A CONTINUOUS VQ CLUSTERING ALGORITHM FOR REALTIME SPEECH RECOGNITION

Xixian Chen & Changnian Cai

Dept. of Telecommunications,
Beijing University of Posts & Telecommunications,
P. O. Box 103, Beijing, 100088, P. R. China

Abstract

This paper presents a continuous VQ clustering (CVQC) algorithm for realtime speech recognition, which incorporates the temporal information of speech into both training and recognition processes. In comparison with the conventional DTW and VQ methods, this new algorithm delivers faster training and recognition speed and smaller codebook size while still retains merits of both. Realtime implementation is emphasized in the design of sophisticated algorithms. And a custom available voice controlled computer command input system based on CVQC is also introduced.

1. Introduction

In the area of automatic speech recognition, time alignment of test and a reference pattern is one of the fundamental problems, because the speaking rate variation causes nonlinear fluctuation in time scale of speech pattern, especially, of a continuous polysyllabic utterance where there may a short silence between the syllables. One of the most successful pattern matching method for speech recognition has been the so called dynamic time warping (DTW) algorithms \(^\text{(1)}\), where in time scale variation between the reference and the test utterance patterns is eliminated by warping the time axis of one that the time alignment of maximum similarity is obtained. Because of errors in endpoint detection, the obtained word patterns may be so short that some parts of speech are lost or so long that some silence is included in. Thus, the optimal alignment of the test and reference patterns is not always obtained at the beginning and the ending frames of speech. Another obstacle to its widespread applications lies in its huge amount of computation.

In recent ten years, several different approaches based on vector-quantization technique \(^\text{(2, 3)}\) have been proposed where the amount of computation is highly reduced over that required for the DTW method. Paper \(^\text{(2)}\) presented a word-based vector quantizer to remove all time-sequence information from the training utterance by representing each vocabulary word as a set of independent feature vectors referred to as codebook. An input utterance is encoded with every codebook and is recognized as the one corresponding to the smallest distortion. Since the nearest neighbor distortion calculation is irrelevant to the sequence of the input speech frame, no time alignment is needed.

In this paper, a more efficient algorithm, referred to as a continuous VQ clustering algorithm (CVQC), is proposed which incorporates the temporal information of speech into both training and recognition processes, while reducing the computation complexity and memory requirements even further. The training speech of each word or phrase in the vocabulary is initially separated into a number of daughter sets with each containing \(m\) consecutive frames of speech, then for each two adjacent daughter sets, a continuous clustering algorithm is performed to create a set of time-mapped cluster centers as a reference pattern. Classification consists of dividing the input speech into appropriate sections, calculating a continuous nearest neighbor distortion, and choosing the recognized word to be the one that gives the smallest distortion.

As an application, a realtime voice controlled computer system based on this new principle has been developed and was evaluated with a number of typical vocabularies, which shows more than 95% recognition accuracy for 40 commonly used DOS command set and 39 Chinese provinces and regions' names and yields 95% accuracy for the carefully selected Chinese characters which have different initials but contain the same vowel "a".

2. Training Procedure

The scheme for creating a reference pattern is as follows: The input speech was first low-pass filtered to 3.4 kHz and sampled at a rate of 6.8 kHz. After the realtime endpoint detection based on the energy temporal contour of the signal, it is then block into frames of 23.52 ms (160 samples) with each consecutive frame spaced 11.76 ms (80 samples). Thus, each speech sample falls within two consecutive analysis frames. The basic feature vector used for modeling the speech frame is a set of weighted parcor-coefficients:

\[
wk(n,i)=k(n,1)\alpha(n,1)/a(n,P)
\]

(1)

\(i=1,\ldots,P; n=1,\ldots,N\)

where \(k(n,1)\) is the parcor-coefficient of the nth frame, \(a(n,1)\) is the ith prediction residual, and \(P=81\) is LPC analysis order, which are extracted from the speech frame in the amount of actual half frame time. Thus, the system outputs a sequence of \(N\) feature vectors representing the input speech:

\[WK(1), WK(2), \ldots, WK(N)\]

with each vector having components \(wk(n,1)\) through \(wk(n,P)\):
where the superscript $T$ stands for the transpose of a vector.

After obtaining the feature vectors of speech, we will consider the strategies that could make full use of them. As it is known, the speech signal can be modeled as a non-stationary stochastic process, with its vocal tract characteristics varying with the time, especially near the beginning and the ending frames of speech. As it is shown below, the CVQC algorithm can successfully get rid of some of the immaterial and non-steady information of the speech, while still keep the changing contour of the vocal tract in compressing the raw feature vectors into the templates.

The first step in condensing the raw data is to apart the $N$ vectors available into $L = \lceil N/(L-1) \rceil$ daughter sets $S(j), j=1,2, \ldots , L$, with each consisting of $m$ ($m=4$ in our experiments) consecutive frames of feature vectors:

$$S(j) = \{ \text{WK}(mj+n), n=1,2, \ldots , m \}$$

where $\text{INT}[x]$ stands for the biggest integer less than $x$. In case that $N$ may not be divided exactly by $m$, the last one $S(L)$ contains the last $m$ consecutive vectors:

$$S(L) = \{ \text{WK}(N-m+n), n=1,2, \ldots , m \}$$

The centroid of each daughter set is then calculated and is taken to be the initial reference pattern:

$$\text{RF}(j) = (\text{WK}(L))_{/m}, j=1,2, \ldots , L$$

where

$$N(j) = \{ i : \text{WK}(i) \in S(j) \}$$

is the index set containing the index of frames whose feature vectors are assigned to the daughter set $S(j)$.

