SUB-PHONETIC POLYNOMIAL SEGMENT MODEL FOR LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

Siu-Kei AU YEUNG, Chak-Fai LI and Man-Hung SIU

Department of Electrical and Electronic Engineering
Hong Kong University of Science and Technology, Hong Kong
jeffay@ust.hk, onefai@ust.hk, eemsiu@ust.hk

ABSTRACT

Polynomial Segment Model (PSM) has opened up an alternative research direction for acoustic modeling. In our previous papers [1, 2], we proposed efficient incremental likelihood evaluation and EM training algorithms for PSM, that significantly improve the speed of PSM training and recognition. In this paper, we shift our focus to use PSM on large vocabulary recognition. Recognition via N-best re-scoring shows that PSM models out-performed HMM on the 5k closed vocabulary Wall Street Journal Nov 92 testset. Our best PSM model achieved 7.15% WER compare with 7.81% using 16 mixture HMM model. Specifically, we used sub-phonetic PSM that represents a phoneme as multiple independent segmental units that allows for more effective model sharing. Also, we derived and compared different top-down mixture growing approaches that are orders of magnitude more efficient than previously proposed bottom-up agglomerative clustering techniques. Experimental results show that the top-down clustering performs better than the bottom-up approaches.

1. INTRODUCTION

In recent years, researchers have examined alternatives to the HMM for representing speech acoustics. One such alternative is the segment models [3] that is a generalization of HMM. The segment model explicitly represents the speech dynamics and temporal correlations between frames. Polynomial Segment Model (PSM) [4] is one type of the segment models that represents the speech acoustics by a polynomial function.

While PSM has been applied to many small vocabulary tasks such as phoneme recognition or classification [1] [2] [5] and has shown improved performance, limited work has been reported of using PSM for large vocabulary continuous speech recognition (LVCSR) [6]. There are several difficulties in applying PSM for LVCSR. First, because of the segmental assumption, training and recognition require searching over all possible segment boundaries. In addition, the joint modeling of all observations within a segment typically requires complete re-computation of segment likelihood when the segment boundary is changed. This increased computation makes it hard to handle LVCSR which typically involves more modeling units and uses larger amount of training data. Second, to reduce the number of segmental units, one complete phonetic unit is usually represented by a single segment making it harder to share across context dependent models. While using multiple segments to represent a phonetic unit is possible, it increases the number of segment boundaries in training and recognition. Third, in order for PSM to work in LVCSR, it is necessary to create mixture models [6] and triphone models which have not been fully explored.

In our previous papers [1, 2], we proposed fast likelihood computation algorithms that significantly improved the PSM recognition and training efficiency. In addition, we introduced the dynamic multi-region PSM with different levels of sharing between the regions, which ranges from complete independent regions to shared mean trajectory and variance. One important advantage of the dynamic multi-region segment model is the data-driven alignment between observations and the region boundaries. We have shown that the new dynamic multi-region model out-performs HMM and traditional PSM in both phone classification and phone recognition task on the TIMIT corpus.

Mixture PSM was first introduced in [5] using bottom-up clustering approach to initialize the mixture components and this approach was also used in our previous papers [1, 2]. However, this approach is too computationally intensive for LVCSR and it is not clear whether the resulting clusters are suitable for mixtures in recognition.

In this paper, we extend our previous work to the large vocabulary tasks including the use of fast likelihood computation in search and training and duration modeling incorporation. To add flexibility to the model and allow us to draw more resources from the HMM framework, we use a sub-phonetic PSM that is a special case of the dynamic multi-region segment. Furthermore, we introduce a number of modified K-means approaches and explore different initialization strategies and distance measures for mixture model estimation. These clustering approaches are compared with the bottom-up clustering approach used in our previous paper.

The organization of this paper is as follows. In section 2, the basic formulation of PSM is presented. In section 3, we outline the experimental setup using the WSJ0 (standard SI-84 WSJ trainset and Nov'92 5000 words evaluation set) and report the HMM baseline performances. In section 4, the proposed method on clustering triphone model is discussed. In section 5, the performance and the processing time of the proposed methods are presented and the paper is concluded in section 6.

2. POLYNOMIAL SEGMENT MODEL

PSM definition and estimation were first derived in [4]. PSM is defined as,

$$C = Z_N B + E, \tag{1}$$

where C is a NxD feature matrix for N frames of D dimensional feature vector. Z_N is a N x (R+1) design matrix for a R^{th} order trajectory model that maps the segments of different durations to a range of 0 to 1 and B is a (R+1) x D parameter model matrix.

