DISCRETE SPEECH RECOGNITION USING A HAUSDORFF BASED METRIC

An automatic word-based speech recognition approach

Tudor Barbu
Institute of Computer Science, Carol I 22A, Iaşi 6600, Romania

Keywords: Speech recognition, Discrete speech, Vocal sound, Mel cepstral analysis, Hausdorff metric, Feature vectors, Supervised classification, Training set

Abstract: In this work we provide an automatic speaker-independent word-based discrete speech recognition approach. Our proposed method consist of several processing levels. First, an word-based audio segmentation is performed, then a feature extraction is applied on the obtained segments. The speech feature vectors are computed using a delta delta mel cepstral vocal sound analysis. Then, a minimum distance supervised classifier is proposed. Because of the different dimensions of the speech feature vectors, we create a Hausdorff-based nonlinear metric to measure the distance between them.

1 INTRODUCTION

In this paper we are interested in developing a automatic speech recognition system. As we know, a speech recognition system receives a vocal sound as an input and provides as an output the text representing the transcript of the spoken utterance.

There are many known speech recognition approaches (Rabiner & Juang 1993). They can be separated in several different classes, depending on the linguistic units used in recognition. Thus, there exist phoneme-based, morpheme-based, word-based or phrase-based recognition techniques. We propose an word-based speech recognition approach.

Depending on the types of the vocal utterances that the system is able to recognize, it can be either a discrete speech recognition system or a continuous speech recognition system. Discrete speech contains slight pauses between spoken words (Logan 2000).

Continuous speech represents the natural speech, its words being not separated by pauses. The word-based segmentation of a vocal sound is easier for the discrete speech case. For realizing the continuous speech segmentation, special methods must be utilized to determine the words boundaries. We choose to perform a discrete speech recognition approach.

Also, depending on the population of users the recognition systems can handle, they can be either speaker-dependent or speaker-independent speech recognition systems. The systems in the first class must be trained on a specific user, while those in the second are trained on a set of speakers (Furui 1986).

We focus on a speaker-independent system. Its modelling consists of the following processing steps:

1. Creating the system training feature set
2. Audio preprocessing of the input vocal sound
3. Word-based segmentation of the input utterance
4. Feature vector extraction for signal segments
5. Supervised classification and labelling of the speech segments
6. Obtaining the final transcript of the speech from the previously determined labels

We will describe each of these developing stages in the next sections. Also, we will present some results of our experiments.

The main contributions of this paper are the developing of a nonlinear metric based on the Hausdorff distance for sets (Gregoire & Bouillot 1998) and creating a supervised classifier which uses the proposed distance.

2 TRAINING FEATURE VECTOR EXTRACTION

We propose a supervised method for vocal pattern recognition, therefore we must develop a speech

Barbu T. (2004).
DISCRETE SPEECH RECOGNITION USING A HAUSDORFF BASED METRIC - An automatic word-based speech recognition approach.
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 363-368
DOI: 10.5220/0001381803630368
Copyright © SciTePress

training set. In this section we focus on this training set.

An word-based approach requires a very large training set to perform an optimal speech recognition. The set dimension depends on the chosen vocabulary. An ideal recognition system uses the whole vocabulary of the speech language, which may contain thousands words. Most speech recognition systems use low-size vocabularies, having tens words, or medium-size vocabularies with hundreds words.

A large vocabulary size may represent a disadvantage for the word-based speech recognition techniques. Vocabulary size is equal with the classes number because each word corresponds to a class in the recognition process. It is obvious that a phoneme-based recognition system uses a much smaller number of classes, because the number of words of a language (tens thousands) is much greater than the number of the phonemes of that language (usually 30-40 depending on language).

We consider creating a vocabulary containing several words only initially and extending it over time. Let *N* be our vocabulary size. For each word from the vocabulary we consider a set of speakers, each of them recording a spoken utterance of that word.

Thus, we obtain a set of digital audio signals for each word. All these recorded sounds represent the *prototypes* of the system. For each *i* we get a set of signal prototypes $\{S_1^i,...,S_{n_i}^i\}$, where $i=\overline{1,N}$, n_i is the number of users which produce the i^{th} word and S_j^i represents the audio signal of the spoken word recorded by the j^{th} speaker, $j=\overline{1,n_i}$. The sequence $\{S_1^1,...,S_{n_1}^1,...,S_1^N,...,S_{n_N}^N\}$ represents the training set of our recognition system.

