TRAINING FOR POLYNOMIAL SEGMENT MODEL USING THE EXPECTATION MAXIMIZATION ALGORITHM

Chak-Fai Li and Man-Hung Siu

EEE Dept., Hong Kong University of Science and Technology, Clearwater Bay, Hong Kong
onefai@ust.hk eemsiu@ee.ust.hk

ABSTRACT

One of the difficulties in using polynomial segment model (PSM) to capture the temporal correlations within a phonetic segment is the lack of an efficient training algorithm comparable with the Baum-Welch algorithm in HMM. In our previous paper, we introduced a recursive likelihood computation algorithm for PSM recognition and can perform Viterbi-style training. In this paper, we extend the recurrsive likelihood computation into a fast forward-backward PSM training algorithm that maximizes PSM likelihood. In addition, we introduce an improved PSM, dynamic multi-region PSM, that allows a data-driven alignment between observations and the segment trajectory. The dynamic multi-region PSM model outperforms HMM and traditional PSM in both phone classification and phone recognition tasks on the TIMIT corpus.

1. INTRODUCTION

One advantage of HMM is that both model training and recognition can be efficiently performed using dynamic programming based algorithms such as the Baum Welch and Viterbi algorithms. While the polynomial segment model (PSM) [1], in which HMM is a special case, that relaxes the conditional independence assumption of the HMM is of interest to different researchers [3],[6], its computational complexity limits its applications.

Computation complexity for PSM is much higher because of two reasons: 1) PSM models are defined in terms of segments. So, an additional search over all possible segment boundaries is needed. 2) Observation likelihoods within a segment are evaluated jointly because of their dependence on the segment boundary and duration. Because of this, extending a segment to include an extra observation requires the re-computation of all the observation likelihoods within the segment. This re-computation makes recognition and training computationally intensive.

In our previous paper [11], we proposed a fast likelihood computation algorithm that significantly improved the PSM-based recognition efficiency. The efficient likelihood computation can also be applied to model training. One major difference between recognition and training algorithms is the computation of the state posterior probability in the expectation step of the training which requires the consideration of all possible segment sequences. In this paper, we introduce an efficient way of computing the forward and backward probabilities, typically denoted as $\alpha_t(j), \beta_t(j)$, which makes use of the fast likelihood computation.

When one compares the HMM and the PSM with multiple variances [2] which can be considered to have multiple "states" with a time varying mean but a constant covariance within the state , one difference is the assignment of the observations to the "states". In HMM, observations are aligned to maximize the likelihood while in PSM with multiple covariances, they are uniformly assigned. In this paper, we propose an improved model that allows the assignment during recognition to be likelihood driven. We call it the dynamic multi-region PSM.

In the next section, we briefly review the PSM formulation including the maximum likelihood parameter re-estimation equations using the EM algorithm. In Section 3, we discuss the application of the fast likelihood evaluation to computate the forward and backward probabilities. In Section 4, we describe the proposed model improvements and the related training and recognition implementations. Experimental results are reported in Section 5 and we conclude the paper in Section 6.

2. BASIC FORMATIONS OF POLYNOMIAL SEGMENT MODEL

Polynomial Segment Model, first proposed in [1], is defined as,

$$C = Z_N B + E,$$

where C is a $N \times D$ matrix for N frames of D dimensional feature vector. B is a $(R+1) \times D$ parameter matrix of a R^{th} order trajectory model and E is the residual error that is the same size as the feature matrix C. Z_N is an $N \times (R+1)$ time normalization matrix, also called a design matrix, that maps segments of different durations within a range of between 0 and 1.

2.1. Parameter Estimation of a Single PSM

As described in [1], the maximum likelihood estimate of the trajectory parameter matrix B_k of a speech segment C_k with N_k frames is given by,

$$B_k = \left[Z'_{N_k} Z_{N_k}\right]^{-1} Z'_{N_k} C_k,$$

and the corresponding residue error covariance is given by,

$$\Sigma_k = \frac{E'_k E_k}{N_k} = \frac{(C_k - Z_{N_k} B_k)'(C_k - Z_{N_k} B_k)}{N_k}.$$

The triplet $\{B_k, \Sigma_k, N_k\}$ are viewed as the sufficient statistics for the segment C_k. Given a set of K segments $S = \{C_1, \ldots, C_K\}$ of model m, the maximum likelihood estimate of the PSM parameter matrix $\hat{B}_m$ and residue covariance $\hat{\Sigma}_m$ are given by

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

and

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

This work is partially funded by RGC CERG: project number HKUST6049/00E.