The maximum likelihood estimation of the trajectory parameter matrix B for a speech segment C with N frames is given by,

$$B = [Z'_N Z_N]^{-1} Z'_N C \tag{2}$$

and the corresponding residue error covariance is given by

$$\Sigma = \frac{(C - Z_N B)'(C - Z_N B)}{N} \tag{3}$$

The triplet {B, Σ, N } can be viewed as the sufficient statistics for C. For a set of segments C_1,...,C_K of model m, the maximum likelihood estimation for the PSM parameter matrix $\hat{B}_m$ and the residue covariance $\hat{\Sigma}_m$ are given by

$$\hat{B}_m = \left[\sum_{k=1}^{K} Z'_{N_k} Z_{N_k}\right]^{-1} \left[\sum_{k=1}^{K} Z'_{N_k} C_k\right] \tag{4}$$

and

$$\hat{\Sigma}_m = \frac{\sum_{k=1}^{K} (C_k - Z_{N_k}\hat{B_m})'(C_k - Z_{N_k}\hat{B}_m)}{\sum_{k=1}^{K} N_k} \tag{5}$$

The likelihood of segment C_j against model m (with mean $\hat{B}_m$ and variance $\hat{\Sigma}_m$) can be evaluated using the segment's sufficient statistics, { B_j, Σ_j, N_j } and is given by,

$$\begin{aligned} L(\hat{B}_j, \hat{\Sigma}_j | B_m, \Sigma_m) = & \\ & -\frac{N_j}{2}[Dlog(2\pi) + log|\hat{\Sigma}_m|] - \frac{N_j}{2} tr[\Sigma_m^{-1}\hat{\Sigma}_j] \\ & -\frac{1}{2} tr[Z_{N_j}(\hat{B}_j - B_m)\Sigma_m^{-1}(\hat{B}_j - B_m)' Z'_{N_j}]. \end{aligned} \tag{6}$$

3. EXPERIMENTAL SETUP AND BASELINES

In this paper, LVCSR experiments were performed on the ARPA Wall Street Journal (WSJ) 5k word task [7]. The models were trained using the standard SI-84 train set with 7138 utterances and tested on the Nov'92 5000 word evaluation set with 330 utterances. This training and testing setup is also consistent with the one used in the Aurora 4 corpus [8]. All experiments were performed using Mel-frequency cepstral coefficients(MFCC) with 39 dimension after applying cepstral mean subtraction(CMS). Three states left-to-right cross-word HMM triphone models were trained using the EM-algorithm. The HMM baseline results are generated using HTK (version 3.2). Triphones states were tied using the decision-tree clustering technique resulting in a total of 3185 tied states. The HMM training and decoding procedure and settings were similar to [9], with bigram language model was used. This makes our experiment comparable with other published WSJ results [10, 11].

For PSM, 3 independent sub-phonetic segments were used to model each phoneme which can be viewed as a special case of the dynamic multi-region PSM [2]. Because of using 3-segment per phoneme, only first order PSM (linear) was used instead of the more commonly used second order (quadratic) model. To allow easy comparison between HMM and PSM, both models used the same MFCC features. In addition, the HMM triphone state tying tree was also applied to tied PSM sub-phonetic segments across different triphones. While this is not optimal, this simplifies our implementation. The PSM segment alignment was initialized by using the phoneme alignment generated by using a single mixture HMM model. Similar to the HMM, the pruning threshold, the grammar weight and the insertion penalty were tuned empirically. For simplicity, PSM models were trained using Viterbi training instead of E-M training.

While it is possible to perform a full PSM search, our current PSM implementation does not support cross-word triphone which gives better performance on this task. PSM recognition was performed using N-Best rescoring with N-best generated from HMM models. However, different from other rescoring work [6], the HMM alignment is not used. Instead, a full search for optimal segment boundaries was performed using the fast PSM computation [2]. Results tabulated in Table 2 used a N-Best size of 10 with a Gaussian duration model.

Number of mixtures	1	2	4	8
WER (%)	14.09	11.92	9.64	8.80

Table 1. *Baseline result on WSJ using HMM with different number of mixtures*

Table 1 summarizes the baseline performance on the WSJ0 tasks using HMM. We further improve the baseline by performing the endpoint process on the training data and increase the mixture to 16, our best HMM baseline achieve 7.81% WER which is comparable with [10, 11].

