A MULTI-RESOLUTION HIDDEN MARKOV MODEL USING CLASS-SPECIFIC FEATURES

Paul M. Baggenstoss

Naval Undersea Warfare Center
Newport RI, 02841, USA
phone: (+001) 401-832-8240
email: p.m.baggenstoss@ieee.org
web: www.nuwc.navy.mil/npt/csf/index.html
This work was supported by the Office of Naval Research ONR321US

ABSTRACT

We address the problem in signal classification applications, such as automatic speech recognition (ASR) systems that employ the hidden Markov model (HMM), that it is necessary to settle for a fixed analysis window size and a fixed feature set. This is despite the fact that complex signals such as human speech typically contain a wide range of signal types and durations. We apply the probability density function (PDF) projection theorem to generalize the hidden Markov model (HMM) to utilize a different features and segment length for each state. We demonstrate the algorithm using speech analysis so that long-duration phonemes such as vowels and short-duration phonemes such as plosives can utilize feature extraction tailored to the their own time scale.

1. INTRODUCTION

The Hidden Markov Model (HMM) [1] combined with spectral analysis using cepstral coefficients [2] on fixed-length analysis windows remains at the forefront of automatic speech recognition (ASR) technology. One problem with this architecture is the necessity of using a fixed analysis window size. This constraint is a problem because in speech and other natural processes, the various phenomena that are being tested (such as phonemes in speech) may occur with differing time scale. The window size used on speech analysis is a compromise between phonemes with long time scale such as vowels and phonemes with shorter time scales such as plosives. The need for a fixed-size window arises from the fundamental probabilistic approach that underlies the method and depends on the comparison of likelihood functions formed on a common feature space. One could not directly compare two likelihood functions if they are defined on different feature spaces. Even if pains are taken to normalize the behavior of similar features obtained from differing-size data windows, the fundamental basis for comparison is suspect.

With the introduction of the class-specific feature theorem [3], [4], [5], and later the probability density function (PDF) projection theorem (PPT) [6], the freedom now exists to use a different feature set for each class, even for each state in a HMM [7], and as we now show, different analysis window lengths for each state. Thus, the topic of this paper is to apply the PPT to the problem of using varying-size analysis windows within the framework of a HMM.

2. THE HMM AND MULTI-RESOLUTION HMM (MRHMM) ON RAW DATA

We assume familiarity with hidden Markov models (HMMs). A good reference is an article by Rabiner [1] from which we borrow some notation. If we ignore the effects of overlapped processing, the underlying assumption when a time-series is segmented for processing is that the data in two different segments are conditionally statistically independent (CSI). In other words, the data in two segments are statistically independent conditioned on the system states in the two segments being known. The CSI property enables the efficient calculation of the joint PDF using the forward procedure. Let there be a raw data time-series, denoted by $\mathbf{X}$, consisting of an integer multiple of T samples, where T is the basic time quantization. The traditional approach, which we describe simply as the HMM, is to divide the data into uniform T-sample segments which are to be processed separately. Let $\mathbf{x}_t$ represent the data in time-step t consisting of data samples $1+(t-1)T$ through tT. In the HMM, it is assumed that:

1. during any T-sample segment, the data is governed by one of M possible states.
2. any two samples, no matter how close together , that are contained in two different segments, are CSI.

For the MRHMM, however, we assume that:

1. during any T-sample segment, the data is governed by one of M possible states.
2. Once the system transitions to state s, it must remain in that state for a number of length-T time steps divisible by K_s, or nK_sT samples, where K_s is the integer minimum duration parameter for state s, and $n \geq 1$. In other words, the dwell time in state s is made up of *meta-segments* of size K_sT sample.
3. Two data samples x_i and x_j are assumed to be CSI and are processed separately if and only if a state transition has occurred between the samples and/or they are contained in different meta-segments. Otherwise, samples x_i and x_j are processed jointly.
4. Because the above assumption is too restrictive in ordinary systems, we need to allow for arbitrary-length dwell times in a given signal state. Let the term *class* refer to a certain signal state or phenomenon. Let each class be associated with several states, each with a different K_s. If, for example, a class is associated with states having $K_s = \{2,3,4,6,8\}$, we can produce a large number of different dwell times using combinations of meta-segments with the above sizes. We call these states *slave states* or

slave partions because they are slaved to a signal class.

