A Preliminary Study on the
Static Representation of Short-timed Speech Dynamics

Chorkin Chan, Jun Bao & Jian-xiong Wu

Department of Computer Science, University of Hong Kong

ABSTRACT

The distribution of feature vectors derived from short speech segments is considered a mixture of Gaussian densities. Each density corresponds to a phonetic state in the speech production process and is trained by the observed feature vectors derived from waveform segments corresponding to a specific phoneme. We propose in this paper that a short utterance of speech like a syllable, can be statically represented by a matrix of state transition probabilities considering the utterance as a chain of discrete acoustic events. With these static models, we experienced very competitive performance in terms of recognition rates and speeds when compared with that using Dynamic Programming and Hidden Markov Models for recognition. We are convinced that the proposed static model is more resilient against the lack of huge amount of training data and characterizes the dynamics of short utterances sufficiently for recognition purposes.

1. Introduction

The motive behind our interest to pursue a static representation of speech comes from the fact that most pattern recognition techniques were designed for patterns as single vectors in the feature space. Being continuous in nature, speech is often approximated by a sequence of vectors each representing an acoustic event. Each vector in the sequence corresponds to a segment of speech short enough to be assumed stationary and speech dynamics is expressed through the discrete changes from one segment to the next. The highly successful technique of dynamic programming (DP) [1] simply measures the dissimilarity between two utterances by lining up the corresponding sequences in a nonlinear fashion and accumulating the distortions between the paired up vectors from the two respective sequences. On the other hand, the equally successful hidden Markov model (HMM) [2] tries to represent an utterance with a speech production model which is most probable to generate the utterance. A noteworthy similarity between these two techniques is that they both try to faithfully portray an utterance by installing all the information provided. However, any student of pattern recognition knows that features effective in discrimination should be given heavy weights while irrelevant ones should be discarded as far as recognition is concerned. With this concept in mind, we look for a representation of speech dynamics which is more compatible with this classical pattern recognition principle. Furthermore, if speech dynamics can be expressed in terms of a static model, many classical pattern recognition techniques can be applied readily.

2. Speech as a discrete process of state transitions

The distribution of feature vectors can be considered a mixture density composed of a number of, say m, Gaussian clusters. Our basic assumption is that each cluster corresponds to an identifiable phonetic state of the speech production mechanism and speech production is considered a discrete process of state transitions. In a short utterance of speech like a syllable, the pattern of state transitions is simple enough that a record of transition probabilities as observed in the utterance, irrespective to their order of happening, is sufficient to represent the utterance. This assumption is different from the well known hidden Markov model in that the densities can be trained by supervised learning of feature vectors derived from speech segments of specific phonetic identities and these densities correspond to identifiable physical states of the speech production mechanism. The justification comes from the hypothesis that different neural units in human brains are sensitive to different sounds corresponding to different densities in the feature space. When an utterance is heard, a sequence of neural units will fire in order. Another layer of neural units detecting the pairwise orders of the firings (the transitions from one state to another) will in turn fire to yet another layer of neural units which construct a static representation of the dynamics of a short utterance of speech.

3. Supervised learning of a mixture of phonemic Gaussian densities

Anyone with experience in vector quantization will feel uncomfortable about the fact that as a consequence of vector quantization, the feature vectors falling within a tessellation are often widely scattered instead of forming a compact cluster. It is therefore meaningless to associate them with any physical entities. We want to show in this section how to structure the feature space into a mixture of Gaussian densities each of which corresponds to a phonetic state. We assume that one is able to isolate segments of a speech waveform corresponding to specific phonemes by boot-strapping and/or identifying islands in the waveform with relatively little spectral variations. It is extremely difficult to locate the exact boundary of two neighbouring phones in an utterance because the transition can be gradual and the exact boundary simply does not exist. However, for the purpose of extracting feature vectors from the waveform to train the density of a phoneme, one does not need to know exactly where the effect of a phone ends and where the next phone starts to pick up in an utterance. One can be conservative and give up waveform segments belonging to the grey area as far as density training is concerned. One must note that the isolation of phonetic segments of waveform was performed only in the training stage when we are estimating the phonetic densities of the feature space and in this paper, this was performed manually. Such phonetic segmentation is no longer needed in the recognition stage. If one has in hand a collection of feature vectors $x_1, \ldots, x_n$ from the same class, assuming the class conditional probability density a Gaussian one - $N(\mu, \Sigma)$, Considering the fact that some phonemes can be short in duration, the number of feature vectors one can derive to train their densities with can be insufficient. It is highly desirable to reduce the dimension of the feature space without severely degrading the recognizability of the target objects before phonetic densities are trained. One can perform a K-L transform based on the training feature vectors and then perform a recognition among the training utterances by dynamic programming in feature spaces of increasing dimensions spanned by the most discriminative eigenvectors of the transform. The discriminative power of an eigenvector is not determined...
by its associated eigenvalue but by the recognition rate when that eigenvector is used as the only feature characterizing the objects. The selection of features was performed by examining their combinations (with back-tracking) that yielded a good recognition rate and yet with low dimensionality.

