DYNAMIC GAUSSIAN SELECTION TECHNIQUE FOR SPEEDING UP HMM-BASED CONTINUOUS SPEECH RECOGNITION†

Jun Cai[1,2], Ghazi Bouselmi[1], Dominique Fohr[1], Yves Laprie[1]

[1] Groupe Parole, LORIA-CNRS & INRIA, BP 239, 54600 Vandoeuvre-les-Nancy, France
[2] Dept. of Cognitive Science, Xiamen Univ., 361005 Xiamen, China

ABSTRACT

A fast likelihood computation approach called dynamic Gaussian selection (DGS) is proposed for HMM-based continuous speech recognition. DGS approach is a one-pass search technique which generates a dynamic shortlist of Gaussians for each state during the procedure of likelihood computation. The shortlist consists of the Gaussians which make prominent contribution to the likelihood. In principle, DGS is an extension of the technique of Partial Distance Elimination, and it requires almost no additional memory for the storage of Gaussian shortlists. DGS algorithm has been implemented by modifying the likelihood computation module in HTK 3.4 system. Results from experiments on TIMIT and HIWIRE corpora indicate that this approach can speed up the likelihood computation significantly without introducing apparent additional recognition error.

Index Terms— Gaussian selection, fast likelihood computation, hidden Markov models, speech recognition

1. INTRODUCTION

Most state-of-the-art large vocabulary continuous speech recognition (LVCSR) systems use continuous density HMMs (CDHMMs) as the underlying technology to perform acoustic modeling of speech signals. In a typical HMM-based LVCSR system, the number of model states ranges from 2000 to 6000, each of which is a Gaussian mixture model (GMM) with typically 8 to 64 multi-dimensional Gaussian distributions. For each input frame, the output likelihoods should be computed for every active state. The state likelihoods estimation is computationally intensive and typically takes about 30% to 70% of the total recognition time [1]. Therefore, this kind of likelihood-based statistical acoustic modeling is so time-consuming that the recognition is several times slower than real time.

Many different algorithms have been proposed to speed up the likelihood computation, the most popular ones are in the category of VQ-based Gaussian selection [1, 2]. A typical VQ-based Gaussian selection technique can lead to significant additional memory requirements. To overcome this problem, we propose an alternative scheme, dynamic Gaussian selection (DGS), based on the partial distance elimination (PDE) framework [3]. DGS aims at maintaining recognition accuracy with no additional memory overhead.

The paper is organized as follows. The Gaussian selection techniques are reviewed and analyzed in Section 2. Section 3 describes a nearest neighbor approximation technique based on PDE. In Section 4, DGS is presented in detail, which uses the extended PDE method to compute the log likelihood of each GMM on several dynamically selected Gaussian components. This technique is tested and evaluated with English continuous speech corpus TIMIT as well as on the French LVCSR project HIWIRE. Experimental results are presented in Section 5. It is concluded in Section 6 that DGS is an efficient technique for fast likelihood computation, and combining DGS with other optimization techniques can give rise to satisfactory real-time performance.

2. VQ-BASED GAUSSIAN SELECTION

In CDHMM-based LVCSR systems, the output likelihood of an HMM state S for a given observation feature vector, $\boldsymbol{x}_n$, can be expressed as a Gaussian mixture model (GMM), which is a weighted sum of multivariate Gaussian densities [4]. Analogous to series expansions used to approach complex functions, the Gaussian mixture model is an approximation mechanism to compute various probability distributions. Usually, for a given observation vector, only a few Gaussians, or just one Gaussian in some cases, will dominate the likelihood of a GMM. So, the computation of a Gaussian mixture can be truncated to one or a small number of Gaussians provided that the approximation accuracy is guaranteed. This basic understanding led to the idea of Gaussian selection. Many different algorithms have been proposed to decide which Gaussians in the mixture dominate the likelihood [1, 2, 5]. The set of selected Gaussians is usually called the shortlist of the mixture model.