To improve the quality of the templates, the so called continuous Clustering algorithm will be performed among the initial divided daughter sets. It begins with the first two daughter sets $S(1)$ and $S(2)$ with the $\text{RF}(1)$ and $\text{RF}(2)$ being the initial centroids. Through the nearest neighbor searching process, the vector elements within them will be re-assigned to the two new daughter sets $S'(j), j=1,2, \ldots , L$, and then two new centroids are calculated. Repeat this process until the last two manipulation results the same which gives the first two centroids. But only the first one is confirmed to be the code word. The same process is continued for the other one $S'(2)$ ( having been clustered with $S(1)$) and a subsequent set $S(3)$, which gives the second code word. This process is continued until the $L$ code words are generated.

Obviously, Our codebook is much different from the conventional VQ codebook in that all code words in order are time-mapped with the input speech. It is of a great potential in improving the recognition accuracy. As a side benefit, the memory requirement is $1/m$ of that required by DTW method.

Further analysis shows that if the input speech consists of multi-syllable, the troublesome boundary mis-allocation effect is eliminated, because the local VQ process guarantees proper syllable separation.

3. Searching Strategy

After having derived the reference pattern for each vocabulary word, we will now consider the strategies for recognition. To compensate the nonlinear variations in speaking rates, the input speech is first divided into appropriate sections which are equal to the number of the code words of its reference counterpart, then a continuous vector quantization algorithm is performed for each frame of the speech by using the nearest three consecutive code words. We define $T(1), T(2), \ldots , T(N)$ as the feature vectors of the test speech and $d(n,j)=d(T(n),RF(j))$ the distortion between the $n$th frame of the test speech and the $j$th cluster center of the reference pattern. A continuous nearest neighbor distortion for each frame is then calculated as:

$$\text{DIS} = \sum_{n=1}^{N} \min \{ d(n,j), d(n,j+1) \}$$

where

$$d(n,j) = \min \{ d(n,j-1), d(n,j), d(n,j+1) \}$$

and the total distortion is

$$\text{DIS} = \sum_{n=1}^{N} d(n)$$

Fig. 1 shows the possible paths for the time alignment operation. There are two main differences between CVQC and DTW in obtaining the optimum path for minimum distortion calculation. In DTW, the warping path is monotonously directed, but in CVQC, it can move up and down locally within two zigzag boundaries. Thus the new method has more flexibility than DTW in the template matching process, which increases the robustness in combating the irregular variation in speech. The insensitivity to the endpoint process is another advantage gained by CVQC. For the basic DTW algorithm, the endpoints of the patterns are assumed to be precisely determined that is seldom in case, and are fixed in distortion calculation. While in
CVQC approach, a fuzzy endpoints' region with length of 2m frames is introduced. All the test frames within this region could be the candidate ending frames which is determined through the local NN rule. It successfully offsets the inexactness of the endpoint location—a key factor in improving the recognition accuracy.

It is interesting to compare the computational efforts and memory requirements of the approach to those of DTW. Let N1 and N2 be the numbers of reference and test frames, respectively. Then there will be L (=INT[N1/m]+1) feature vectors which require L·P memory locations, where P is the dimension of each feature vector. From (6), the comparison for each reference pattern requires 3·N2 distortion computations.

In DTW approaches, each reference template requires N1·P storage locations, and the distortion calculation for each reference template is proportional to N1·N2. Obviously, a significant reduction of both memory storage locations and computational efforts is achieved by the CVQC method.

4. Multiple training sequences

In order to achieve more reliable and consistent reference patterns for discrimination, the several repetition of the same word utterances can be used to create the averaged templates. Suppose there are M repetition of the utterances for a certain word in the recognition vocabulary and the kth utterance contains N(k) frames of speech, the averaged templates can then be created as follows:

\[
M \rightarrow \frac{1}{M} \cdot \sum_{k=1}^{M} N(k)
\]

(1) Calculate the mean frame number:

(2) Compute the averaged number of daughter sets:

\[
L = \text{INT}[\frac{N}{m}] + 1
\]

(3) Divide the feature vectors of each utterance into L daughter sets:

\[
S(k,1), S(k,2), \ldots, S(k,L) \quad k=1,2,\ldots,M
\]

where \( S(k,j) \) stands for the jth daughter set of the kth utterance. Then the feature vectors of the corresponding daughter sets are merged into a larger daughter set:

\[
S(j) = \cup_{k=1}^{M} S(k,j), \quad j=1,2,\ldots,L
\]

(4) After calculating the centroid of each subset, the averaged templates can then be derived from the new daughter sets \( S(j), j=1,2,\ldots,L \) by using the CVQC clustering algorithm.

