A FAST CALCULATION METHOD IN LVCSRS BY TIME-SKIPPING AND CLUSTERING OF PROBABILITY DENSITY DISTRIBUTIONS

Seiichi NAKAGAWA and Yukihisa HORIBE

Dept. of Information and Computer Sciences,
Toyohashi University of Technology,
1-1 Hibarigaoka, Tempaku-cho, Toyohashi-shi, Aichi-ken, 441-8580 JAPAN

Abstract

In this paper, we propose a rapid output probability calculation method in HMM based large vocabulary continuous speech recognition systems (LVCSR). This method is based on time-skipping of calculation, clustering of probability density distributions, and pruning of calculation. Only distributions covering input feature vectors with high probabilities are used to calculate output probabilities strictly, and representative distributions for other distributions are used to calculate them approximately. Here a skipping method for likelihood calculation is adopted in the time domain. Using the rapid calculation method by clustering of probability density distributions, the recognition time in a LVCSR system was reduced by about 40%. Using a pruning method of likelihood calculations on the way, it was further reduced by 25%. Finally, using time-skipping, the calculation time, furthermore, was reduced by 15% without compromising recognition accuracy.

1. Introduction

In speech recognition systems, computing the respective likelihood of the acoustic models accounts for much of system computations. For speech recognition systems to become more useful, these computations must be done faster while keeping the memory required to store the models as small as possible and the recognition accuracy as high as possible.

One commonly used approach that reduces computational cost is selection of calculated distributions.

In [1], using vector quantization (VQ) of the input feature vector identifies a subset of Gaussian neighbors. And only the selected Gaussian neighbors instead of all the Gaussians are used to compute the likelihood.

In [2], a tree-structured of a set of probability density functions (pdf) was proposed. A leaf node corresponds to a pdf, and a non-leaf node corresponds to a cluster composed of pdfs. Each cluster is attached with an approximating pdf. In the recognition, the likelihood set is calculated by searching the tree.

In [3], a hierarchical method was proposed. The algorithm first estimates output probabilities using rough HMMs with a few distributions, and then re-estimates those of the probable states using detailed HMMs with many distributions.

In [4], the likelihoods of code vectors in the codebook clustered by VQ are calculated for all probability density functions with full Gaussian matrices in advance, and stored them in the table. In the recognition, the likelihood calculation looks-up likelihoods in the column of the nearest code vector for the input vector in the table.

In this paper, we propose three methods. In the first, a subset of probability density distributions is selected for an input feature vector, and then these distributions are used to calculate the likelihoods strictly. For unselective distributions, the approximated likelihoods assigned to a corresponding cluster are used. In the second method, the likelihood calculation is pruned on the way whenever the likelihood becomes worse than a given threshold. In the third method, time skipping for stable frames is employed. If the change of acoustic features over successive frames is small, the likelihood calculation is skipped and the likelihood for a past frame is reused.

2. OVERVIEW OF LVCSR SYSTEM

We use a word N-gram-based large vocabulary continuous speech recognition system as a baseline system. Our system is implemented based on a standard one-pass beam search algorithm in the first pass [5] and generates N-best candidates. The second pass selects the most probable candidate from the N-best using a tri-gram language model. The search algorithm employs a tree-organized lexicon in which each node corresponds to a sub-word unit. The search process employs the 1-best approximation for the different word contexts in order to reduce computational cost, instead of a coping node to hold the cumulative likelihoods of different word contexts that remain within a beam width in the search process.

We employ a factoring method (language model lookahead [6]) in order to effectively improve the search process. The basic idea of the factoring is to incorporate the language model into the pruning process of the time-synchronous search algorithm with a tree-organized lexicon. Factored probabilities assigned for each node are calculated based on the word unigram model.

Our system employs the segmental unit input HMM
3. FAST LIKELIHOOD CALCULATION METHOD

One of the most computationally expensive operations in the HMM-based speech recognition system is the calculation of the state likelihoods. To reduce the computational effort of the state likelihood calculation, we developed a rapid state likelihood calculation method based on vector quantization (VQ).

The calculation method is as follows:

1. All probability density distributions in acoustic models (HMMs) are clustered using VQ. The VQ is performed for mean vectors of all Gaussian distributions. Clustering results are stored in a table in order to relate cluster (VQ cell) with probability density distributions.