2.2. Log Likelihood Evaluation

The likelihood of segment C_j against model m can either be evaluated using its sufficient statistics, $\{B_j, \Sigma_j, N_j\}$ or be computed by accumulating the observation likelihoods one at a time, against the corresponding sampling point on the Polynomial Segment Model. This log likelihood, $L(C_j|m)$, can also be written as

$$L(C_j|m) = -\frac{N_j}{2}[D\log(2\pi) + \log|\hat{\Sigma}_m|] - \frac{1}{2}tr[(C_j - Z_{N_j}\hat{B}_m){\hat{\Sigma}_m}^{-1}(C_j - Z_{N_j}\hat{B}_m)']. \quad (1)$$

2.3. Maximum Likelihood Training of PSM Parameters

The basic idea of using the EM algorithm for training PSM is to find a set of model parameters so that the expected log likelihood is maximized over all possible segment alignments.

The log-likelihood formula given in Equation 1 can be extended from single segment to a sequence of segments. The sequment notation is changed from C_k to O_d^t, which represents the segment that end at time t with duration d, because segment boundaries are no longer pre-defined as in the previous sections. An observation sequence $O_T^T = [o_1, o_2 \dots o_T]$ containing N segments, S_1^N, each segment s_n is defined by its label q_n and the begin and end times $\tau_n, \hat{\tau}_n$, i.e. $s_n = (q_n, \tau_n, \hat{\tau}_n)$. Also denote $s(t)$ as the segment index at time t that

$$s(t) = n \quad if\ \tau_n \le t \le \hat{\tau}_n$$

thus $s(1) = 1$ and $s(T) = N$.

The log-likelihood of the observation and the segment against model λ, $\log P(O_T^T, S_1^n|\lambda)$, is given by

$$\log P(O_T^T, S_1^N|\lambda) = \log(\hat{\pi}_{q_1}) + \sum_{n=1}^{N-1}\log(a_{q_n q_{n+1}}) + \sum_{n=1}^{N}\log(P(O_{\hat{\tau}_n-\tau_n+1}^{\hat{\tau}_n}|q_n)) \quad (2)$$

Here, it is assumed that the transition probability a_{ij} only depends on the segment labels. The expected log likelihood is expressed as the auxiliary function, Q-function,

$$Q(\lambda, \hat{\lambda}) = \sum_{N, S_1^N} P(S_1^N|O_T^T, \hat{\lambda})\log P(O_T^T, S_1^N|\lambda).$$

As described in [3], the solution can be computed by a generalized forward-backward algorithm. Define $\gamma_{t,d}(m)$ as the posterior probability of segment $s(t)$ ends at time t with duration d and $q_{s(t)} = m$. That is,

$$\gamma_{t,d}(m) = p(q_{s(t)} = m, \tau_{s(t)} = t - d + 1, \hat{\tau}_{s(t)} = t|O_T^T).$$

Similar to posterior probability in HMM, $\gamma_{t,d}(m)$ can be decomposed into the forward probability $\alpha_{t,d}(m)$, the probability that all the observations up to time t with the last segment ending at time t with duration d comes from model m, and the backward probability $\beta_{t,d}(m)$, the probability that all the observations from $t+1$ to T given that the last segment that ends at t comes from model m and is of duration d. Thus, in the **E-step**,

$$\alpha_{t,d}(m) = P(O_d^t|m)\sum_{i=1}^{N}\sum_{l=1}^{t-d} a_{im}\alpha_{t-d,l}(i), \quad (3)$$

$$\beta_{t,d}(m) = \sum_{j=1}^{N} a_{mj}\sum_{l=1}^{T-(t+1)} P(O_l^{t+1+l}|j)\beta_{t+l,l}(j), \quad (4)$$

$$\gamma_{t,d}(m) = \frac{\alpha_{t,d}(m)\beta_{t,d}(m)}{p(O_T^T)}. \quad (5)$$

In the **M-step**, the solution for B_m and Σ_m are given by,

$$\hat{B}_m = \left[\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_{t,d}(m)Z_d'Z_d\right]^{-1}\left[\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_{t,d}(m)Z_d'O_d^t\right],$$
$$\hat{\Sigma}_m = \frac{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_{t,d}(m)(O_d^t - Z_d\hat{B}_m)'(O_d^t - Z_d\hat{B}_m)}{\sum_{d=1}^{D}\sum_{t=d}^{T} d\times\gamma_{t,d}(m)}. \quad (6)$$

It is worth noting that for HMM, $R = 1$ and $d = 1$. Thus, Z_d becomes an $N\times 1$ matrix of all 1's and the above equations become the HMM re-estimation equations.