Let $Q = [s_1, s_2 \ldots s_N]$ be a set of state values, where $1 \leq s_t \leq M$, $1 \leq t \leq N$. We call Q a *trajectory* because it defines one of the many paths through the state diagram or trellis. The restrictions imposed above having to do with dwell time can be imposed by a structured STM by properly expanding and structuring π and $\mathbf{A}$. For each state s, we can define a *partition* of states, which we call *wait states*, of size K_s. Let $\mathbf{A}^e$ be the *expanded* MRHMM STM and let π^e be the expanded set of prior state probabilities. We structure $\mathbf{A}^e$ so that state transitions *into* the state s partition are only allowed into the first wait state. From the first wait state, the state is forced to increment to the second, third, ... and finally to wait state K_s. From wait state K_s, the state is allowed to transition to the first wait state of any state partition. Note that although $\mathbf{A}^e$ is dimension $M^e \times M^e$ where $M^e = \sum_{m=1}^{M} K_m$, there are only M^2 free parameters in $\mathbf{A}^e$.

At this point, the MRHMM can be seen as nothing more than a HMM with a specially structured π and $\mathbf{A}$. But the more important difference, which we will explain below, is in the way that $p(\mathbf{X}|Q)$ is calculated. For the moment, let us talk about our goals. We seek an algorithm to solve the following four problems:

1. Segmentation. Find the most likely trajectory through the trellis subject to the restrictions described above. Knowing the most likely trajectory is tantamount to segmentation because as the trajectory dwells in a given state, it defines a segment.
2. State probabilities. Determine the *a posteriori* state probabilities $\gamma_{t,m} = p(s_t = m|\mathbf{X})$. This is a more complete description of the trajectories than knowing the single most likely trajectory.
3. Joint PDF. The joint likelihood function of all the data given the model is given by

$$L(\mathbf{x}) = \sum_{Q \in \mathcal{Q}} p(\mathbf{X}|Q)\, p(Q), \tag{1}$$

   where $\mathcal{Q}$ is the set of all possible trajectories and $P(Q)$ is the *a priori* probability of a given trajectory through the trellis. Note that $L(\mathbf{X})$ averages $p(\mathbf{X}|Q)$ over all trajectories through the trellis weighted by the probability of the trajectory. Invalid trajectories have zero contribution.
4. Re-estimation. We would like to estimate the model parameters from the data. Parameters include π, $\mathbf{A}$, and the parameters θ_s of the state-conditional data PDFs. In section 3 we will explain how the raw data PDFs can be computed from low-dimensional feature PDFs.

For the HMM, the above problems are solved by the *forward procedure* and the associated *backward procedure* and the Baum-Welch algorithm [1]. For the MRHMM, we need to adapt these algorithms, not only by structuring the π and $\mathbf{A}$, but by changing the way that $p(\mathbf{X}|Q)$ is calculated.

At this point, all we have is a HMM with restrictions on the dwell times of each state. To change the HMM into a MRHMM, we need to define *partial PDF values*. To understand partial PDF values, refer to figure 1. Here we see two potential segmentations of the data into meta-segments. Let's assume that we process the meta-segments in order to determine the PDF of the raw meta-segment data given the state. In section 3 we will show how we calculate these meta-segment PDFs using low-dimensional feature PDFs. Although the data is processed in meta-segments, the partial PDF values are computed at each time step by taking the K-th root of the likelihood value in the meta-segment. Partial PDF values are constant within a meta-segment. This value is entered at each time-step within the meta-segment.

