ISOLATED WORD RECOGNITION BASED ON INTELLIGENT SEGMENTATION BY USING HYBRID HTD-HMM

A. Reza Kazemi, B. Bahram Ehsandoust, C. Alborz Rezazadeh, D. Shahrokh Ghaemmaghami

Sharif University of Technology, Tehran, Iran
Electrical Engineering Department
rezakazemi@ieee.org - ehsandoust@ee.sharif.edu - rezazadeh@ee.sharif.edu - ghaemmag@sharif.edu

ABSTRACT

In recent years, IWR (Isolated Word Recognition) was one of the main concerns of speech processing. The challenging problems in this field appear when the database become so large or when we have a lot of word with similarly pronounce in the database. This paper introduces a general solution for a traditional problem in isolated similarly pronounced word recognition, especially in large databases. One the important problem of traditional IWR is referred to their segmentation algorithm, their methods were lacking in efficiency due to the following reasons: First, using equal segmentation is not at all intelligent at all and as a result, cannot produce accurate results; besides, utilizing manual segmentation based on events is not possible in large databases. In this paper, by introducing the intelligent segmentation based on HTD (Hierarchical Temporal Decomposition), we present a more satisfactory answer to these problems. Based on this method, at first by using TD algorithm we categorize words into some groups with the same number of segments and then by using HTD method, all words of each group will be segmented to the number of events of the biggest group in TD step and based on the phi functions and related events. Then these segments will be sent to HMM step. Experimental results show that the proposed method significantly improves the recognition accuracy in comparison with recognition based on traditional segmentation.

Index Terms. HMM, HTD, Intelligent Segmentation.

1. INTRODUCTION

The Hidden Markov Model (HMM) [1] was a breakthrough in recognition systems. Thanks to this method, many practical approaches, utilizing different statistical models, were then developed, which causes noticeable improvements in speed and accuracy [2]. The strength of this probabilistic model is in the classification of a wide range of data based on their statistical characteristics. However, the HMM may fail to classify sounds of highly similar phonetic structure [3], as is the case in Persian digit recognition.

The HMM based on traditional segmentation can hardly distinguish words with similar pronunciation. Therefore, we decided to use an event-based segmentation, HTD, to overcome this problem. Hierarchical TD (HTD) relies on the idea of MTD. It uses a specific EAF as a priori information about event functions. The decomposition problem is then reduced to the problem of searching for the best locations to locate EAF over the given speech block, in the sense of minimum Mean Square Error (MSE) between the approximated and the original parameter sets. The EAF is thus located at the centroids of the event functions, in the order of effectiveness in the approximation error reduction. Accordingly, HTD optimizes the event locating task.

Isolated Persian digits are clear-cut examples of similarly pronounced words; so, we have applied our proposed method to this category. The process of recognition introduced here consists of two stages. First, the word is decomposed into small parts using intelligent segmentation (HTD method) based on events in every single word, which plays a significant role in improving our recognition algorithm due to its efficiency and high speed. Then the results of the first stage are passed onto the second stage in order to generate their Hidden Markov Model.

2. THE PROPOSED METHOD

As mentioned earlier, the main problem with recognition of words which have similar pronunciation is structural similarities between them that are of similar time-frequency patterns as well. An example is the similarity between digits two and nine in Persian, which are pronounced as /dv/ and /νv/ respectively. But for example, the pronunciation of six is completely different.

Figure 1 illustrates the spectrogram of samples of these digits. It can be deduced from these spectrograms just how close the spectral-temporal characteristics of these two numbers are. The energy distribution of these digits over the time-frequency plane is approximately the same. Moreover, the maximum and minimums of energies in these two spectrograms are very close and appear to follow the same patterns. In addition to these clues, our experiments also confirm these similarities. This figure completely shows the differences between the spectral-temporal characteristic of six with that of two and nine. The conventional HMM (based on traditional segmentation) recognizes these two numbers almost 15% of the time. The same assertions are true for other Persian digit groups, such as digits seven and eight, as well as one and three, due to their very similar phonetic structure.