3. EFFICIENT FORWARD-BACKWARD TRAINING ALGORITHM FOR PSM

3.1. Incremental Likelihood Evaluation

In our previous paper [11], we introduced a way to recursively compute the segment likelihood $L(O_d^t|m)$ for segment O_d^t against model m. The key idea is that by re-grouping the terms in the observation likelihood separately by different powers of $(\frac{1}{d-1})$ into $\theta_{i,m}$'s. The factors $(\frac{1}{d-1})^i$ come from the denominator of the elements of the i^{th} column of design matrix Z and are the same for all elements in a column. $d-1$ is used to characterize the duration of a design matrix. Each $\theta_{i,m}$ can be recursively estimated so that the log likelihood of adding one observation can be updated efficiently. That is,

$$L(O_d^t|m) = \sum_{i=0}^{2R}\theta_{i,m}(O_d^t). \quad (7)$$
$$\theta_{i,m}(O_{d+1}^{t+1}) = \theta_{i,m}(O_d^t)(\tfrac{d-1}{d})^i + \delta_{i,m}(o_{t+1}),$$

where o_{t+1} is the $t+1$ observation. For a R^{th} order polynomial model, there are $(2R+1)$ terms in θ. For a quadratic polynomial, the $\delta_{i,m}$ can be written in vector form as:

$$\begin{bmatrix}\delta_{0,m}(o_{t+1})\\ \vdots \\ \\ \\ \delta_{4,m}(o_{t+1})\end{bmatrix} = -\tfrac{1}{2}\begin{bmatrix}(K + (o_{t+1}-\beta_1)\Sigma_m^{-1}(o_{t+1}-\beta_1)')\\ -2(o_{t+1}-\beta_1)\Sigma_m^{-1}{\beta_2}'\\ -2(o_{t+1}-\beta_1)\Sigma_m^{-1}{\beta_3}' + \beta_2\Sigma_m^{-1}{\beta_2}'\\ 2\beta_2\Sigma_m^{-1}{\beta_3}'\\ \beta_3\Sigma_m^{-1}{\beta_3}'\end{bmatrix},$$

where $K = D\log(2\pi) + \log|\Sigma_m|$ is a constant term for each observation and β_i is the i^{th} row of parameter matrix B_m. Then,

$$L(O_{d+1}^{t+1}|m) = L(O_d^t|m) + \Delta_m(O_d^t).$$
$$\Delta_m(O_d^t) = \sum_{k=0}^{2R}\theta_{k,m}(O_d^t)\frac{(d-1)^k - d^k}{d^k} + \delta_{k,m}(o_{t+1}),$$

In the real implementation, Δ does not need to be explicitly computed. Instead, by storing $\theta_{i,m}$'s and updating them, the log likelihood can be computed via Equation 7. Also, notice that the $\delta_{i,m}$'s are independent of d. So, they can be re-used when computing the likelihood of appending the same observation to a segment with a different starting point. This is often the case in EM based training in which different segment starting points have to be considered.

While additional time is required to "transform" $\theta_{i,m}$'s from (d) to $(d+1)$, this is much more efficient compared with re-computing the log-likelihood of the whole segment.

A similar incremental likelihood can be derived to move the segment begin by one frame while fixing the segment end. This

is useful for computing the backward probability. One simple approach is to reverse the polynomial. We use the symbol $\hat{}$ to denote the reversed parameters. Assuming that t is between zero and one, the reversed model is obtained by substituting $\hat{t} = 1 - t$ in the original polynomial.