The most commonly used Gaussian selection technique is the VQ-based Gaussian selection. Though many different

† Thanks to the State Scholarship Fund of China and the 985 Innovation Project on Information Technology of Xiamen Univ. (2004-2007) for funding.

methods can be used to implement this technique, the key idea is to partition the acoustic space into a number of sub-spaces, called clusters, each of which being represented by a centroid. After training the HMM models, for each state-centroid pair (S, C) a shortlist of Gaussians of S is formed according to a certain distortion measure between the centroid and the Gaussians. During recognition, each observation vector is mapped to a centroid C first, and then the likelihood of each state S is computed only on the shortlist corresponding to the (S, C) pair. Therefore, VQ-based Gaussian selection technique is essentially a two-pass search. In the first pass, a rough model is used to determine the location of the observation vector in the acoustic space, and a shortlist is correspondingly decided for each GMM. In the second pass, the input vector is computed on the derived shortlists and thus the likelihoods of all GMMs are evaluated. Though there is a computational saving due to Gaussian selection, extra memory requirement is introduced because the use of the shortlists implies a significant memory overhead.

3. PARTIAL DISTANCE ELIMINATION

A nearest-neighbor approximation of likelihood, which requires no additional memory for shortlists, can be used as a fast likelihood computation technique to reduce the computational overhead [3]. Instead of computing the likelihood by summing across all mixtures, the maximum mixture probability is taken as the state likelihood. This nearest-neighbor approximation can be expressed as:

$$\log\left[p\left(\boldsymbol{x}_n|S\right)\right] \approx \max_{1\le k\le M}\left\{\log\left(Z_k\right)-\frac{1}{2}\sum_{q=1}^{N}\frac{\left(x_{nq}-\mu_{kq}\right)^2}{\sigma_{kq}^2}\right\} \qquad (1)$$

M represents the number of mixture components for state S; Z_k is a constant for each Gaussian and can be computed before recognition. N is the dimension of the feature vector, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean and covariance matrix for the kth Gaussian density in state S.

This nearest-neighbor search problem can be thought as a vector quantization (VQ) codebook search problem, where the Gaussians in that state are the codewords and the distortion measure is given on the right side of (1). Let $D_k(\boldsymbol{x}_n|y)$ denote this distortion measure for the codebook search (here, y is the codebook), then

$$D_k\left(\boldsymbol{x}_n|y\right) = \log\left(Z_k\right)-\sum_{q=1}^{N}\left(x_{nq}-\mu_{kq}\right)^2\frac{1}{2\sigma_{kq}^2} \qquad (2)$$

In the codebook search, we must maximize $D_k(\boldsymbol{x}_n|y)$. By inspecting (2), we can find that the right-hand side is actually a weighted Euclidean distance measure, and the computation of the distortion measure $D_k(\boldsymbol{x}_n|y)$ is performed recursively on each element of the observation vector. Furthermore, with the progress of each recursion, the value of $D_k(\boldsymbol{x}_n|y)$ decreases monotonically. Therefore, a technique called partial distance elimination (PDE) can be used to reduce the computational complexity. The algorithm starts by accumulating all the Euclidean distances and deriving the distortion measure for the first Gaussian of the mixture, according to (2). The value of this distortion measure is used to initialize the maximum distortion D_{max}. For many other Gaussians in the mixture, $D_k(\boldsymbol{x}_n|y) < D_{max}$. For such a Gaussian the intermediate value of the distortion will drop below D_{max} at a certain element j(j<N). So the recursion for evaluating such a $D_k(\boldsymbol{x}_n|y)$ can be stopped at the jth element. This means that the computation of a part of the Euclidean distances is eliminated in searching the codebook. This kind of PDE algorithm can increase the efficiency of the codebook search, and therefore speed up the likelihood approximation.

The efficiency of PDE technique relies heavily on how quickly a high estimate of D_{max} is obtained. A high efficiency can easily be accomplished by exploiting the high correlation between adjacent observation vectors. Using the previous "best" Gaussian as the prediction of the current "best" one, and computing the distortion of this Gaussian first could result in a high D_{max} immediately and speed up the elimination process for the codebook search. This method is called the "best mixture prediction" (BMP).