4. Speech production model as a matrix of state transition probabilities

Each phonetic density is assumed to correspond to a state of the speech production mechanism. Since the densities are Gaussian, each feature vector has a finite probability of belonging to each density. Any short utterance, whether training or testing, will have an \( m \times n \) matrix of state transition probabilities serving as its representative. The \((i,j)^{th}\) element of such a matrix, \( q_{ij} \), is the estimated probability of transition from state \( i \) to state \( j \) in the course of utterance production. Each pair of consecutive feature vectors in the observed sequence of feature vectors \{ \( \mathbf{a}_1, \ldots, \mathbf{a}_p \) \} derived from a short utterance has a contribution to each element of the matrix. Assume that the probability of vector \( \mathbf{a}_i \) produced in state \( i \) is \( p_{0i} \), and that of \( \mathbf{a}_{i+1} \) in state \( j \) is \( p_{nj} \). The probabilities are computed according to:

\[
p_{ij} = p(\mathbf{a}_i | \mathbf{a}_j) = \frac{p(\mathbf{a}_i | \mathbf{a}_j)P(\mathbf{a}_j)}{\sum_{l=1}^{n} p(\mathbf{a}_i | \mathbf{a}_l)P(\mathbf{a}_l)} \text{ for } k = 1, m-1
\]  

where \( p(\mathbf{a}_i | \mathbf{a}_j) \) is the class-conditional Gaussian pdf and \( P(\mathbf{a}_j) \) is the a priori probability of finding a phoneme in an utterance belonging to the density associated with state \( k \). The contribution of the \( \{ \mathbf{a}_i, \mathbf{a}_j \} \) pair to the \((i,j)^{th}\) element of the matrix of state transition probabilities \( Q = [q_{ij}] \) is defined as \( p_{0ij}q_{nj} \) and

\[
q_{ij} = \sum_{l=0}^{m-1} p_{0il}q_{nlj}
\]  

The summation extends from 0 to \( T \) because each utterance is assumed to start from and terminate at a state of silence - \( o_0 \). \( \mathbf{a}_0 \) and \( \mathbf{a}_{m+1} \) are fictitious. They correspond to the unobserved feature vectors of silence before and after the utterance. They are introduced for the simplicity of equations (2) and (4) only and they are defined as:

\[
p(\mathbf{a}_1 | \mathbf{a}_0) = p(\mathbf{a}_0 | \mathbf{a}_{m+1}) = 1, \quad \text{if } k = 0
\]

\[
= 0, \quad \text{otherwise}
\]  

The matrix is normalized by dividing each element \( q_{ij} \) by \( T + 1 \) so that

\[
\sum_{i=0}^{m-1} \sum_{j=0}^{m-1} q_{ij} = 1
\]  

5. Distortion measure

Because the objects of recognition are now expressed in terms of matrices of probabilities, template matching distortions can be measured in terms of the differences in information contained in the matrices of probabilities. If we linearize all the \( m \times n \) matrices of transition probabilities into vectors from now on by denoting the \((i,j)^{th}\) matrix element \( q_{ij} \) as \( q_i \), where \( i = m(j-1) + j \), the distortion from matrix \( \mathbf{R} = [r_{ij}] \) to matrix \( \mathbf{Q} = [q_{ij}] \) is defined as [3]:

\[
d(\mathbf{R}, \mathbf{Q}) = \sum_{l=1}^{L} \ln \frac{q_i}{r_i}
\]  

where \( L = m^2 \). The centroid of a collection of \( L \) utterances, each represented as \( p_0 = [p_{01, \ldots, p_{0L}}] \), \( p_{0L} \) is the column of probabilities \( C = [c_{ij}] \), from which, the total distortion of the collection is minimized. With distortions from one utterance from another so defined, the centroid can be easily shown to be:

\[
c_l = \frac{\sum p_{0ij}q_{lj}}{\sum p_{0ij}q_{lj}}, \quad l = 1, L
\]  

where \( L \) and \( L \) are defined as before while \( l \) stands for the \( l^{th} \) utterance of the collection.