4. CLUSTERING FOR MIXTURE DENSITY MODEL

Mixture models are often used to capture speech variations. E-M re-estimation formulation for mixture PSM was derived in [4] and generalized in [12]. For the E-M training to be efficient, good mixture initializations are needed. In HMM models, binary splitting and K-mean clustering are often used to initialize the mixtures. In PSM, however, the most commonly used approach is the bottom-up clustering based on the pairwise, likelihood-ratio distance between segments pairs [5], which was also used in our previous work [2]. However, the computation can become very intensive for LVCSR. Therefore, we investigate several top-down clustering techniques based on the K-means clustering algorithm similar to what is applied in HMM.

4.1. Bottom-up Clustering

As described in [5], the distance between 2 segments can be measured using the likelihood ratio of whether the two segments are generated by the same model or being generated by two distinct models. Given two segments X, Y, and their corresponding sufficient statistics { B_X, Σ_X, N_X } and { B_Y, Σ_Y, N_Y }, the **likelihood ratio distance** [5] is,

$$d_{TRAJ}(X, Y) = \frac{N_X + N_Y}{2} \log |I + W^{-1} S| \tag{7}$$

where W = $\frac{N_X \Sigma_X + N_Y \Sigma_Y}{N_X + N_Y}$ is the weighted average of the individual variances, B is the mean trajectory and S is the covariance of this joint model of X and Y. S can be expressed as:

$$S = \frac{(Z_X B_X - Z_X B)'(Z_X B_X - Z_X B)}{N_X + N_Y}. \tag{8}$$

Based on all the pairwise distances, one can then use bottom-up agglomerative clustering to construct a dendogram (clustering tree). This dendogram can then be cut to obtain the desired number

of clusters. Once all the data is partitioned into different clusters, the segments within a cluster are considered to have a common trajectory and are combined to form the initial mixture models. Because the pairwise distances between all segments are computed, the number of distance computations is of order $O(N^2)$ where N is the number of segments in a cluster. To reduce computation, the covariances Σ_X and Σ_Y can be assumed to be diagonal.

There are two issues with this mixture initialization in LVCSR. First, there are a large number of models (3000 triphone-states) that require separate clustering. Many of these triphone states contain a large number of segments in which $O(N^2)$ distances are too intensive to compute. While one can use a subset of segments for clustering, it may affect the quality of the resulting clusters. The second issue is the difficulty in cutting the dendogram into the right clusters for mixtures. While the clusters produced do group together "similar" segments; sometimes, some clusters are similar to each other resulting in some mixture components with very small weights. Unlike the splitting algorithm used in HMM in which a mixture with very small weight can be removed and replaced by splitting the component with the largest weights, the bottom-up approach does not provide an obvious mechanism for increasing the number of mixtures after the dendogram is cut.

4.2. K-means Clustering

An alternative to the agglomerative clustering is the K-mean clustering which involves two steps: 1) the assignment of data to the nearest centroid, and 2) the estimation of a centroid given a set of data. If the data is in Eucludian space, it is well-known that using Eucludian distance in step (1) and data average in step (2) can minimize the total square error.

For PSM, all the triphone instances can be represented by their sufficient statistics $\{B_j, \Sigma_j, N_j\}$. Since B_j determines the shape of the trajectory, we can simplify the problem by ignoring Σ by assuming them to be the identity matrix. That is, we focus on clustering segments that have similar B_j.

The square error between an N-frame PSM segment with mean B_j and another segment mean B_m is given by:

$$d(j,m) \quad = \quad tr[Z_{N_j}(B_j - B_m)(B_j - B_m)' Z'_{N_j}]. \quad (9)$$

As shown in [5], for a given set of segments means, $\{B_k\}, 1 \leq k \leq K$, the maximum likelihood (or equivalently when using identity matrix as covariance), the minimum square error centroid is given by

$$\hat{B}_m \quad = \quad \left[\sum_{k=1}^{K} Z_k^{'} Z_k\right]^{-1} \left[\sum_{k=1}^{K} Z_k^{'} Z_k B_k\right], \quad (10)$$

By using Equation 9 for distance computation and Equation 10 to re-estimate the centroid, we can form the minimum square error clustering. Because the total square error would always decrease in both steps, the algorithm converges.