Unlike actual PDF values, it is not always valid to compare the partial PDF values from different segmentatons at a given time-step. If the partial PDF values are derived from different meta-segment sizes, the comparison is meaningless. On the other hand, and this is the main point, *the product of the partial PDF values along any two valid paths, subject to restrictions of dwell time, can be fairly compared.* By restricting the dwell times in the way described above, the product of partial PDF values along valid paths always equals the product of the full PDF values of the set of meta-segments corresponding to the path. This is because state transitions are not allowed until the meta-segment is complete. To implement the MRHMM, we gather the partial PDF values into a matrix. Let $P_{t,q}$ be the partial PDF value at time t and wait state q. Recall there are M^e wait states in total. The range of wait states is divided into partitions and each partition corresponds to a particular state s. Let q' be the differential wait state taking values from 1 to K_s within the partition for state s. Matrix entry $P_{t,q}$ is equal to the K_s-root of the PDF of the length K_s meta-segment that started at time step $(t - q' + 1)$. Note that $P_{t,q}$ is constant along diagonal lines within a partition. To calculate (1), we apply the standard *forward procedure* using the expanded state probabilities $\mathbf{A}^e$ and π^e and treating $P_{t,q}$ as the state probabilities of a normal HMM. At this point we have a raw-data based MRHMM model that we can compute efficiently using the forward procedure operating on $P_{t,q}$. To create a feature-based MRHMM model, we need only to apply the PPT.

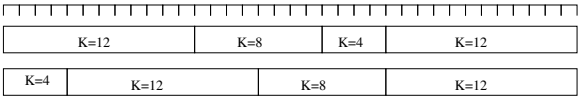

Figure 1: Two possible segmentations for a section of data equal to 36 time steps.

3. CLASS-SPECIFIC MULTI-RESOLUTION CLASS-SPECIFIC (CS-MRHMM)

The standard feature-based HMM is the same as the raw-data based HMM with the raw data segments $\mathbf{x}_t$ replaced by the feature vector

$$\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2 \ldots \mathbf{z}_N\},\ \mathbf{z}_t = T(\mathbf{x}_t).$$

With this simple replacement, the forward procedure computes the feature-based likelihood function

$$L(\mathbf{Z}) = \sum_{Q \in \mathcal{Q}} p(\mathbf{Z}|Q)\, P(Q), \tag{2}$$

For the CS-MRHMM, we need to use the PPT to transition to the feature domain. Let

$$\mathbf{x}_t^K = [x_{1+(t-1)T} \ldots x_{(t+K-1)T}],$$

be the length KT sample *analysis window* which starts at sample $1 + (t-1)T$. It includes segments $\mathbf{x}_t$ through $\mathbf{x}_{t+K-1}$.

The term $p(\mathbf{x}_t^K|s)$ will be calculated using the PDF projection theorem [6]. As we have written several publications on the topic including the tutorial article [8], we describe the method only briefly. Let $\mathbf{x}$ be a general segment of raw time-series data. Let $\mathbf{z}_s = T_s(\mathbf{x})$ be a feature set calculated from $\mathbf{x}$ specifically designed for state s. Let $\hat{p}(\mathbf{z}_s|s)$ be a PDF estimate of the feature set $\mathbf{z}_s$ based on training data from state s. The feature likelihood function is *projected* from the feature space to the raw data by pre-multiplying by the J-function as follows:

$$\hat{p}_p(\mathbf{x}|s) = J(\mathbf{x};T_m,H_{0,s})\ \hat{p}(\mathbf{z}_s|s). \tag{3}$$

The function $\hat{p}_p(\mathbf{x}|s)$ can be regarded as a function only of $\mathbf{x}$ by substituting $T_s(\mathbf{x})$ for $\mathbf{z}_s$ and can be shown to integrate to 1 over $\mathbf{x}$ (thus it is a PDF). The J-function is a unique function of $\mathbf{x}$ determined precisely from the feature transformation T_s and the class-dependent reference hypothesis $H_{0,s}$:

$$J(\mathbf{x};T_s,H_{0,s}) = \frac{p(\mathbf{x}|H_{0,s})}{p(\mathbf{z}_s|H_{0,s})}. \tag{4}$$

Since $J(\mathbf{x};T_s,H_{0,s})$ is determined *a priori* without regard to training data, it can be considered the *untrained* part of $\hat{p}_p(\mathbf{x}|s)$, while $\hat{p}(\mathbf{z}_s|s)$ is the trained part.