3.2. Fast Evaluation for Training

EM training can be evaluated by implementing Equations 3-5. Equation 3 shows that the computation of $\alpha_{t,d}(j)$ involves two terms, i) evaluating the segment likelihoods, ii) summing over different α's that depend on both the current segment boundary and the previous segment boundaries, current state j and previous state i. How to make the training more efficient? Similar to Equation 7, the log α's for $d > 1$ can be written as,

$$\begin{aligned}
&\log \alpha_{t,d}(j) \\
&= L(O_d^t|j) + \log\left(\textstyle\sum_{i=1}^{N}\sum_{l=1}^{t-d} a_{ij}\alpha_{t-d,l}(i)\right) \\
&= \Delta_j(O_{d-1}^{t-1}) + \underbrace{L(O_{d-1}^{t-1}|j) + \log\left(\sum_{i=1}^{N}\sum_{l=1}^{t-d} a_{ij}\alpha_{t-d,l}(i)\right)}_{\alpha_{t-1,d-1}(j)} \\
&= \Delta_j(O_{d-1}^{t-1}) + \log(\alpha_{t-1,d-1}(j)),
\end{aligned}$$

where $\Delta_j(O_{d-1}^{t-1})$ is defined in Equation 8. For $d = 1$, a new segment is formed and the recursion is not needed. Therefore,

$$\log \alpha_{t,1}(j) = p(o_t|j) + \log(\textstyle\sum_{i=1}^{N}\sum_{l=1}^{t-1} a_{ij}\alpha_{t-1,l}(i)) \quad .$$

Similarly, β_t's can be reformulated using the incremental reversed log likelihood. Denote

$$\Omega_{t+1,l}(j) = P(O_{t+1}^{t+l}|j)\beta_{t+l,l}(j),$$

being the probability of the future observation and that the segments begin at time $t + 1$.

$$\beta_{t,d}(i) = \textstyle\sum_{j=1}^{N} a_{ij} \sum_{l=1}^{T-(t+1)} \Omega_{t+1,l}(j).$$

For $d \geq 1$, Ω's can be rewritten as

$$\log \Omega_{t,l}(j) = \begin{cases} \hat{\Delta}_j(o_t) + \log \Omega_{t-1,l-1}(j) & d > 1 \\ p(o_t|j) + \log \beta_{t-1,1}(j) & d = 1 \end{cases} \quad .$$

where $\hat{\Delta}_j(o_t)$ denotes the reversed incremental likelihood.

During the **M-step**, the normalized observations, such as ${O_d^t}'O_d^t$ and $O_d^t Z_d$ are required. Terms that are independent of the observations, such as $Z_d' Z_d$, can be precomputed. For normalized observation terms, incremental computation techniques can also be used to make the accumulation more efficient.

As in to HMM, pruning can speed up parameter estimation and recognition. Two types of pruning are implemented, i) fixed beam and ii) fixed number of hypothesis. While aggressive pruning may introduce search error, a conservative pruning that removes highly unlikely paths can still provide good computation savings. In our implementation, both types of pruning are implemented for PSM training and search.

4. DYNAMIC MULTI-REGION PSM

While PSM generalizes the HMM model by capturning the correlations between speech frames, it also imposes more constraints. Consider the case of a multi-region PSM as introduced in [2]. The parts corresponding to the different covariances within a segment can be considered as different PSM states. In HMM, data alignment between states and observations are determined using likelihood. In PSM, observations within a segment are aligned to states/regions uniformly. This lack of dynamic "within segment warping" limits the power of the model with multiple covariances within each segment.

To overcome this constraint, [12] suggested using HMM state alignment to assign speech frames to different PSM regions. The design matrix is then adjusted to estimate a new model with non-uniform warping. While this allows a more flexible alignment, the within segment warping information is only used in model training and can only be performed in conjunction with an HMM model. Furthermore, the HMM alignments may not be suitable for PSM. In this section, we propose a general approach of dynamic warping within PSM segments called dynamic multi-region PSM (DPSM) that allows ML alignment between observation and the regions, and show that using incremental likelihood, DPSM training and recognition is possible.

Assuming that there are u regions, the PSM segment C_k (using the notation of Section 2.1) is expressed in the following form.

$$C_k = \begin{bmatrix} C_{k,1} \\ \vdots \\ C_{k,u} \end{bmatrix} = \begin{bmatrix} Z_{u,1}(N_{k,1})B \\ \vdots \\ Z_{u,u}(N_{k,u})B \end{bmatrix} + \begin{bmatrix} E_{k,1} \\ \vdots \\ E_{k,u} \end{bmatrix},$$

