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, making it possible to train and recognize using PSM alone. In this paper, we shift our focus to make it feasible to use PSM on large vocabulary recognition. First, we used sub-phonetic PSM that represents a phoneme as multiple independent segmental units. Second, we derived and compared different top-down mixture growing approaches that are orders of magnitude more efficient than previously proposed agglomerative clustering techniques. Experimental results show that the top-down clustering performs better than the bottom-up approach. Recognition via N-best re-scoring shows that PSM models outperformed HMM by 7% to 19% on the 5k closed vocabulary Wall Street Journal Nov '92 testset. Our best PSM model achieve 7.15% WER compare with 7.81% use 16 mixture HMM model.

In our previous papers [1, 2], we proposed fast likelihood computation algorithms that significantly improve the PSM recognition and training efficiency. In addition, we introduced the dynamic multi-region PSM with different levels of sharing between the regions, which range 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 paper [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 formations of PSM is presented. In section 3, we outline the experimental setup using the WSJ0 (standard SI-84 WSJ train-set 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,

\[ C = Z_N B + E, \]

(1)

where C is a NxD 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.
2.1. PSM Parameter Estimation

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

$$B = [Z_N^T Z_N]^{-1} Z_N^T C$$  \hspace{1cm} (2)

and the corresponding residue error covariance is given by

$$\Sigma = \frac{(C - Z_N B)(C - Z_N B)^T}{N}$$ \hspace{1cm} (3)

The triplet \{B, $\Sigma$, N\} can be viewed as the sufficient statistics for C. For 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

$$\hat{B}_m = [\sum_{k=1}^K Z_N^T C_k]^{-1} [\sum_{k=1}^K Z_N^T B_m]$$ \hspace{1cm} (4)

and

$$\hat{\Sigma}_m = \frac{\sum_{k=1}^K (C_k - Z_N \hat{B}_m)^T (C_k - Z_N \hat{B}_m)}{\sum_{k=1}^K N_k}$$ \hspace{1cm} (5)

2.2. Log Likelihood Evaluation

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, \{ $B_j$, $\Sigma_j$, N$_j$ \} and is given by,

$$L(B_j, \Sigma_j | B_m, \Sigma_m) =$$ \hspace{1cm} (6)

$$= -\frac{N}{2} [D \log (2\pi) + \log|\Sigma_m|] - \frac{N_j}{2} [D \log (2\pi) + \log|\Sigma_j|] - \frac{1}{2} tr[Z_N (B_j - B_m)^T \Sigma_m^{-1} (B_j - B_m) + Z_N^T \Sigma_j Z_N]$$

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 andsettings were similar to [9]. Only bigram language model was used and language modeling weights, pruning thresholds and insertion penalties were tuned empirically.

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. An N-Best size of 10 was used in all the experiments and were generated from HMM model. 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]. A Gaussian duration model was applied during rescoring and the duration modeling weight was also estimated empirically.

<table>
<thead>
<tr>
<th>Number of mixtures</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>WER (%)</td>
<td>14.09</td>
<td>11.92</td>
<td>9.64</td>
<td>8.80</td>
</tr>
</tbody>
</table>

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 with 1 to 4 mixtures. 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].

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 [11]. 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 two segments [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 \{ $B_X$, $\Sigma_X$, N$_X$ \} and \{ $B_Y$, $\Sigma_Y$, N$_Y$ \}, the likelihood ratio distance [5] is,

$$d_{LRA}(X, Y) = \frac{N_X + N_Y}{2} \log \left| I + W^{-1} S \right|$$ \hspace{1cm} (7)

where $W = \frac{N_X B_X + N_Y B_Y}{N_X + N_Y}$ 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:

$$S = \frac{(Z_X B_X - Z_Y B)^T (Z_X B_X - Z_Y B)}{N_X + N_Y}.$$
Based on all the pairwise distances, one can then use bottom-up agglomerative clustering to construct a dendrogram (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 $\Sigma_X$ and $\Sigma_Y$ 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 dendrogram into the right clusters for mixtures. While the clusters produced do group together “similar” segments; sometimes, some clusters are similar right clusters for mixtures. While the clusters produced do group together “similar” segments; sometimes, some clusters are similar.

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 are in Euclidean space, it is well-known that using Euclidean 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 $\Sigma$ by assuming them to be 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 mean $B_m$ is given by:

$$d(j,m) = tr[Z_{nj}(B_j - B_m)(B_j - B_m)^tZ_{nj}].$$

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

$$\hat{B}_m = \left(\sum_{k=1}^{K} Z_{kj}^t Z_{kj}\right)^{-1} \left(\sum_{k=1}^{K} Z_{kj}^t B_k\right).$$

By using Equation 8 for distance computation and Equation 9 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 8 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,

$$\hat{d}(k,m) = N_k \sum_{d}^{D} \int_{0}^{1} (B_{d,k}(t) - B_{d,m}(t))^2 dt,$$

where $D$ is the dimension of the feature vector, $B_{d,k}(t)$ and $B_{d,m}(t)$ are 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, we can consider $B_k$’s as data points in Euclidian 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 the $B_m$, denoted as $\Sigma^2_m$ by

$$\Sigma^2_m = diag(\sum_{i=1}^{K_m} (iv(B_{ki}) - iv(\hat{B}_m))/iv(v(B_k) - iv(\hat{B}_m)))$$

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

$$B_{m,+} = iv(v(B_m)) + \epsilon \Sigma^2_m$$

$$B_{m,-} = iv(v(B_m)) - \epsilon \Sigma^2_m,$$

where $\epsilon$ 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.
Table 2. Result on WSJ0 N-Best Rescoring with mixture model.

<table>
<thead>
<tr>
<th>Model</th>
<th>WER</th>
<th>Relative Imp.</th>
<th>Clust. Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM (2mix)</td>
<td>11.92%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PSM (method 1)</td>
<td>10.46%</td>
<td>12.2%</td>
<td>6.5 weeks</td>
</tr>
<tr>
<td>PSM (method 2)</td>
<td>10.23%</td>
<td>14.2%</td>
<td>3 weeks</td>
</tr>
<tr>
<td>PSM (method 3)</td>
<td>10.01%</td>
<td>19%</td>
<td>2 hrs</td>
</tr>
<tr>
<td>PSM (method 4)</td>
<td>10.18%</td>
<td>14.6%</td>
<td>1 hrs</td>
</tr>
<tr>
<td>PSM (method 5)</td>
<td>10.07%</td>
<td>15.5%</td>
<td>5 hrs</td>
</tr>
</tbody>
</table>

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 5 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 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 is binary splitting coupled 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 is used. With the use of endpointing and gamma duration model, we achieve 7.15% WER which is 8.5% better than our best HMM performance with 16 mixture.

In terms of computation, the processing time required for clustering the two mixture model is shown in the 4-th column of Table 5. 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 triphone instances 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 when using PSM compared with HMM with similar complexity.

To perform recognition in LVCSR, we also explored several modified K-means algorithm for mixture initialization. These algorithms are compared with the bottom-up agglomerative clustering used in our previous work. We have showed from the experiment that the performance of 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 [12] 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. Furthermore, instead of using the tying tree derived for HMM, we are developing a tying scheme for PSM.

7. ACKNOWLEDGEMENT

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

8. REFERENCES