Also, we set class labels for all these signals. The label of a signal of a spoken word will be its transcript (the written word). Therefore, for each $i=\overline{1,N}$ and $j=\overline{1,n_i}$, we set a signal label $l(S_j^i)$. Obviously, it results:

$$l(i)=l(S_1^i)=...=l(S_{n_i}^i),\forall i=\overline{1,N}\ , \qquad (1)$$

where $l(i)$ represents the label of the i^{th} word related class.

The *prototype vectors*, representing the feature vectors of the training set, are then computed. We perform the training feature extraction by applying a mel cepstral analysis to the signals S_j^i, the Mel Frequency Cepstral Coefficients (MFCC) being the dominant features used for speech recognition (Minh 2000, Furui 1986, Logan 2000).

A short-time signal analysis is performed on each of these vocal sounds. Each signal is divided in overlapping segments of 256 samples with overlaps of 128 samples. Then, each resulted signal segment is windowed, by multiplying it with a Hamming window of length 256. We compute the spectrum of each windowed sequence, by applying DFT (Discrete Fourier Transform) to it, and obtain the acoustic vectors of the current S_j^i signal. Mel spectrum of these vectors is computed by converting them on the melodic scale that is described as:

$$mel(f)=2595\cdot\log_{10}(1+f/700)\,, \qquad (2)$$

where *f* represents the physical frequency and *mel*(*f*) is the mel frequency. The mel cepstral acoustic vectors are obtained by applying first the logarithm, then the DCT (Discrete Cosinus Transform) to the mel spectral acoustic vectors.

Then we compute the delta mel frequency cepstral coefficients (DMFCC), as the first order derivatives of MFCC, and the delta delta mel frequency cepstral coefficients (DDMFCC), as the second order derivatives of MFCC. We prefer to use the delta delta mel cepstral acoustic vectors for describing speech content. These acoustic vectors have a dimension of 256 samples. To reduce this size, we truncate each acoustic vector to the first 12 coefficients, which we consider to be sufficient for speech featuring. Then we create a 12 row matrix by positioning these truncate delta delta mel cepstral vectors as columns. The obtained DDMFCC-based matrix represents the final speech feature vector.

Thus, the training feature set becomes $\{V(S_1^1),...,V(S_{n_1}^1),...,V(S_1^N),...,V(S_{n_N}^N)\}$, where each feature vector $V(S_j^i)$ represents a 12 row matrix whose column number depends on S_j^i length.

3 INPUT SPEECH ANALYSIS

In this section we focus on the input vocal sound analysis. As we mentioned in introduction, we consider only discrete speech sounds to be recognized by our system. Also, we set the condition that the words of input spoken utterance belong to the given vocabulary.

Let *S* be the signal of the vocal sound to be recognized. First, several pre-processing actions may be performed on it (Rabiner & Schafer 1978). For example, if an amount of noise is still present in the sound, some filtering techniques should be applied for smoothing.

Also, in this pre-processing stage an important audio special effect may be added to the signal. The *preemphasing* effect is applied as follows:

$$S[a] = S[a] - \alpha \cdot S[a-1], \quad (3)$$

where we set the control parameter $\alpha = 0.5$.

The next stage of speech analysis consist of an word-based vocal segmentation. We must extract from *S* the signal segments corresponding to the spoken words. This task is performed by detecting the pauses that separate the words.

A pause segment is characterized by its length and by its low amplitudes. Thus, we use two threshold values for the pause identification purpose, let them be *T* and *t*. A pause represents a signal sequence $\{S[a],...,S[a+t]\}$, having the property $S[a],...,S[a+t] \le T$. We choose the length related parameter *t* = 500 and a small enough amplitude related threshold, *T*.

As a result of the pauses identification, the word related segment extraction process becomes quite simple to fulfil. Let $s_1,...,s_n$ be the segment signals extracted from *S*. Obviously, the recognition of *S* consist of determining the written words corresponding to these signals.

For each $i = \overline{1,n}$, the recognition system has to find the word represented by s_i. An automatic recognition is performed by comparing these speech signals with the prototypes of the training set.

A delta delta mel cepstral based feature extraction process is applied to each s_i sequence. For each $i = \overline{1,n}$, a feature vector $V(s_i)$ is computed as a truncated delta delta mel cepstral matrix, having 12 coefficients per column and a number of columns depending of s_i length.