where $C_{k,v}$, $E_{k,v}$ and $N_{k,v}$ denotes the feature vector, residues and duration of the v^{th} region. The new design matrix is composed of u sub-matrices $Z_{u,v}(d_v)$. Instead of normalizing the speech to between 0 and 1, $Z_{u,v}(d_v)$ normalizes the frames to between $\frac{v-1}{u}$ to $\frac{v}{u}$. For example, a three-region segment has $Z_{3,1}$ that normalizes the frames to a region between 0 and $\frac{1}{3}$,$Z_{3,2}$ that normalizes the frames to a region between $\frac{1}{3}$ and $\frac{2}{3}$, etc. Thus,

$$Z_{u,v}(d_v) = \begin{bmatrix} 1 & (\frac{(v-1)}{u} + \frac{0}{(d_v-1)u}) & \cdots & (\frac{(v-1)}{u} + \frac{0}{(d_v-1)u})^{R-1} \\ 1 & (\frac{(v-1)}{u} + \frac{1}{(d_v-1)u}) & \cdots & (\frac{(v-1)}{u} + \frac{0}{(d_v-1)u})^{R-1} \\ \vdots & \vdots & & \vdots \\ 1 & (\frac{v}{u}) & \cdots & (\frac{v}{u})^{R-1} \end{bmatrix}$$

The above is a general formulation of dynamic multi-region PSM without any assumption on how each region relates to another. Possible variations include complete independent regions in which both means and variances are separately estimated, variable variance in which a single trajectory mean is used with different variances, and a single trajectory case in which both the mean and variance are the same across all regions. Irrespective of the variation used, training and recognition of DPSM is very similar to that in traditional PSM case when using the incremental approach.

For recognition, one can consider the regions as the new acoustic units, and the same recognition procedure can be followed. The only subtle difference is that the units now have design matrices mapped to different ranges instead of always between 0 and 1. However, the formulation and algorithm for recognition remain the same.

During the E-step in training, the change is minimal. One can again treat the regions as different acoustic units and collect the γ's separately, except that a new indexing of γ is needed. So, the posterior probability of model m, region v with region duration d_v is denoted as $\gamma_{t,d_v}(m, v)$.

During the M-step, the update of the model depends on the variation. If the regions are completely independent, the only difference between using multiple regions, and multiple whole seg-

ments, is the range of normalization in the design matrices. If only a single mean is used, all the regions contribute to the re-estimation of the segment mean. It can be shown that the new mean for model m with u regions is given by,

$$B_m = \left[\sum_{v=1}^{u}\sum_{t=1}^{T}\sum_{d_v=1}^{t}\bar{\gamma}_{t,d_v}(m,v)Z_{u,v}(d_v)'Z_{u,v}(d_v)\right] \left[\sum_{v=1}^{u}\sum_{t=1}^{T}\sum_{d_v=1}^{t}\bar{\gamma}_{t,d_v}(m,v)Z_{u,v}(d_v)'O_{d_v}^{t}\right]. \quad (8)$$

This is quite similar to Equation 6 except that observations are normalized with different sub-design matrices.

The new variances of different regions are given by

$$\Sigma_{m,v} = \frac{\sum_{t=1}^{T}\sum_{d_v=1}^{t}\bar{\gamma}_{t,d_v}(m,v)(O_{d_v}^{t}-Z_{u,v}(d)B_m)'(O_{d_v}^{t}-Zu,v(d)B_m)}{\sum_{t=1}^{T}\sum_{d_v=1}^{t}d_v\bar{\gamma}_{t,d_v}(m,v)}.$$

If a single variance is used across all regions, Σ_m is either obtained by summing over all the regions or merging the region variances. That is

$$\Sigma_m = \frac{\sum_{v=1}^{u}\Sigma_{m,v}\times\left[\sum_{t=1}^{T}\sum_{d_v=1}^{t}d_v\bar{\gamma}_{t,d_v}(m,v)\right]}{\sum_{v=1}^{u}\sum_{t=1}^{T}\sum_{d_v=1}^{t}d_v\bar{\gamma}_{t,d_v}(m,v)}.$$

5. EXPERIMENTS

Two sets of experiments, TIMIT classification and recognition were performed to demonstrate the performance of efficient forward-backward training and the dynamic multi-region PSM.

All experiments were performed using Mel-frequency cepstral coefficients (MFCC) with delta and accelerated coefficients at a rate of 200 frames per second, and SX and SI utterances in the standard TIMIT train and test set [9] were used. The phone set was similar to [10] with minor modifications (el => ax + l, en => ax + n). No language model was used and the insertion penalty was tuned empirically. In all the PSM experiments 2^{nd} order PSM was used, and the HMM baseline was obtained using three-state left-to-right HMM models which were trained using the EM-algorithm. Only one single Gaussian mixture was used for both PSM and HMM.