6. Experimental verification of the effectiveness of Static Models for recognition purposes

In the following sections, we'll report our experimental findings on the recognition rates and time requirements for speaker independent isolated syllable recognition by means of dynamic programming with and without vector quantization and by means of hidden Markov models. There were two databases which we experimented the various recognition algorithms with, viz., the /a/-set and the ten digits. The training data of the /a/-set consisted of 16 male and female speakers as unknowns. Each utterance was sampled at 22 KHz and segmented into half overlapping and Hamming windowed frames of 512 samples (23 ms) before DFT to yield 20 critical bandpass filtered [4] spectral features. A K-L transformation was then performed using just the spectral features derived from the training data. Twelve most discriminative principle components were selected to span the feature space. We arrived at this decision by first dividing the training utterances into two groups. A group of utterances from 30 speakers were used as references and the rest from the other 10 speakers as unknowns. A new database of the /a/-set of Cantonese syllables, just like the the /a/-set of the Latin alphabet, is an acid test to any speech recognition algorithm (and the suitability of the speech model as well) because what differentiates the syllables from each other lies in the initial portion of each syllable which is usually short in duration.

EUROSPEECH '89, Paris, France, September 1989

1463
weak in energy. Secondly, their phonetic structures are so simple that it is quite easy to isolate the initial consonants from the final vowel /a/ which helped us tremendously to represent the feature space as a mixture of Gaussian densities.

Due to the fact that we did not have sufficient training data for some of the consonant densities, we simply assumed that all 15 consonants have the same covariance matrix (but different mean vectors). On the other hand, the vowel /a/ had its own covariance matrix throughout the experiment because of the abundance in /a/ samples. The a priori probabilities \( P(a_i) \) of each phonemic density was estimated according to the relative observed occurrence of feature vectors of each phonetic class. The feature space was thus considered a mixture of 16 Gaussian phonetic densities.

The second database was composed of the ten Cantonese digits (0-9) which again are mono-syllables. They were spoken by the same 40 speakers above for training data and the same 10 speakers for testing data. K-L transform was performed and the feature space was reduced to a dimension of 12 spanned by the principle components selected in a similar manner as for the /a/-set above. The reduced feature space comprised of a mixture of 14 Gaussian phonetic densities. The isolation of phonemic feature vectors and the training of phonetic densities were similar to those with the /a/-set.

Upon recognition, Sakoe and Chiba’s DP algorithm was programmed with an adjustment window width \( w = 18 \) and slope constraint \( p = 1 \). When vector quantization of the feature space was performed for DP and HMM, the clustering algorithm of Isooda was employed to create the codebooks. The technique of left-to-right, no state skipping HMM was implemented with Viterbi’s algorithm to determine the probability of a model generating the utterance. Since all utterances are mono-syllables, we used 5-states models for all discrete hidden Markov models.

Table 1 summarizes the recognition rates and time requirements of the /a/-set for (a) DP with 640 (=40x16) references in 20 dimensional feature space without VQ, (b) DP with 640 references and with 283 codewords VQ in 20 dimensional feature space, (c) 16 discrete HMM with 283 codewords VQ in 20 dimensional feature space, (d) 16 static models with 16 phonetic states in 12 dimensional feature space, (e) 640 static models (40 per syllable) with 16 phonetic states in 12 dimensional feature space.

Table 2 summarizes the recognition rates and time requirements of the ten digits for (a) DP with 400 (=40x10) references in 20 dimensional feature space without VQ, (b) DP with 400 references and with 327 codewords VQ in 20 dimensional feature space, (c) 10 discrete HMM with 327 codewords VQ in 20 dimensional feature space, (d) 10 static models with 14 phonetic states in 12 dimensional space, (e) 400 static models (40 per digit) with 14 phonetic states in 12 dimensional space.