2. Each cluster is assigned with an approximation distribution (mean vector and covariance matrix). The mean vector is a code vector corresponding to the cluster, and the covariance matrix is the average of covariance matrices belonging to the cluster.

3. Euclidean distances between an input feature vector and clusters are calculated.

4. The likelihoods for distributions in a cluster whose distance is less than a pre-defined distance, are exactly calculated using each original probability density distribution.

5. The likelihoods for distributions in a cluster which is a considerable distance from the input feature vector are calculated using the approximation distribution assigned to the cluster.

In the very distant clusters, the likelihood calculation is only one time for each cluster. As a result, the number of distribution likelihood calculations are reduced.

4. Likelihood Calculation Pruning

Acoustic likelihood is calculated by the product of vectors (the difference between HMM's mean vectors and input feature vectors) and the inverse covariance matrix. And the likelihood is obtained from accumulating of every dimension's likelihood.

Whenever the accumulated likelihood on the way of every dimension exceeds a given threshold, the calculation is pruned and the likelihood is set to a constant value.

A pruning threshold is given as follows:

$$th_i = \mu_{i,th} - 3\sigma_{i,th}$$  \hspace{1cm} (1)

where \(th_i\) is a pruning threshold for dimension \(i\), \(\mu_{i,th}\) is the mean of accumulated likelihood up to dimension \(i\), and \(\sigma_{i,th}\) is the variance of acoustic likelihood for dimension \(i\), respectively.

5. Time Domain Skipping

When doing speech recognition, if the fluctuation of input feature vectors is within a narrow range, the fluctuation of the likelihoods calculated by these vectors will also fall within a narrow range. Then we propose that when there is such a narrow range, a current frame likelihood is replaced by the likelihood calculated before. But for a fluctuating input feature vector, the likelihood is calculated in this time. Figure 1 shows the calculation process of this idea.

![Figure 1: Multi-step time domain calculation skipping](image)

In Figure 1, a box shows a cluster of probability density distributions that are sorted by distances between all distributions and an input feature vector at \(t/N\). And \(d_{it}\) shows a distance between an input feature vector \(x_i\) at \(t\) and \(x_i\) at \(t-1\). \(\Delta d\) shows the averaged VQ distortion, and \(t_1\) shows the last time when likelihoods in \(D_i\) are calculated. \(w\) shows a parameter controlling the calculation skipping rate in \(D_i\), provided that \(w \geq w_{i+1}\). And \(i\) shows the boundary rank of \(D_i\) and \(D_{i+1}\). \(N\) is the number of all clusters (i.e., the codebook size of VQ), and \(M\) is the number of calculation skipping levels.

The likelihood calculation skipping method is as follows:

1. All clusters are sorted by distance between an input feature vector \(x_i\) and all cluster centroids at \(t = 0\). The sorted cluster set is divided into \(M\) subsets.

2. When likelihoods are calculated at an arbitrary time \(t\), the comparison of \(d_{it}\) and \(w\Delta d\) is performed in the descending order of \(i\) \((i = M, M - 1, \cdots, 1)\).

3. If \(d_{it} \geq w\Delta d\), likelihoods of the input vector for distributions involved in \(D_i\) are calculated, and \(t_1\)
Table 1: Result by lump vector quantization (LQ) (20000 words, bigram)

<table>
<thead>
<tr>
<th></th>
<th>calculated dist.rate[%]</th>
<th>×RT</th>
<th>Cor[%]</th>
<th>Acc[%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>100.0</td>
<td>7.7</td>
<td>84.9</td>
<td>80.9</td>
</tr>
<tr>
<td>NC 100</td>
<td>56.8</td>
<td>4.7</td>
<td>85.4</td>
<td>81.2</td>
</tr>
<tr>
<td>NC 80</td>
<td>48.3</td>
<td>4.6</td>
<td>84.3</td>
<td>79.8</td>
</tr>
<tr>
<td>Thr 3500</td>
<td>51.3</td>
<td>4.5</td>
<td>84.8</td>
<td>80.7</td>
</tr>
<tr>
<td>Thr 4000</td>
<td>55.0</td>
<td>4.5</td>
<td>84.9</td>
<td>80.8</td>
</tr>
</tbody>
</table>

is updated to \(t\). By way of exception, if likelihoods in \(D_M\) are calculated newly, the system calculates distances between \(x\) and all cluster centroids, and re-sorts the clusters by distances and set to \(t(i = 1, 2, \ldots, M)\).