Having different dimensions, the feature vectors and the training feature vectors cannot be compared to each other using linear metrics, like Euclidean distance. A solution for measure the distance between a feature vector $V(s_k)$ and a training vector $V(S_j^i)$, could be a resampling of one of these matrices.

They have always the same number of rows, only the number of columns could differ. The matrix having a greater number of columns may be resampled to get the same number of columns as the other. The transformed feature vectors, being same sized matrices, can be compared using an Euclidean distance for matrices.

This vector resampling solution is not an optimal one because resampling process may often produce a speech information loss on the transformed feature vector. If there is a great size difference between the vectors to be compared, this operation may result in further classification errors. Therefore, we propose a new type of distance measure in the next section.

4 A HAUSDORFF-BASED NONLINEAR METRIC

The classification stage of our speech recognition process requires a distance measure between different sized vectors. Therefore, we introduce a nonlinear metric which works for matrices and it is based upon the Hausdorff distance for sets.

First, let us present some general theory regarding Hausdorff metric. If *A* and *B* are two different-sized sets $(|A| \neq |B|)$, the Hausdorff metric related to them is defined as the *maximum distance of a set to the nearest point in the other set*. It can be formally described as:

$$h(A,B) = \max_{a \in A}\{\min_{b \in B}\{dist(a,b)\}\}, \quad (4)$$

where *h* represents the Hausdorff distance between sets and *dist* is any metric between points (Gregoire & Bouillot 1998).

In our case we must compare two matrices having one common dimension, instead of two sets of points. So, let us consider $A = (a_{ij})_{n \times m}$ and $B = (b_{ij})_{n \times p}$ the two matrices. We use notation *n* for the rows number, although we already used it in the last section. Let us assume that $m \neq p$.

We introduce two more vectors, $y = (y_i)_{p \times 1}$ and $z = (z_i)_{m \times 1}$, then compute $\|y\|_p = \max_{0 \le i \le p} |y_i|$ and $\|z\|_m = \max_{0 \le i \le m} |z_i|$. With these notations we create a new nonlinear metric *d* having the following form:

$$d(A,B)=\max\left\{\begin{array}{l}\sup\limits_{\|y\|_p\le 1}\inf\limits_{\|z\|_m\le 1}\|By-Az\|_n,\\ \sup\limits_{\|z\|_m\le 1}\inf\limits_{\|y\|_p\le 1}\|By-Az\|_n\end{array}\right\} \quad (5)$$

This restriction based metric represents the Hausdorff distance between the sets $B\left(y:\|y\|_p\le 1\right)$ and $A\left(z:\|z\|_m\le 1\right)$ in the metric space R^n. It can be expressed as:

$$d(A,B)=h\left(B\left(y:\|y\|_p\le 1\right),A\left(z:\|z\|_m\le 1\right)\right) \quad (6)$$

As resulting from (6), the metric *d* depends on *y* and *z*. Trying to eliminate these terms, we obtain a new form for *d* which do not depend on these vectors and it is not a Hausdorff distance anymore. This Hausdorff-based metric can be described as:

$$d(A,B)=\max\left\{\begin{array}{l}\sup\limits_{1\le k\le p}\inf\limits_{1\le j\le m}\sup\limits_{1\le i\le n}\left|b_{ik}-a_{ij}\right|,\\ \sup\limits_{1\le j\le m}\inf\limits_{1\le k\le p}\sup\limits_{1\le i\le n}\left|b_{ik}-a_{ij}\right|\end{array}\right\} \quad (7)$$

This nonlinear function *d* given by (7) verifies the main distance properties:

1. Positivity: $d(A,B)\ge 0$
2. Simetry: $d(A,B)=d(B,A)$
3. Triangle inequality: $d(A,B)+d(B,C)\ge d(A,C)$

The distance between any two matrices having a single common dimension, can be measured using this metric. In our speech recognition context matrices *A* and *B* are speech feature vectors.

The created distance constitutes a very good discriminator between feature vectors in the classification process. From our tests it results that if two speeches are similar enough, then the distance between their feature vectors, $d(v_1,v_2)$, become quite small. In the next section we use this metric for creating a proper classifier.

5 A SUPERVISED SPEECH CLASSIFICATION APPROACH

The next stage of the automatic speech recognition process is the speech classification. Our pattern recognition system use a supervised classifier (Duda & Hart & Stork 2000).