Classification was performed using different types of PSM variations to show the effectiveness of the improved models. Results are summarized in Table 5 using the single mixture monophone HMM, which was trained using EM algorithm, as the baseline. The TIMIT phone alignment and Viterbi algorithm were used for training the DPSM. The traditional PSM with uniform alignment and single variance is not as good as that of the HMM; allowing multiple covariances does not give a significant gain. However, allowing within segment warping gives a significant improvement with and without multiple covariances. The performance of DPSM with three covariances, which have the same number of parameters as HMM, matches that of HMM. While using 3 independent regions, DPSM performs slightly better that HMM.

In all the PSM recognition experiments used DPSM with a single mean and three covariance. The PSM performance is slightly better than that of HMM even though both have the same number of parameters. In addition, we compared the performance of different training method, Viterbi training and EM training with different number of active hypotheses in pruning. The results show that EM training gives better performance and for this task, more hypotheses in pruning are not needed.

	Accuracy%
HMM	61.63
Tradition PSM	57.62
PSM with 3 regions, 3 Cov.	57.67
DPSM with 3 regions, 1 Cov.	60.08
DPSM with 3 regions, 3 Cov.	61.83
DPSM with 3 Indenpendent regions	62.83

Table 1. Classification result using different types of PSM

	HMM	DPSM Vit.	DPSM (EM 15 hyp.)	DPSM (EM 30 hyp)
Accuracy%	45.43	46.29	46.59	46.59

Table 2. Recognition result using different training methods

6. CONCLUSION

In this paper, an efficient forward-backward training approach for Polynomial Segment Model, which allows PSM models to be trained via maximum likelihood was proposed. We also introduced dynamic multi-region PSM so that the models can capture the dynamic expansion and compression of the phone units. We showed that with this dynamic multi-region PSM and EM training, classification and recognition performance of DPSM is better than that of the tradition PSM or HMM.

Via a presentation of the efficient forward-backward training and evaluation algorithm formulations, we demostrated that the framework for both HMM and PSM are very similar. In fact, a three-state HMM can be viewed as a DPSM of 0^{th} order and three independent regions. This will help to further explore the possibility of merging both models. Also, the application of techniques, that are well developed in HMM, to PSM is feasible under this framework.

7. REFERENCES

[1] H. Gish and K. Ng, "A Segmental Speech Model with Applications to Word Spotting," Proc. ICASSP-93

[2] H. Gish and K. Ng, "Parametric Trajectory Models for Speech Recognition," Proc. ICSLP, 1996

[3] M. Ostendorf, "From HMM's to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition," IEEE Trans, SAP., vol 4, no. 5, pp. 360-378, 1996

[4] W. Holmes and M. Russell, "Probabilistic-trajectory segmental HMMs," Computer Speech and Language 13, 1999

[5] T. Fukada, M. Bacchian, K.K. Paliwal, Y. Sagisaka, "Speech Recognition Based on Acoustic Derived Segment Units," ICSLP-96

[6] J. Goldberger, D. Burshtein, and H. Franco, "Segmental Modeling Using a Continuous Mixture of Non-parametric Models," ESCA. Eurospeech97 pp. 1195-1198

[7] L. Deng, M. Aksmanovic, D. Sun, and J. Wu, "Speech recognition using hidden Markov models with polynomial regression functions as nonstationary states," IEEE Trans. SAP.,vol2,no.4, pp. 507-520, 1994

[8] R. Iyer, H. Gish, M. Siu and G. Zavaliagkos, "Hidden Markov Model for Trajectory Modeling", Proc. ICSLP, 1998

[9] V. Zue, S. Seneff, and J. Class. Speech Database Development at MIT: TIMIT and Beyond. Speech Communication, 9(4): 351-356, August,1990

[10] K.F. Lee, "Automatic Speech Recognition: The Development of the SPHINX System," Kluwer Academic Publisher, 1989

[11] C.F. Li, M. Siu, "An Efficient Incremental Likelihood Evaluation for Polynomial Trajectory Model using with Application to Model Training and Recognition," Proc. ICASSP-03

[12] J. Lei, X. Bo, "Introduce Segmental Inner Time Warping into Parameteric Trajectory Segment Model for LVCSR", Proc. ICSLP-02