Another algorithm, called feature element reordering (FER), also known as feature component reordering (FCR), can complement PDE and BMP techniques for further reduction in likelihood computation. On right-hand side of (2), the contribution of some of the feature elements to the value of the Gaussian is greater than others. The idea of FER is to shuffle the elements in the computation of (2) in such a way that elements contributing prominently in the weighted Euclidean distance are computed first followed by the elements contributing less. With FER, PDE process is further speeded up because the Gaussians whose probabilities are smaller than the current D_{max} are eliminated as early as possible. In FER, the reorder rule is usually learned offline from a portion of the development set and remains fixed during recognition.

Though combining PDE framework with both BMP and FER is reported as an efficient technique [3], this nearest-neighbor search technique uses only the probability of one Gaussian in the mixture to approximate the whole likelihood. The contribution of all other Gaussians to the likelihood is omitted in the approximation. Thus, the resolution of the HMM model is degraded, as well as the recognition accuracy.

4. DYNAMIC GAUSSIAN SELECTION

To overcome the above problem of PDE method, we propose an alternative scheme called dynamic Gaussian selection (DGS). It aims at utilizing the advantages of PDE, BMP and FER techniques to speed up the computation of likelihood, while introducing less degradation of recognition accuracy than PDE. In DGS scheme, a shortlist of Gaussian components is selected to compute the likelihood instead of using only the maximum Gaussian to

approximate the likelihood. But the generation of the Gaussian shortlist is totally different from the static shortlist generation in VQ-based Gaussian selection. DGS scheme uses a dynamic data-driven method to generate the shortlist for each observation-state pair. Unlike the two-pass search in VQ-based Gaussian selection, there is no pre-decided shortlist in DGS and no mapping of the observation vector to a certain centroid in the acoustic space before likelihood computation. The Gaussian shortlist is generated dynamically during the computational procedure of likelihood, according to a heuristic knowledge about the distance between each Gaussian and the best one to date. It is thus a single-pass search, within which not only the Gaussian shortlist is decided but likelihood is computed as well. The algorithm of this DGS scheme is described below.

Algorithm: Dynamic Gaussian Selection

INPUT: $\boldsymbol{x}_n$, an N-dimensional observation vector ; $\left\{ \mathcal{N}\left(Z_k , \boldsymbol{\mu}_k , \boldsymbol{\Sigma}_k \right), k = 1, 2, \cdots, M \right\}$, a GMM with M mixture components ; Q_{thresh} , a threshold number of loops on right-hand side of (2) ;

OUTPUT: D_{approx} , the approximation of log likelihood of the GMM .

PROCEDURE
BEGIN
(1) Compute D_{BMP}, the log likelihood of the BMP Gaussian component;
(2) $D_{max} =: D_{BMP}$;
(3) $D_{approx} =: D_{max}$;
(4) **WHILE** (the algorithm has not traversed all Gaussians) **DO**
BEGIN
(4.1) Perform PDE on an untouched Gaussian $\mathcal{N}\left(Z_k , \boldsymbol{\mu}_k , \boldsymbol{\Sigma}_k \right)$;
(4.2) **IF** (after Q_{thresh} loops the intermediate value of D_k is not less than D_{max})
(4.2.1) Complete the loops to derive D_k of the Gaussian ;
(4.2.2) $D_{approx} =:$ ***logadd*** [$D_{approx} + D_k$] ;
(4.2.3) **IF** ($D_k > D_{max}$)
(4.2.3.1) $D_{max} =: D_k$;
ENDIF
ENDIF
END
(5) **RETURN** D_{approx} ;
END