While it is true that $\hat{p}_p(\mathbf{x}|s)$ is a PDF, it is only an estimate of $p(\mathbf{x}|s)$. The degree to which $\hat{p}_p(\mathbf{x}|s)$ is a good estimate of $p(\mathbf{x}|s)$ depends on (a) the accuracy of $\hat{p}(\mathbf{z}_s|s)$ and (b) the degree to which $\mathbf{z}_s$ is a *sufficient statistic* for the binary hypothesis test between s and $H_{0,s}$. In the rare case that $\mathbf{z}_s$ is in fact a sufficient statistic, the accuracy of $\hat{p}_p(\mathbf{x}|s)$ depends only upon the accuracy of the low-dimensional PDF estimate $\hat{p}(\mathbf{z}_s|s)$. The J-function takes many forms [6], one of which can be used when $\mathbf{z}_s$ are maximum likelihood (ML) estimates of a set of parameters. In this case, $J(\mathbf{x};T_s,H_{0,s})$ has a simple form based on the Fisher's information matrix [6].

4. PRACTICAL IMPLEMENTATION DETAILS

When values of K_s are large, a highly overlapped set of windows is required. The amount of processing required can be mitigated, by recursive processing. For example, the FFT or autocorrelation function (ACF) of a segment can be updated to reflect data that has been shifted out and data that has been shifted in [9]. Applying the MRHMM to real data warrants additional details beyond what has been so far described.

4.1 Training the CS-MRHMM.

In the standard Baum-Welch algorithm for re-estimation of HMM parameters [1], the state feature PDFs for state s are trained by maximizing log-likelihood functions weighted by $\gamma_{s,t}$. Since the standard HMM does not differentiate between wait states, we would need to a separate PDF estimate for each wait state. However, for the CS-MRHMM, there are only PDF estimates associated with the initial wait states, the first wait states of each partition. Logically, the CS-MRHMM produces values of $\gamma_{q,t}$ that are constant in diagonal streaks in a partition. That is, $\gamma_{q,t} = \gamma_{q+1,t+1}$ if wait states q and $q+1$ are in the same partition. Thus, in the CS-MRHMM, each analysis window can be traced to a given constant-valued streak in the $\gamma_{q,t}$ matrix. When training the CS-MRHMM, the features from the associated analysis window are weighted by the corresponding value of $\gamma_{q,t}$ in the streak. Training becomes slightly more complicated, however, once we consider slave partitions and if the number of signal states exceeds the number of signal classes. While each partition is associated with a PDF estimate, we may not want all partition PDF estimates to be independent. To remedy this situation, we "gang together" all partitions that associate with a given signal class. To gang partitions, we first create a compressed version of $\gamma_{q,t}$, denoted by $\gamma^c_{m,t}$, which sums $\gamma_{q,t}$ over all wait states associated with signal class m. Then we then weight an analysis window by the *smallest* value of $\gamma_{m,t}$ in the set of time steps t contained in the analysis window. This works very well in practice but is a clear departure from the Baum-Welch algorithm and may produce an algorithm without guaranteed monotonicity.

4.2 Efficient Implementation

The number of wait states in the expanded HMM problem can be very large. The forward and backward procedures have a complexity of the order of the square of the number of states. Thus, an efficient implementation of the forward and backward procedures and Baum-Welch algorithm may be needed that takes advantage of the redundancies in the expanded problem. We have obtained a time reduction factor of 42 with a problem that had 7 signal classes and expanded to 274 wait states. The two algorithms were tested to produce the same results within machine precision.