The system on which the experiments were conducted was a Compaq Deskpro 386/20 with a 20 MHz 80387 co-processor run under MS-DOS.

<table>
<thead>
<tr>
<th>Case</th>
<th>Experiment condition</th>
<th>Recognition rate (%)</th>
<th>Ave. recognition time required per syllable (Sec.)</th>
<th>Ave. processing time per syllable (Sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha )</td>
<td>640 Ref. - DP</td>
<td>64</td>
<td>327</td>
<td></td>
</tr>
<tr>
<td>( \beta )</td>
<td>640 Ref. - DP with 283 codewords VQ</td>
<td>50</td>
<td>109</td>
<td>5</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>16 Ref. - HMM, 283 codewords VQ</td>
<td>39</td>
<td>0.54</td>
<td>5</td>
</tr>
<tr>
<td>( \delta )</td>
<td>16 Static Models</td>
<td>67</td>
<td>0.14</td>
<td>2.9</td>
</tr>
<tr>
<td>( \epsilon )</td>
<td>640 Static Models</td>
<td>71</td>
<td>5.6</td>
<td>2.9</td>
</tr>
</tbody>
</table>

Table 1. Recognition rates and timing requirements for recognition of the /a/-set of Cantonese syllables

<table>
<thead>
<tr>
<th>Case</th>
<th>Experiment condition</th>
<th>Recognition rate (%)</th>
<th>Ave. recognition time required per digit (Sec.)</th>
<th>Ave. processing time per syllable (Sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha )</td>
<td>400 Ref. - DP</td>
<td>99</td>
<td>155</td>
<td></td>
</tr>
<tr>
<td>( \beta )</td>
<td>400 Ref. - DP with 327 codewords VQ</td>
<td>99</td>
<td>46</td>
<td>4</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>10 Ref. - HMM, 327 codewords VQ</td>
<td>97</td>
<td>0.32</td>
<td>4.0</td>
</tr>
<tr>
<td>( \delta )</td>
<td>10 Static Models</td>
<td>96</td>
<td>0.083</td>
<td>2.2</td>
</tr>
<tr>
<td>( \epsilon )</td>
<td>400 Static Models</td>
<td>98</td>
<td>3.32</td>
<td>2.2</td>
</tr>
</tbody>
</table>

Table 2. Recognition rates and timing requirements for recognition of ten digits in Cantonese

7. Discussion

From Tables 1 and 2, we observe that the static models with/without inhibition, performed quite competitively compared with DP and HMM in terms of speed and recognition rate. Furthermore, they have the additional advantages that the models are intuitively straightforward and practically less sensitive to the inadequacy of training data.
The static model of a short segment of speech tries to capture the average state transition probabilities in the course of speech production while ignoring the timing information about the transitions. Naturally, this is expected to be suitable for short segments of speech only. We have implemented static models for monosyllables but we believe that static models in terms of state transitions will be suitable for isolated words in English as well. When compared with the hidden Markov model which in theory at least, can accommodate an utterance of any length, both models try to portray the same thing, but from different angles. The major difference between the two models lies perhaps in the fact that static models assume that states are not only phonetically identifiable, but also have estimated probability densities while hidden Markov models do not even have the states phonetically identified. Judging by the recognition rates of hidden Markov models reported in [6] where each state assumed a Gaussian probability density in the feature space with the covariance matrix equal to an identity matrix (apart from a factor), they performed better than the discrete HMM implemented according to [2]. Our experiments with HMM with Gaussian phonetic states seemed to confirm this finding although the implementation is somewhat different, which led to one conjecture, viz., a crude estimate is better than no estimate. This is exactly the explanation we'll offer for the observation that static models are more robust than HMM when there is insufficient training data for the models.

If we have \( R \) references and every utterance is of length \( T \), the time complexity of Dynamic Programming is \( O(R \times T^2) \), while that of Hidden Markov Models is \( O(R \times T) \) but that of Static Models is \( O(R + T) \).

8. Conclusion

We believe that the static models of short-timed speech segments offer a competitive alternative to represent the dynamics of speech for recognition purposes. Being static models, they lend themselves readily to the manipulation of classical pattern recognition techniques and that makes this approach very attractive.

9. References


EUROSPEECH '89, Paris, France, September 1989