The basic idea of this algorithm is to use the number of loops at which the recursion on the right-hand side of (2) stops as a clue to decide whether the Gaussian should be included in the shortlist. In this algorithm, the BMP Gaussian is computed first, and then each Gaussian is evaluated using the standard PDE algorithm. For a Gaussian, the recursion of the right-hand side of (2) will stop at the jth element. If j is a number of small value, i.e., the summation loops stop at an early element, then the log likelihood of this Gaussian component could be far lower than BMP Gaussian since the value of (2) decreases monotonically with the progress of each loop. Therefore, the smaller the value of j, the greater the distance between this Gaussian component and the BMP Gaussian could be expected. This means that Gaussian components with a small value of j contribute little to the likelihood of the state and thus can be omitted in the likelihood computation. Otherwise, if j is a large number, the summation loop of (2) stops at a later element. This indicates that the log likelihood of this Gaussian component is close to BMP one, because the elements of the Gaussian component are reordered in such a way that elements with higher contribution to the distortion measure in (2) are computed first, followed by the elements contributing less. This Gaussian component contributes significantly to the state likelihood and thus it should be included in the shortlist. In the algorithm, a threshold number Q_{thresh} is given to decide whether a Gaussian component should join the shortlist. If j is greater than Q_{thresh} , the Gaussian component is selected to be included in the shortlist and all the loops of (2) are completed in order to include its full contribution in the likelihood. All the selected Gaussian components constitute the shortlist which is decided dynamically in the procedure of likelihood computation itself. PDE technique is used here to reduce the computational complexity. So, DGS scheme is an extended PDE technique in terms of that a Gaussian shortlist is decided based on PDE framework. In comparison with VQ-based Gaussian selection method, DGS scheme is memory saving because no shortlist should be pre-decided and kept in memory.

5. EXPERIMENTS AND RESULTS

Experiments on continuous speech recognition tasks have been carried out to evaluate and compare the performance of DGS scheme with those of PDE and its variants. The toolkit HTK 3.4 is used as the baseline system. The likelihood computation module in HTK has been modified to implement PDE and DGS schemes. Two accent-variant large vocabulary continuous speech corpora of English, TIMIT and HIWIRE [6], are used to perform the recognition.

The CMU phoneme set is adopted and 40 continuous HMMs for monophones are used as the acoustic models, including an HMM for silence. All HMMs have 3-state, left-to-right topology with the same number of Gaussian mixtures ranging from 16 to 128. The speech data is coded into 12 MFCCs, along with normalized log-energy and their first and second time derivatives, resulting in 39-dimensional feature vectors. To complement PDE with FER, the 39 elements of the feature vector are shuffled according to their contributions to the whole likelihood. The reordering is learned off-line from all the SA sentences (the dialect sentences) in the TIMIT test set.

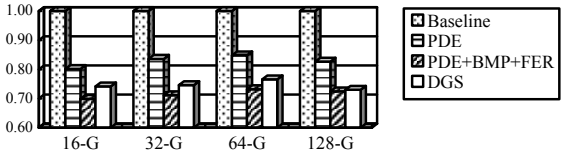

Figure 1 Normalized Recognition Time for TIMIT

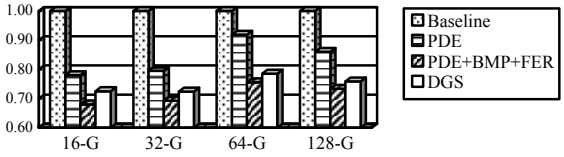

Figure 2 Normalized Recognition Time for HIWIRE

PDE and DGS schemes are assessed by calculating the total time for recognition, as well as the recognition accuracy. Instead of word or sentence accuracy, phone level accuracy is calculated in order to clearly demonstrate the differences between performances of different schemes. In the implementation of DGS scheme, the threshold number Q_{thresh} is set to be 35. All experiments are performed on a 3.4GHz Intel Pentium 4 machine with 2GB RAM. Figure 1 and Table 1 show the results of the experiment on TIMIT. Different HMMs with different number of Gaussians are trained and tested. The results for HIWIRE are shown in Figure 2 and Table 2. (With the acoustic model trained on TIMIT, the phone recognition accuracy on HIWIRE foreign accent corpus is quite low.) The recognition times are normalized by that of the corresponding baseline. All the results are an average of 10 runs on each configuration in order to reduce interference from outside processes.