However, the computation of Equation 9 can still be intensive because of the per-segment re-scaling. If we consider the two trajectories as continuous time functions, then, we can formulate the square error function between the two polynomials which is also a polynomial. Instead of sampling the polynomials by Z_k to compute the total point-wise distance, it can be approximated by the integral of this square error function which is easy to compute. That is,

$$\hat{d}(k,m) \quad = \quad N_k \sum_{d}^{D} \int_0^1 \left(B_{d,k}(t) - B_{d,m}(t)\right)^2 dt, \quad (11)$$

where D is the dimension of the feature vector, $B_{d,k}(t)$ and $B_{m,k}(t)$ are the polynomials with the d-th rows of the B_k and B_m matrices respectively as coefficients. We called this the **integral distance**. This approximation is more accurate for longer segments.

Alternatively, one can consider B_k's as data points in Eucludian space. Suppose we denote $v(B_k)$ as the vectorized form of B_k by concatenating its columns. That is, if B_k contains l columns, $B_k = [B_{k,1}, \ldots, B_{k,l}]$, then,

$$v(B_k) = \begin{bmatrix} B_{k,1} \\ \vdots \\ B_{k,l} \end{bmatrix}.$$

We also denote the conversion of the vectorized B_k back to the matrix form as iv. That is $iv(v(B_k)) = B_k$. Then, one can use the simple K-means algorithm to cluster the data. We call this the **vectorized distance**.

4.3. Top-down Clustering

However, K-means clustering would require initial estimate of the centroids. The idea of top-down clustering is to combine all the data into a single cluster and then progressively increase the number of clusters as needed.

To apply the top-down clustering on PSM, we would need to design a way to split a centroid into two. The vectorized parameters provide a handy solution. For each cluster, say, cluster m, with K_m segments, in addition to estimating the centroid, we can also estimate the variance of B_m, denoted as $\tilde{\Sigma}_m$ by

$$\tilde{\Sigma}_m = diag[\frac{\sum_{i=1}^{K_m}((v(B_i) - v(\hat{B}_m))'(v(B_i) - v(\hat{B}_m)))}{K_m}] \quad (12)$$

Then, using the iv notation defined above, a centroid B_m can be split into $B_{m,+}, B_{m,-}$, the new centroids. That is,

$$B_{m,+} \quad = \quad iv(v(B_m) + \epsilon\tilde{\Sigma}_m) \quad (13)$$

$$B_{m,-} \quad = \quad iv(v(B_m) - \epsilon\tilde{\Sigma}_m), \quad (14)$$

where ϵ is a small constant.

5. EXPERIMENTS

In our first experiment, we evaluated the PSM performance using single mixture which gave a WER of 13.1% which is 7% better than HMM with 1 mixture. We then proceed to compare the HMM and PSM system with mixtures using different mixture initialization schemes. We compared five different clustering approaches:

1. **Method 1** Likelihood ratio based distance bottom-up clustering with full covariance.
2. **Method 2** Likelihood ratio based distance bottom-up clustering with diagonal covariance.
3. **Method 3** Top-down clustering using integral distance.
4. **Method 4** Top-down clustering using vectorized distance.
5. **Method 5** K-means clustering with 5 random initialization with the vectorized distance.

For method 1 and 2, to reduce computation, only 500 segments per triphone state are used in clustering.

Table 2 summarizes the recognition performance of the five mixture initialization methods in terms of WER in column 2, the relative improvement over the HMM model of the same number

model	WER	Relative imp.	Clust. time
HMM (2mix)	11.92%	-	-
PSM (method 1)	10.46%	12.2%	6.5 weeks
PSM (method 2)	10.23%	14.2%	3 weeks
PSM (method 3)	10.01%	19%	2 hrs
PSM (method 4)	10.18%	14.6%	1 hrs
PSM (method 5)	10.07%	15.5%	5 hrs

Table 2. *Result on WSJ0 N-Best Rescoring with mixture model.*

of mixtures in column 3 and the processing time in column 4. More than 12% relative improvement is achieved using either approaches. Among the five methods, method 3, which coupled binary splitting with a K-means algorithm using the integral distance performs the best with 19% improvement. The three modified K-means algorithms (method 3-5) out-perform the bottom-up methods probably because all the segments were used in the initialization step. Meanwhile, the cheaper diagonal covariance method (method 2) in bottom-up clustering out-performs the full covariance (method 1). One possible reason is that our final models use diagonal covariances.

The PSM model is further increased to 8 mixture and a 50-Best List generated from 8mix HMM is used. With the use of endpointing in training and gamma duration model, we achieve a 7.15% WER which is 8.5% better than our best HMM performance with 16 mixture with 7.81% WER.