5. EXAMPLES

5.1 Simulated Data

To illustrate the concepts, we tested the concept of the CS-MRHMM using simulated data. To independent identically distributed (iid) Gaussian noise, we added a low frequency (LF) pulse of autoregressive (AR) process of 128 samples in length with a peak frequency response of 0.4 radians per sample, followed by a random-length gap of at least 256 samples, followed by high frequency (HF) pulse of AR process of 64 samples with a peak frequency response of 1.2 radians per sample. An example of the signal and noise is shown in Figure 2. We implemented the MRHMM with three signal states, each corresponding to a signal class : "noise", "LF pulse", and "HF pulse". We used nine partitions including six slave partitions. The elemental segment length was $T = 32$ samples. There were a total of 25 wait states. Parameters of the nine partitions are listed in table 1. Autoregressive (LPC) features of model order P (see table 1) were extracted by overlapped window processing. A separate feature processor was used for each combination of K and P. Features were shared between partitions that had the same K and P values.

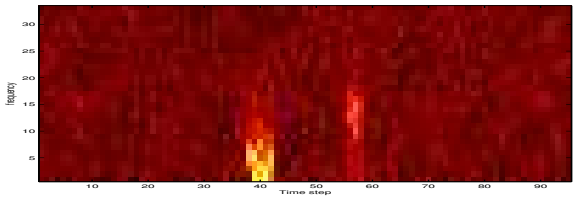

Figure 2: Example of spectrogram of synthetic data. The data consists of three signal classes. Class 1 (noise) occurs first, then a low-frequency pulse of duration 128 samples, then noise, then a high-frequency pulse of duration 64 samples.

Partition	Signal class	KT	K	P
1	Noise	256	8	4
2	Noise	128	4	4
3	Noise	64	2	4
4	Noise	32	1	3
5	LF Pulse	128	4	4
6	LF Pulse	64	2	4
7	LF Pulse	32	1	3
8	HF Pulse	64	2	4
9	HF Pulse	32	1	3

Table 1: Partition parameters for the illustrative example. K is the partition length in elemental segments. KT is the length of the partition in samples. Parameter P is the autoregressive (AR) model order (same as LPC model order).

Analysis windows were shifted always by the elemental segment length of 32 samples for each update, so the amount of overlap depended on the length of the analysis window. To handle end effects, data was assumed to wrap around in time.

Features were extracted from each analysis window by first taking the FFT, computing the magnitude squared, then computing the inverse-FFT to produce the autocorrelation function (ACF). The Levinson algorithm was used to produce the reflection coefficients of order P. The total power in window is also stored as the $P+1$st feature. The J-function [6] is obtained by use of the saddle-point approximation [10]. Further details of the implementation details of the AR models can be found in [8].

In Figure 3, we see the partial PDF matrix $P_{i,m}$ for a typical sample. Wait states $q = 1$ through $q = 15$ are associated with the "Noise" signal class. wait states $q = 16$ through $q = 22$ are associated with the "LF Pulse" signal class, and wait states $q = 23$ through $q = 25$ are associated with the "HF Pulse" signal class. The gamma probabilities are a by-

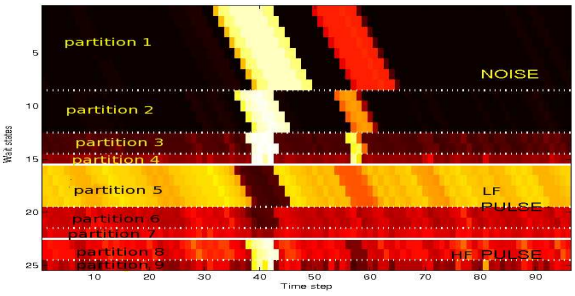

Figure 3: Partial PDF matrix $P_{i,m}$ showing devisions between signal classes (solid horizontal lines) and between wait state partitions (dotted lines). Higher probability is darker.

product of the Baum Welch algorithm [1] and indicate the relative probability of each wait state given the data. The gamma probabilities corresponding to figure 3 are shown in Figure 4. This figure can be interpreted as the trajectory through figure 4 that pick up the highest probabilities while meeting the restrictions set by the state transition matrix.