In case the memory size is limited so that the feature vectors of a large number of utterances could not be pre-stored, a recursive clustering algorithm could be employed to refine the templates. On thought that the feature sequences derived from the same word utterances belong to the same statistical model, so their variability and other corruptive influences may be well behaved. Therefore, it is possible to updating the templates on the bases of one utterance after another.

To be more specific, we will consider a simple case where the cluster centers are calculated through an arithmetic mean. After the first time the reference pattern being created, the newly incoming speech for training is first matched with the previously created word templates by CVQC searching strategy with the frames nearest the jth cluster center being clustered to daughter set \( S(j), j=1,\ldots,L \). Let \( N(j) \) be the index set containing the index of frames whose feature vectors are clustered to the daughter set \( S(j) \), the sum

\[
\frac{N(j)}{L} = \frac{1}{L} \sum_{i=1}^{N(j)} 1
\]

provides the new feature information for the jth template and is accumulated with previous ones:

\[
S(k,j) = S(k,j) + \frac{1}{L} \sum_{i=1}^{N(j)} 1
\]

which keeps the statistical feature information contributed by all the training utterances up to now. The new cluster centers can then be generated by averaging \( S(k,j) \) for \( j=1,2,\ldots,L \). This process is repeated, with the newly generated templates acting as cluster centers, until a satisfactory result is achieved. It is clear that the recursive training procedure requires smaller memory size and delivers faster training speed.

Increasing the training times does improve the recognition performance, but to conduct a large quantities of training work on a large number of utterances is dull
and annoyed and could not be accepted by most users. A better method is to refine the templates during the recognition phase. As soon as the the decision is made on the input speech, the templates corresponding to the recognized word are automatically updated according to the following formula:

\[
RF(j) = (1-\beta)RF(j) + \beta \left( \frac{I WK(1)}{m(j)} \right) m(j) \quad (13)
\]

where \( m(j) \) is the number of feature vectors in \( S[j] \) and \( \beta \) is a small positive number less than one. In this way, we can increase the training times for most of the words having a high recognition rate without performing a large amount of training work and thus gradually improve the quality of the templates being used.

5. Voice Controlled Computer System

This section describes the implementation of a voice controlled computer system which accepts spoken command through the microphone instead of the input from the keyboard. The above mentioned algorithm is optimized for realtime recognition. Of particular significance to speed up the feature extraction and CVQC based clustering and pattern matching algorithm in realtime, is UDSP-10, a TMS32010 based high speed universal digital signal processing board. The host computer may be IBM PC/XT, AT and other compatibles. It allows the user to easily perform complex operations such as modifying the energy threshold used in endpoint detection, entering DOS command (or other vocabulary) strokes, training and refining the voiceprints, testing the recognition performance, and loading and storing the templates. Once the command key strokes and their corresponding voiceprints (templates) are created, they can be fitted into the host computer operating system by 'Speech Keyboard' software. The system can then recognize spoken words or phrases by matching the speech patterns to these pre-recorded templates. If a match is found, the corresponding command is executed just as input from the keyboard. The maximum response time is less than 20ms for 40-word templates.

In the following, we will conduct an interesting experiment on carefully selected ten Chinese characters which have different initials but contain the same vowel "a". These words are a, ba, da, fa, la, ma, na, pa, ta, ya, with each voiced 25 times by a female. The first 5 utterances are used to create the templates and the rest of them are tested in the recognition phase. The recognition results are shown in Table 1.

The speech recognition system was also evaluated with a number of commonly used vocabularies: 10 Chinese digits, 33 Chinese provinces and regions' names, 40 commonly used MS-DOS commands, and 39 alphadigits. The performance index is given in Table 2.

To conclude, the new algorithm introduced in this paper gives a qualified recognition performance with a relatively lower computation and memory requirement than DTW and VQ approaches, which is particularly suitable for realtime application where the time and memory requirements are critical.

Table 1. Recognition Results for ten Chinese words with the same vowel a.

<table>
<thead>
<tr>
<th>Words</th>
<th>Number of Utterances</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>ba</td>
<td>20</td>
<td>10%</td>
</tr>
<tr>
<td>da</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>fa</td>
<td>20</td>
<td>15%</td>
</tr>
<tr>
<td>la</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>ma</td>
<td>20</td>
<td>5%</td>
</tr>
<tr>
<td>na</td>
<td>20</td>
<td>10%</td>
</tr>
<tr>
<td>pa</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>ta</td>
<td>20</td>
<td>5%</td>
</tr>
<tr>
<td>ya</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>200</td>
<td>5%</td>
</tr>
</tbody>
</table>

Table 2. Word Recognition Performance Index for A Number of Vocabularies

<table>
<thead>
<tr>
<th>Vocabulary</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 digits</td>
<td>Less than 1%</td>
</tr>
<tr>
<td>33 Chinese Prov. &amp; Regions' Names</td>
<td>Less than 2%</td>
</tr>
<tr>
<td>40 DOS Comm. Set</td>
<td>Less than 2%</td>
</tr>
<tr>
<td>39 Alphadigits</td>
<td>About 9%</td>
</tr>
</tbody>
</table>

References