As we know, the patterns to be classified are the sound signals $s_1,...,s_n$. We also know that each of them represents a vocabulary word, but we do not know *what* word it is. That word can be identified by inserting that signal in a class and labelling it with the class label, which represents a written word. A signal s_k can be inserted in a word related class if its feature vector $V(s_k)$ is closed enough, in terms of a chosen metric, to the training feature vector set, $\{V(S_1^i),...,V(S_{n_i}^i)\}$, corresponding to that class.

We propose an extended variant of the minimum distance classifier. The classical form of this classifier consist of a set of prototypes, one for each class, and an appropriate metric. A pattern to be recognized is inserted in the class corresponding to the closest prototype. The extended form of the classifier has not only one but more prototypes for each word related class. The nonlinear metric presented in last section is used as a distance measure between feature vectors.

For each speech signal, each class is considered and the mean value of the distances between its feature vector and the training vectors of that class is computed. The speech signal is then inserted in the class corresponding to the smallest mean distance value and receives the label of that class.

Therefore, if s_k is the current speech signal, it must be placed in the x^{th} class, where $x=\arg\min\limits_i \dfrac{\sum_{j=1}^{n_i} d(V(s_k),V(S_i^j))}{n_i}$, the metric *d* representing the distance given by relation (7). Thus, the speech recognition process is formally described by:

$$l(s_k)=l\left(\arg\min_i \frac{\sum_{j=1}^{n_i} d(V(s_k),V(S_i^j))}{n_i}\right)\Bigg|_{\forall k=\overline{1,n},i=\overline{1,N}} \quad (8)$$

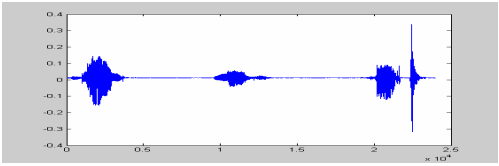

Figure 1: Discrete speech sound signal

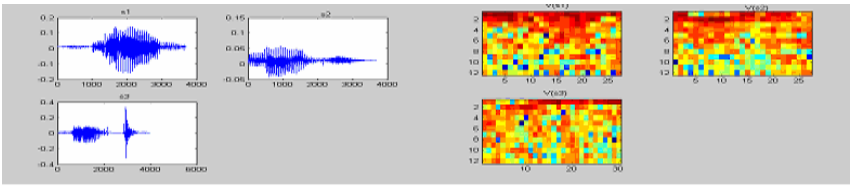

Figure 2: Speech signals and their feature vectors

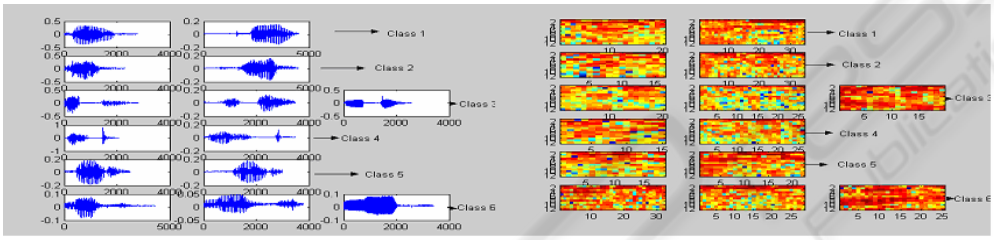

Figure 3: Training set and the training feature vectors

Thus, for each $k = \overline{1,n}$, the written word corresponding to speech signal s_k is identified as $l(s_k)$. The transcript of the entire speech S results as a concatenation of these labels.

Thus, the final result of our automatic speech recognition process is the label $l(S)$, computed as follows:

$$l(S) = l(s_1) + \ldots + l(s_n), \qquad (9)$$

where the meaning of the operator '+' is the string concatenation.

6 EXPERIMENTS

In this section we present some practical results of our experiments. We have tested the described recognition system for English language and obtained many satisfactory results. We get a 80% speech recognition rate using the proposed approach. For space reasons we present a simple test in this paper.

Thus, we consider a discrete vocal sound whose speech has to be recognized. Its signal S is the one represented in Figure 1. We perform a vocal segmentation on S, using $T = 0.01$, and the three speech signals, s_1, s_2, s_3, represented in Figure 2, are thus obtained.

For each s_i, a delta delta mel cepstral feature vector $V(s_i)$ is then computed. The feature vectors are displayed as color images in Figure 2.