In order to overcome this recognition problem, we applied the HTD method for intelligent segmentation (event-based segmentation) rather than traditional equal segmentation (with specific overlaps) of each word. Our aim was the recognition the nuances in phonetic structures. Utilizing the TD method does not suit this project for a reason: Since the outputs of the segmentation stage would have to be recognized by the HMM system, they should have equal events; but the TD method does not guarantee equal segmentation for different inputs; therefore we employed the HTD method to be able to control the number of events of a sample.
speech, even for different inputs. For example, imagine that we have 100 samples of speech of saying a digit. If we segment these samples by TD, it is not guaranteed that every sample has the same number of segments. So, first we segment these samples by TD, and check that how many of them have the same number of events. Then we choose the number of event of biggest group of samples with same number of events as the input of HTD. It means that we decide each specific digit has how many events. We decide this method because in HMM system, for recognition of digits, each specific digit has to have equal events. And how many events it should has? We determine by TD. Figure 2 shows that if we do not use TD for determining the number of segments, then the quality of results will get worse. In this chart the ratio of correct recognitions for the Persian Digits “Two”, “Six”, and “Nine”.

Fig. 2. The correction percentage in term of the number of segments for the Persian Digits “Two”, “Six”, and “Nine”.

2.1. Clustering And Intelligent Segmentation

Inasmuch as the conventional HMM (based on traditional segmentation) does not have enough efficiency for the recognition of words with similar pronunciation (such as our case study in Persian digits), and for large databases it is not practical to segment samples manually, we decided to use automatic event-based segmentation.

First, we need to train the recognition system. Imagine that we have 100 speech samples of a same word (in this case study, one of the Persian digits). Firstly, we use TD to determine number of events in each of these samples; then by using a clustering method (e.g. K-means) we can find the most probable number of events for this specific word (Let us call this most probable number of events NOE.) So after this step we use HTD to find the events of each sample. Since we need to give number of segments as an input to HTD, we should use NOE as this input. In this way, after finishing HTD step, all samples of each word have same number of segments. Then we send these segments to the next part, HMM. In fact, the main purpose of this step is to put the words with the same number of events into one class and then carry out an intelligent segmentation with HTD. In this way, the inputs of HMM (which are words with same number of events, as discussed in the related section) are provided. We did this for all of Persian digits and trained our system for all of them. After completing the training step, this system can recognize a new sample (which is one of the Persian digits in our case study) in this way: the input sample will go to HTD segment and we use different number of segments (NOE of each Persian digit which we have already calculated in training step) to cluster this sample. Then in HMM step we will find the similarity between each segment of the sample to segments of each of Persian digits (again, that we have already found out in training step) and calculate a possibility for measuring this similarity (It means that for each new unknown sample we will calculate 9 number, numbers of Persian digits). So the Persian digit with maximum possibility will be the answer.