Note that the wait states for "LF pulse" ($q = 16$ through $q = 19$) are clearly seen where the pulse occurs. The same is true of the "HF pulse" event ($q = 23$ through $q = 24$) . It is possible to see various competing trajectories through

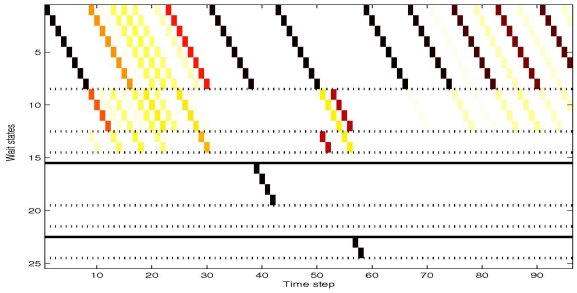

Figure 4: Wait state probabilities $\gamma_{q,t}$ for illustrative example.

the trellis. Note for example in time steps 43 through 56, the noise gap between the two pulses, the HMM is in the noise signal class. In steps 43-50, it is in partition 1, (wait states $q = 1$ through $q = 8$). Then after exiting wait state $q = 4$, it has located two possibilities to span the six time steps remaining before HF pulse occurs. It can either go into partition 2 (wait states $q = 9$ through $q = 12$) then partition 3 (wait states $q = 13$ through $q = 14$), or it can choose the reverse, partition 3 then partition 2.

The gamma probabilities can be collapsed to indicate just the signal classes, as shown in Figure 6. The class probabili-

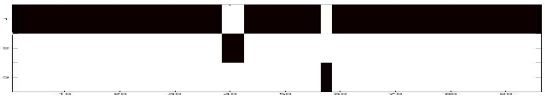

Figure 6: Signal class probabilities calculated by summing figure 4 over the wait states of each class. Darker is higher probability. Time runs from left to right. Signal class identity is on the vertical axis: top: Noise, Middle: LF pulse, Bottom: HF pulse.

ties (figure 6) is an accurate indication of the true content of the data to a time resolution of $T = 32$ samples.

5.2 Speech Data

We used the CS-HMM to analyze the spoken word "stool" at 16 kHz sample rate. Space restrictions do not permit a detailed description of the experiment. We identified seven signal classes and assigned values of K and P (LPC order) (1) **Noise** used for both background and the "T" closure: $K = 12$ or 384 samples, $P = 7$, (2) **"S"** : $K = 12$ or 384 samples, $P = 7$, (3) **"T" Burst** : $K = 4$ or 128 samples, $P = 5$, (4) **"T" Aspiration** : $K = 8$ or 256 samples, $P = 6$, (5) **"oo" vowel part 1**: $K = 24$ or 768 samples, $P = 8$, (6) **"oo" vowel part 2**: $K = 24$ or 768 samples, $P = 8$, (7) **"L"** : $K = 24$ or 768 samples, $P = 8$. After adding slave partitions, we had a total of 36 partitions and a total of 258 wait states. The expanded STM was 258 by 258. Using efficient programming, neither the partial probability matrix nor the expanded STM actually need to be created. Figure 5 shows the result of analysis of one example with the CS-MRHMM. Important to note is that the three components of the "T" can be clearly seen by observing $\gamma^c_{m,t}$. This may have applications in not just ASR, but automatic phoneme labeling and any application where accurate segmentation is desired.