We use in this experiment a very small English vocabulary, which is: {*car*, *John*, *apple*, *help*, *hurry*, *needs*}. All the spoken words of S belong to this vocabulary.

We consider three speakers for the third and the last word, and only two speakers for each of the others. Thus, the following training set is obtained: $\{S_1^1, S_2^1, S_1^2, S_2^2, S_1^3, S_2^3, S_3^3, S_1^4, S_2^4, S_1^5, S_2^6, S_3^6\}$. All these prototype signals are displayed in Figure 3. The signals related to i^{th} class are represented on the row i.

Therefore, for these classes we obtain the following labels: $l(1)$ = '*car*', $l(2)$ = '*John*', $l(3)$ = '*apple*', $l(4)$ = '*help*', $l(5)$ = '*hurry*' and $l(6)$ = '*needs*'. The training feature vectors $V(S_j^i)$, as they result from the delta delta mel cepstral analysis, are represented as color images in the same figure.

We compute first the distances given by (7) between the feature vectors displayed in Figure 2 and the training vectors displayed in Figure 3. For

each s_i the mean distance to each class is then computed. The obtained values are registered in the next table.

Table 1

	1	2	3	4	5	6
$V(s_1)$	4.10	3.02	4.36	4.27	3.99	5.19
$V(s_2)$	6.41	4.24	5.46	5.79	6.59	3.01
$V(s_3)$	3.36	3.03	2.91	2.76	3.53	4.47

As it results from Table 1, the minimum mean distance from $V(s_1)$ to a training feature vector set is 3.02. This value corresponds to the second class, therefore s_1 must be inserted in that class and $l(s_1) = l(2) = 'John'$.

From the table row related to $V(s_2)$ it results that the minimum mean distance value is 3.01 and it is related to the sixth class. Therefore, we get $l(s_2) = l(6) = 'needs'$. For the third feature vector, $V(s_3)$, the minimum mean distance is 2.76 and it is related to the fourth class. Thus, $l(s_3) = l(4) = 'help'$.

The final speech recognition result, being the label of vocal signal *S*, is obtained as $l(S) = l(s_1) + l(s_2) + l(s_3)$. This means that the initial vocal sound speech transcript is $l(S) = 'John\ needs\ help'$.

7 CONCLUSIONS

We have described a model for an automatic speaker-independent word-based discrete speech recognition system. The main novelty brought by this work is a nonlinear metric which works properly in discriminating between different sized speech feature vectors.

There exist Hausdorff-based metrics utilized in image processing domain. We have created such a distance which works in the speech recognition field.

Also, we have tested our method and obtained good results. In our experiments low-size vocabularies were used. Our idea is to extend such a vocabulary over time, adding more and more words, until it reaches a considerable size.

Obviously, the Hausdorff-based distance we have provided will work properly with other types of speech recognition approaches, which were mentioned by us in the introduction. Thus, our future research will focus on the continuous speech recognition and the phoneme-based speech recognition. In both cases we want to keep using this nonlinear metric in the feature classification stage.

REFERENCES

Rabiner, L., Juang, B. H., 1993. Fundamentals of Speech Recognition. Prentice Hall Signal Processing Series. Prentice Hall, Englewood Cliffs, New Jersey 07632, A. V. Oppenheim, Series Editor.

Rabiner, L., Schafer, R., 1978. Digital Processing of Speech Signals. Prentice Hall Signal Processing Series. Prentice Hall, Englewood Cliffs, NJ.

Minh N. Do, 2000. An Automatic Speaker Recognition System. Digital Signal Processing Mini-Project. Audio Visual Communications Laboratory, Swiss Federal Institute of Technology, Lausanne.

Furui, S, 1986. Speaker-independent isolated word recognition using dynamic features of the speech spectrum. IEEE Transactions on Acoustic Speech and Signal Processing. Vol ASSP-34, No.1, 52-59.

Logan B., 2000. Mel Frequency Cepstral Coefficients for Music Modelling. In Proc Int. Symposium on Music Information Retrieval (ISIMIR). Plymouth, MA.

Gregoire, N., Bouillot, M., 1998. Hausdorff distance between convex polygons. Web project for the course CS 507 Computational Geometry, McGill University.

Duda, R., Hart, P., Stork, D., G., 2000. Pattern Classification. John Wiley & Sons.