We train our system with some samples by Temporal Decomposition algorithm. In this way, we classify the data with respect to their number of events (the words with close number of events are in one category); though experiment shows that almost all Persian digits have three events. Then, a new word would be compared to the words of the category with the equivalent number of events. One of the most important capabilities that HTD provides is intelligent windowing. As we know, each rectangular or triangular window is suitable for a particular group of inputs with specific properties and utilizing the appropriate window improves the performance of a system; but since the variety of the inputs is very high in speech, having an intelligent window could be very useful and significant. As we know HTD, in addition to determining the number of events and their centers, can provide us with the pattern of the $\phi$ functions. These patterns could be suitable candidates for the shape of our intelligent windows; but since the variety of
these shapes is high and their exact implementation requires many calculations, we consider the fundamental Gaussian shape as the primary shape (because by changing the parameter sigma, it can be converted to triangular, square-shaped, and other kind of shapes) and we regulate its \( \sigma \) in order to have maximum similarity with the considered pattern \( \phi \) functions. For maximizing the similarity, we try to equalize their surfaces; this can be achieved by regulating the \( \sigma \) in such a way that the surface of the windowing part of the Gaussian diagram amounts to the surface of the pattern \( \phi \) function (Since the Gaussian diagram is expanded from \(-\infty\) to \(+\infty\) but the length of our window is finite, only the central part of the Gaussian diagram is used for windowing). The Gaussian Diagram is also multiplied in a constant number in such a way that its maximum matches the maximum of the pattern \( \phi \) function. Using such an intelligent windowing increases the quality of the results without increasing the order of calculation of this completely intelligent segmentation, because in fact, it uses the unused outputs of the functions. HTD also furnishes us with another tool in addition to intelligent segmentation: After utilizing the HTD method, we have access to the \( \phi \) functions (events). As a result, we can use the Gaussian equivalent of each \( \phi \) function in the windowing of each segment. In fact a \( \phi \) function determines a unique center and \( \sigma \) for its equivalent Gaussian function. In this paper, by windowing these Gaussian functions for each segment, we improved the accuracy and efficiency of our work.

We provide an example for this segmentation in Figure 3. The input of HTD is sample speech of the digit: two. And we asked HTD to split it to three segments.

![Figure 3](image.png)

**Fig. 3.** Applying HTD with three segments to a sample Persian "Two".

Actually, the upper chart in this figure is illustrated \( \phi \) functions of each segment and the bottom one shows the acoustic signal of two and tells us how segmentation occurs.

### 2.2. The Proposed HMM Structure

Firstly, an HMM is trained for each vocabulary word using a number of representative examples of that word. It should be mentioned that all of the words of this section are equally segmented by HTD. Secondly, for the recognition unknown words, the likelihood of each model generating that word is calculated and the word is then identified by the most likely model.

In HMM, feature vectors can be Mel Frequency Cepstral Coefficients (MFCCs), Linear Predictive Coding (LPC) coefficients, or any appropriate parameter set. However, the problem lies in the fact that these feature vectors are so distributed as to make grouping and calculating needed matrixes for HMM impossible and, therefore, could not be assigned appropriate probability densities. To overcome this problem, one can use the K-means or the LBG algorithm \([8,9]\) to split the vector space into smaller divisions, in which there exists a representative that could be assigned to the subspace probability density and could be used to measure the distance of any desired vector from the desired subspace.

The number of subspaces and the metric that is used to estimate the spectral distances are arbitrary parameters, which are chosen according to the volume of the desired database.

Figure 4 summarizes all the stages of our experiment and intelligent windowing. This figure describes the entire system and the proposed algorithm. As it is illustrated in this figure (in the training part of the figure), first, the trainer words (in fact, reference data) are classified based on the number of their events; then, by using a clustering algorithm, like K-means based on the volume of the database, the words are reclassified to larger groups by combining groups with close number of events into one group and choosing a number of events for this new group (for instance, we combine groups with 2, 3 and 4 events to one group and call it groups of 3-event words). In the next step, the words of each group are broken up to the number of events assigned to their group; finally, based on what is explained above, intelligent windowing is applied and windowed segments are sent to HMM for training. The recognition part has exactly the same job. The only difference is that windowed segments are applied for recognition instead of training.