4. If \(d_1 < w\), likelihoods in sub-cluster \(D_i\) calculated at \(t\) are reused.

6. Experimental Results

6.1. Experimental Condition

The baseline system is a one-pass beam-search decoder (see Section 2). The vocabulary size is 20,000 words. The language model used in the first pass is word-bigram trained by 3 million sentences. The feature parameters consist of 42 dimensions [7, 8]: 20 dimensional LPC mel cepstrum segmented from 4 successive frames, \(\Delta\) 10 dimensions calculated over 9 successive frames, \(\Delta\Delta 10\) dimensions and \(\Delta\Delta\) energies. The acoustic models are syllable unit HMMs that have 5 states, 4 densities and 4 Gaussian mixture models per density with full covariance matrices. There are 114 syllables in Japanese. So, the number of Gaussian density distributions is 1824. The computation cost corresponds to more than 20000 Gaussian distributions with diagonal covariance matrices. The sampling frequency is 12 KHz and the frame shift length is 8 msec. The codebook size of VQ-based clustering is 256. The test set is 100 sentences for newspaper corpus uttered by 9 male speakers. The calculation time is measured on Alpha21264 677 MHz workstation (768 MByte memory) in real time ratio for the first pass.

6.2. Distribution Selection Method’s Result

At first, we show experimental results by the rapid likelihood calculation method described in Section 3. Table 1 shows results in the case of an input feature vector regarded as one vector (lump vector quantization: LQ). Table 2 shows results in the case of the input feature vector separated into sub-vectors (e.g., Cepstrum, \(\Delta\) Cep, \(\Delta\Delta\) Cep and Energy) (separate vector quantization: SQ).

In these tables, the “calculated dist. rate[\%]” means the percentage of distributions used by likelihood calculation per frame. “×RT” means the real time factor of recognition processing time. “Cor” and “Acc” mean word correct and word accuracy rates, respectively. In Table 1, “NC n” means the n nearest neighbor clusters against an input feature vector and the selected clusters calculate the likelihoods strictly. “Thr \(\theta\)” means the threshold \(\theta\) for selecting clusters in which distributions are used for calculating exact likelihoods (for example, “Thr 3500” calculates likelihoods of distributions in a cluster in which the distance between an input vector and a code vector is less than 3500). In Table 2, “NC a-b-c” means the a (b, or c) nearest neighbor clusters for the first sub-vector (for example, for “NC 64-80-128,” the 64 nearest neighbor cepstrum clusters are selected, \(\Delta\) Cep ones are 80 and \(\Delta\Delta\) Cep ones are 40, respectively).

As shown in Table 1, using the distribution selection method, the calculated distribution rate was reduced by about 45% compared with the baseline system without degrading the recognition accuracy and the recognition time was reduced by 40%. As shown in Table 2, the results based on separate vector quantization were worse than those based on lump vector quantization.

Table 3 shows results that approximation likelihoods are replaced with a constant value (equivalent to a likelihood of minus infinity). This approach was used in order to reduce the computation cost for calculating likelihoods for approximation distributions.

As shown in Table 3, calculated distribution rate and the recognition time were further reduced without loss of recognition accuracy (55% and 45%, respectively).

6.3. Likelihood Calculation Pruning Result

We show experimental results of implementation of the likelihood calculation pruning in Table 4. Baseline corresponds to NC100 in Table 3.

In Table 4, “thres. constant” denotes that a pruning threshold is used by a constant value, “various thres.” denotes pruning thresholds defined as Eq.(1) (after learning by 100 sentences). And “5 dim. fix” denotes that the pruning threshold for the first 5 dimensions is used by a constant pruning threshold, and the following dimensions are based on Eq.(1). As a result, using var-
Table 4: Result by likelihood calculation pruning (20000 words, bigram)

<table>
<thead>
<tr>
<th></th>
<th># of product [%]</th>
<th>×RT</th>
<th>Cor [%]</th>
<th>Acc [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>100.0</td>
<td>4.4</td>
<td>85.4</td>
<td>81.2</td>
</tr>
<tr>
<td>thres. constant</td>
<td>87.7</td>
<td>4.1</td>
<td>85.4</td>
<td>81.2</td>
</tr>
<tr>
<td>various thres. (5 dim. fixed)</td>
<td>44.2</td>
<td>3.3</td>
<td>85.0</td>
<td>81.0</td>
</tr>
</tbody>
</table>