In terms of computation, the processing time required for clustering the two mixture model is shown in the 4-th column of Table 2. The processing time were the elapsed time computed using a P-4 2.4GHz machine. This processing time does not include the time to generate sufficient statistics for each segment which can take several minutes to hours depending on whether covariance is used. For method 1 and 2, only a maximum of 500 instances triphones were used in clustering while for the K-means methods (method 3-5), all data were used in clustering. It is clear that the proposed K-Mean algorithms required much less computation compared with the likelihood ratio distance with bottom-up clustering. For Algorithm 3, the processing time is directly proportional to the number of random initial points.

6. SUMMARY

In this paper, we report our experience using PSM for LVCSR tasks. Significantly improvement is achieved. Our best PSM LVCSR model achieved a WER of 7.15% compared to 7.8% when using HMM with similar complexity.

To perform recognition in LVCSR, we first proposed the use of sub-phonetic segments. This allows us to use well established model tying approach commonly used in HMM. We also explored several modified K-means algorithms for mixture initialization. These algorithms are compared with the bottom-up agglomerative clustering used in our previous work. We have showed from our experiments that the proposed algorithms out-perform the traditional clustering techniques both in terms of recognition accuracy and processing speed. For the two-mixture model, PSM with the integral distance K-means clustering achieved 19% relative improvement over HMM. For the 4-mixture case, the relative improvement is 8%. The processing time of the proposed algorithms are simply 1/100 of the agglomerative clustering.

Comparing the vectorized distance with the integral distance, it should be noted that they will be the same if the orthogonal polynomial basis is used [13] because then, the integral of the polynomial square error function will be reduced to the sum of squares of differences in coefficients.

Currently, we are implementing the orthogonal transformation based clustering. We also plan to apply EM training instead of Viterbi training.

7. ACKNOWLEDGEMENT

This work is partially supported by HK Government Research Grant Council CERG grant #HKUST/6049/00E.

8. REFERENCES

[1] C.F. Li and M. Siu, "An efficient incremental likelihood evaluation for polynomial trajectory model using with application to model training and recognition," in *Proceedings of ICASSP 2003*, 2003, pp. 756–759.

[2] C.F. Li and M. Siu, "Training for polynomial segment model using the expectation maximization algorithm," in *Proceedings of ICASSP 2004*, 2004, pp. 841–844.

[3] M. Ostendorf, V. V. Digalakis, and O. A. Kimball, "From hmm's to segment models: A unified view of stochastic modeling for speech recognition," *IEEE Trans of Speech and Audio Processing*, vol. 4, pp. 360–387, 1996.

[4] H. Gish and K. Ng, "A segmental speech model with applications to word spotting," in *Proceedings of ICASSP 93*, 1993, pp. 447–450.

[5] H. Gish and K. Ng, "Parametric trajectory models for speech recognition," in *Proceeding of ICSLP 96*, 1996, vol. 1, pp. 466–469.

[6] M. Siu, R. Iyer, H. Gish, and C. Quillen, "Parametric trajectory mixtures for lvcsr," in *Proc. of ICSLP*, 1998.

[7] D. Paul and J. Baker, "The design of wall street journal-based csr corpus," in *Proceedings of ICSLP 1992*, 1992, pp. 899–902.

[8] N. Parihar and J. Picone, "Dsr front end lvcsr evaluation au/384/02," 2002.

[9] Siu-Kei Au Yeung and Man-Hung SIU, "Improved performance of aurora 4 using htk and unsupervised mllr," in *Proceedings of ICSLP 2004*, 2004, pp. 161–164.

[10] Francis Kubala, Anastasios Anastasakos, John Makhoul, Long Nguyen, Richard Schwartz, and George Zavaliagkos, "Comparative experiments on large vocabulary speech recognition," in *Proceedings of ICASSP 1994*, 1994, pp. 561–564.

[11] Yanghong Yang, Xintian Wu, Johan Schalkwyk, and Ron Cole, "Development of cslu lvcsr: The 1997 darpa hub4 evaluation system," in *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[12] Toshiaki Fukada, Yoshinori Sagisaka, and Kuldip K. Paliwal, "Model parameter estimation fro mixture density polynomial segment models," in *Proceedings of ICASSP 1997*, 1997, pp. 1403–1406.

[13] Pui-Fung WONG and Man-Hung SIU, "Decision tree based tone modeling for chinese speech recognition," in *Proceedings of ICASSP 2004*, 2004, pp. 905–908.