![Figure 4](image.png)

**Fig. 4.** (a) Network training by a standard database (b) Block diagram of proposed system when a new input enters it

### 3. SIMULATION RESULTS

In our experiments, we used approximately 400 Persian male speech samples and 50 Persian female speech samples of different age groups for each of the digits one through nine. We used 330 speech...
samples for training and the remaining 120 samples for testing our recognizer. These speech samples were recorded in the speech laboratory of the Electronics Research Center (ERC) of Sharif University of Technology with the sampling frequency of 11025 Hz. The MFCC parameters were chosen to be vectors of length 13. These parameters were calculated using the Bark frequency model with 13 linear and 27 logarithmic triangular (conventional) filters. As illustrated in Table 1, our experimental results show a significant improvement in recognizing isolated Persian digits, as compared to the conventional HMM (based on traditional segmentation recognizers). This is mainly due to the problem we already mentioned in the introduction. In the Persian language, as an example, there are some groups of digits that are so similar in pronunciation as to cause confusion in a human listener. As shown in Table 1, the results of the proposed method for some particular Persian digits with similar phonetic structures, like one and three, or two and nine are outstanding. The importance of differentiating between these digits in particular becomes more obvious by considering the fact that objective recognition of these digits is hard and Persian people usually make mistakes in their daily life. This issue is borne out by the high spectrum similarity shown in Figure 1.

<table>
<thead>
<tr>
<th>Clean</th>
<th>Two</th>
<th>Nine</th>
<th>Six</th>
<th>Two</th>
<th>Nine</th>
<th>Six</th>
</tr>
</thead>
<tbody>
<tr>
<td>70%</td>
<td>70%</td>
<td>80%</td>
<td>75%</td>
<td>73%</td>
<td>88%</td>
<td></td>
</tr>
<tr>
<td>30dB</td>
<td>65%</td>
<td>63%</td>
<td>66%</td>
<td>68%</td>
<td>67%</td>
<td>69%</td>
</tr>
<tr>
<td>20dB</td>
<td>55%</td>
<td>54%</td>
<td>59%</td>
<td>60%</td>
<td>59%</td>
<td>62%</td>
</tr>
<tr>
<td>10dB</td>
<td>50%</td>
<td>51%</td>
<td>52%</td>
<td>60%</td>
<td>58%</td>
<td>61%</td>
</tr>
</tbody>
</table>

Table 1. Proposed method results, as compared to results using a conventional HMM based speech recognizer.

As shown in Table 1, the proposed method yields a better performance in both clean and noisy conditions. The higher accuracy of this method is well demonstrated when we face the outstanding errorless detection of some digits; for example the digit six, which has less similarity in pronunciation with other Persian digits. In this method, the ability of the HTD in separating a huge amount of data is combined with the concise classification of the HMM in confined areas. As confirmed by the results shown in Table 1, the novel idea of the intelligent segmentation in the HTD part has improved the results.

A major advantage of this intelligent segmentation method is to rid the signal of the noisy segments in the initial and final part of the speech samples that can affect our decision in the conventional methods. In addition, the resulting segments became more stationary, making them more convenient for classifying the data. Because the similar samples now have less diversity compared to those in the non-stationary case, they concentrate better in the feature space. This concentration of the data makes it more easily divisible into smaller groups.

4. CONCLUSION

In this paper, by using the hybrid of HTD/HMM, we have applied event-based segmentation and windowing (a Gaussian window for each segment) and observed that the results were more accurate than using traditional segmentation in the recognition of similarly pronounced words. Due to the many similarities in phonetic structure and time-frequency pattern in Persian digits, one through nine, that make the recognition process a rather error-prone task, we chose this group of words for analyzing. To overcome this problem, a novel isolated word recognition system has been proposed. The system is based on a two-step hybrid HTD/HMM classification, which is appropriately chosen so as to increase the statistical efficiency of both the HTD and the HMM subsystems. Inasmuch as HTD is done in a short period of time, it has widespread applications and could be used in large databases. The HMM serves as a global classifier and was the final step of our method. It is shown that by means of a novel intelligent segmentation process and intelligent windowing, the efficiency of the recognition is increased dramatically. The intelligent segmentation idea provides the more stationary HTD and therefore increases the classification precision of this machine. The simulation results also confirm that the proposed method achieves a significant improvement over conventional methods in both clean and noisy environments.

5. REFERENCES