebel” [ʃ ʧ ebel], “chrząszcz” [xʃ owʃ ʧ], etc. In that case the method can give false positive recognition. This can be eliminated for example by the speech therapist by selection of proper exercises during the diagnosis.

4. SUMMARY

The computer application (Hmm) based on the presented algorithms gave very promising results. The recognition correctness of 80% is hopeful for developing the complete system for speech disorders diagnosis. In comparison to the speech recognition systems, it can be much simpler because there is no need to integrate any language constraint or dictionaries but it is sufficient to provide information about prolonged phonemes.

REFERENCES

[1] WIŚNIEWSKI M., KUNISZYK–JÓŹKOWIAK W., SMOŁKA E., SUSZYŃSKI W., Automatic detection of prolonged fricative phonemes with the Hidden Markov Models approach, Journal of Medical Informatics and Technogies vol. 11/2007, Computer System Dept. University of Silesia.
[2] http://cmusphinx.sourceforge.net/wiki/tutorialconcepts.
[3] http://htk.eng.cam.ac.uk/docs/docs.shtml.
[4] http://julius.sourceforge.jp/book/Julius–3.2–book–e.pdf.
[5] HUANG X., ACERO A., HON H., Spoken Language Processing, Prentice Hall PTR, New Jersey, 2001.
[6] DELLER J. R., HANSEN J. H. L., PROAKIS J. G., Discrete–Time Processing of Speech Signals, IEEE, New York 2000.
[7] WAHAB A., SEE NG G., DICKIYANTO, R., Speaker Verification System Based on Human Auditory and Fuzzy Neural Network System, Neurocomputing Manuscript Draft, Singapore.
[8] PICONE J.W., Signal modeling techniques in speech recognition, Proceedings of the IEEE, 1993, 81(9): pp. 1215−1247.

[9] SCHROEDER, M.R., Recognition of complex acoustic signals, Life Science Research Report, T.H. Bullock, Ed., (Abakon Verlag, Berlin) Vol. 55, 1977, pp. 323−328.
[10] SUSZYŃSKI W., Komputerowa analiza i rozpoznawanie niepłynności mowy, rozprawa doktorska, Gliwice 2005.
[11] HORNE R.S., Spectrogram for Windows, Ver. 3.2.1.
[12] JASSEM W, Podstawy fontetyki akustycznej, PWN, Warszawa 1973.