The experimental results indicate that DGS scheme achieves a significant saving (>21%) of phone recognition time with a smaller degradation of accuracy than PDE. Confidence interval measuring of the results indicates that there is no significant difference between baseline and DGS in the sense of accuracy while the difference between PDE and baseline is apparent. It is noticeable that DGS algorithm takes a little more time to compute the likelihood than the scheme of PDE incorporated with both BMP and FER. The extra time cost comes from the completion of the summation loop of (2) for the selected Gaussians in the dynamic shortlists. Experimental results show that high recognition accuracy can be achieved with the average length of the dynamic shortlists less than 3. Therefore the extra time cost of DGS algorithm is quite limited.

6. CONCLUSION

A fast likelihood computation technique, called Dynamic Gaussian Selection (DGS) is proposed based on the concept of Gaussian selection. This approach is a one-pass search technique which generates a dynamic shortlist of Gaussians for each state during the procedure of likelihood computation. DGS algorithm is an extension of PDE technique. It uses the number of summation loops in the likelihood computation to dynamically decide a small set of "sub-optimal" Gaussians which are numerically close to the "best" Gaussian. Though the theoretic gain of DGS remains to be analyzed, experiments show that limiting the likelihood computation on the selected shortlist can significantly speed up the likelihood computation while introducing almost no additional recognition error. DGS does not require extra memory for the storage of Gaussian shortlists, making it particularly suited for applications on embedded platforms. Furthermore, we can integrate DGS with other optimization techniques, such as the context-independent HMM-based two-pass search used in Julius system [7], so as to improve the speed of likelihood computation as much as possible.

Table 1 Recognition Accuracy for TIMIT

Scheme	Phone Accuracy (%)			
	16-G	32-G	64-G	128-G
Baseline	56.7	58.7	59.9	60.2
PDE	56.3	58.3	59.4	59.7
PDE+BMP+FER	56.3	58.3	59.4	59.7
DGS	56.6	58.7	59.8	60.1

Table 2 Recognition Accuracy for HIWIRE

Scheme	Phone Accuracy (%)			
	16-G	32-G	64-G	128-G
Baseline	36.6	38.7	41.0	42.2
PDE	36.2	38.4	40.5	41.8
PDE+BMP+FER	36.2	38.4	40.5	41.8
DGS	36.6	38.8	41.0	42.2

7. REFERENCES

[1] M. J. F. Gales, K. M. Knill, and S. J. Young, "State-Based Gaussian Selection in Large Vocabulary Continuous Speech Recognition Using HMM's," *IEEE Trans. on Speech and Audio Processing*, Vol. 7, No. 2, pp. 152-161, March 1999.

[2] E. Bocchieri, "Vector Quantization for the Efficient Computation of Continuous Density Likelihood," *Proc. of ICASSP'93*, Vol. 2, pp. 692-695, April 1993.

[3] B. L. Pellom, R. Sarikaya, and J. H. L. Hansen, "Fast Likelihood Computation Techniques in Nearest-Neighbor Based Search for Continuous Speech Recognition," *IEEE Signal Processing Letters*, Vol. 8, No. 8, pp. 221-224, August 2001.

[4] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory Algorithm and System Development*, Prentice Hall PTR, New Jersy, April 2001.

[5] J. Fritsch, and I. Rogina, "The Bucket Box Intersection (BBI) Algorithm For Fast Approximation Evaluation of Diagonal Mixture Gaussians," *Proc. of ICASSP'96*, Vol. 2, pp. 837-840, May 1996.

[6] G. Bouselmi, D. Fohr, I. Illina, and J.-P. Haton, "Multilingual Non-Native Speech Recognition using Phonetic Confusion-Based Acoustic Model Modification and Graphemic Constraints," *Proc. of INTERSPEECH'07*, pp. 1449-1452, August 2007.

[7] A. Lee, T. Kawahara, and K. Shikano, "Gaussian Mixture Selection Using Context-Independent HMM," *Proc. of ICASSP'01*, Vol. 1, pp. 69-72, May 2001.