Table 5: Result by 3 level time domain calculation skipping with LQ (20000 words, bigram)

<table>
<thead>
<tr>
<th></th>
<th>calculated dist. rate [%]</th>
<th>×RT</th>
<th>Cor [%]</th>
<th>Acc [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>100.0</td>
<td>7.7</td>
<td>81.9</td>
<td>80.9</td>
</tr>
<tr>
<td>1.5-3.6-5.0</td>
<td>30.8</td>
<td>3.8</td>
<td>84.5</td>
<td>80.7</td>
</tr>
<tr>
<td>1.5-5.6-5.0</td>
<td>25.0</td>
<td>3.6</td>
<td>83.8</td>
<td>80.0</td>
</tr>
<tr>
<td>2.6-3.6-5.0</td>
<td>29.3</td>
<td>3.8</td>
<td>84.3</td>
<td>80.7</td>
</tr>
<tr>
<td>2.6-5.6-5.0</td>
<td>23.9</td>
<td>3.6</td>
<td>84.7</td>
<td>81.2</td>
</tr>
</tbody>
</table>

Table 6: Result by 3 level time domain calculation skipping with SQ (20000 words, bigram)

<table>
<thead>
<tr>
<th></th>
<th>calculated dist. rate [%]</th>
<th>×RT</th>
<th>Cor [%]</th>
<th>Acc [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>100.0</td>
<td>7.7</td>
<td>81.9</td>
<td>80.9</td>
</tr>
<tr>
<td>1.5-3.6-5.0</td>
<td>23.4</td>
<td>4.3</td>
<td>82.3</td>
<td>78.9</td>
</tr>
<tr>
<td>1.5-5.6-5.0</td>
<td>20.1</td>
<td>4.2</td>
<td>82.0</td>
<td>78.8</td>
</tr>
<tr>
<td>2.6-3.6-5.0</td>
<td>21.6</td>
<td>4.2</td>
<td>81.6</td>
<td>78.2</td>
</tr>
<tr>
<td>2.6-5.6-5.0</td>
<td>18.2</td>
<td>4.1</td>
<td>81.8</td>
<td>78.5</td>
</tr>
</tbody>
</table>

6.1. Time Skipping Result

We show experimental results of implementation of the time domain likelihood calculation skipping method described in Section 5. The experiment was performed by 3 level (M = 3) time domain calculation skipping. Sub-clusters, in which distributions are calculated strictly, are $D_1$ and $D_2$. The other sub-cluster, in which the likelihood uses an approximation likelihood, is $D_3$.

Tables 5 and 6 show results in the case of LQ and SQ, respectively. Table 5 is based on "Thr 3500" in Table 1 and Table 6 is based on "NC 64-80-10" in Table 2. In Tables 5 and 6, the expression, for example, "1.5-3.6-5.0," denotes $w_1 = 1.5$, $w_2 = 3.0$ and $w_3 = 5.0$ in Figure 1, respectively.

As shown in Table 5, the calculated distribution rate was reduced about 75% compared with the baseline system and 45% compared with no time domain calculation skipping without compromising recognition accuracy. The recognition time was reduced by about 50% compared with the baseline system and 15% compared with no time domain calculation skipping. In the case of SQ, although the calculated distribution rate was reduced remarkably, the calculation time reduction and recognition accuracy became worse in comparison with the case of LQ.

7. CONCLUSION

We proposed three fast output probability calculation methods. The first method is based on clustering of probability density distributions. Distributions in clusters covering input feature vector with high probabilities are used to calculate the likelihoods strictly. In the other clusters, the likelihood uses an assigned approximate likelihood or a constant value. The second is based on a pruning method for likelihood calculation. We further proposed the multi-level time domain skipping method of likelihood calculation. Using the fast calculation method by clustering, the recognition time in the LVCSR system was reduced by about 40%. By integrating with the pruning method, it was reduced by 55%. Then, using the 3 level time domain skipping method, the recognition time, furthermore, was reduced by 15% without diminishing recognition accuracy. Finally, we performed the recognition time of $2.8 \times$ RT.

8. References