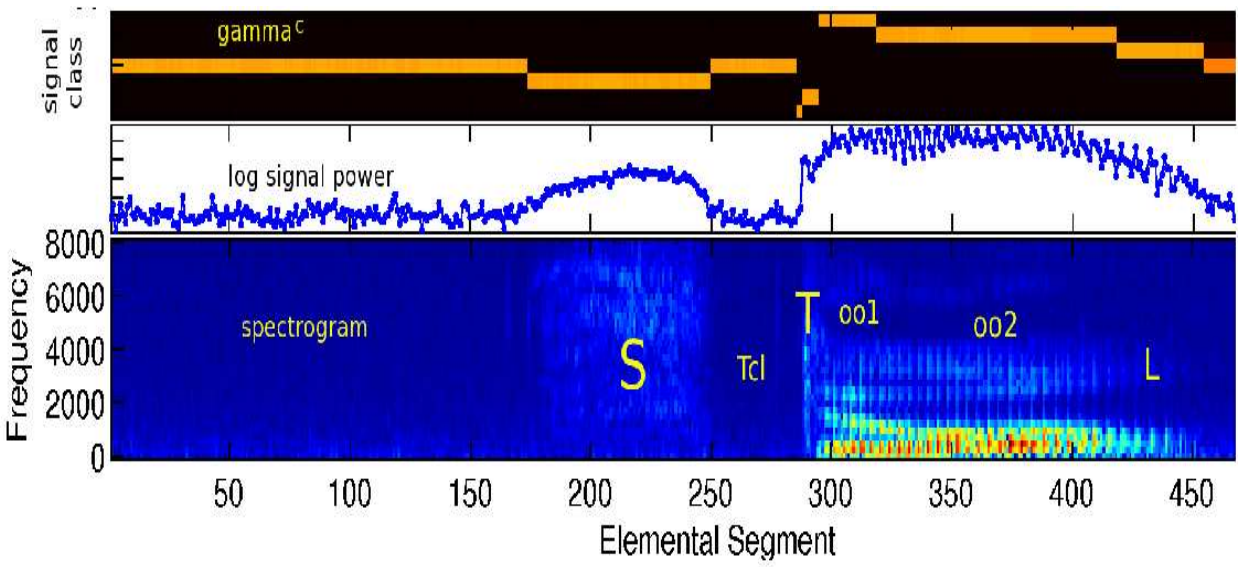

Figure 5: Example of CS-MRHMM operating on the word "stool". From top to bottom: compressed gamma probabilities $\gamma^c_{m,t}$, log signal power, and spectrogram. The dwell time in each state is composed of sequences of meta-segments that are assigned to each state. Short analysis windows have been employed for the "T", while longer processing has been used for background noise and the sounds "S", "oo" and "L". The three components of the "T" can be clearly identified.

REFERENCES

[1] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, February 1989.

[2] J. W. Picone, "Signal modeling techniques in speech recognition," *Proceedings of the IEEE*, vol. 81, no. 9, pp. 1215–1247, 1993.

[3] P. M. Baggenstoss, "Class-specific features in classification.," in *IASTED International Conference on Signal and Image Processing*, 1998.

[4] S. Kay, "Sufficiency, classification, and the class-specific feature theorem," *IEEE Trans. Information Theory*, vol. 46, pp. 1654–1658, July 2000.

[5] P. M. Baggenstoss, "Class-specific features in classification.," *IEEE Trans Signal Processing*, pp. 3428–3432, December 1999.

[6] P. M. Baggenstoss, "The PDF projection theorem and the class-specific method," *IEEE Trans Signal Processing*, pp. 672–685, March 2003.

[7] P. M. Baggenstoss, "A modified Baum-Welch algorithm for hidden Markov models with multiple observation spaces.," *IEEE Trans. Speech and Audio*, pp. 411–416, May 2001.

[8] P. M. Baggenstoss, "The class-specific classifier: Avoiding the curse of dimensionality (tutorial)," *IEEE Aerospace and Electronic Systems Magazine, special Tutorial addendum*, vol. 19, pp. 37–52, January 2002.

[9] P. Baggenstoss, "Time-series segmentation," *United States Patent 6907367*, June 2005.

[10] S. M. Kay, A. H. Nuttall, and P. M. Baggenstoss, "Multidimensional probability density function approximation for detection, classification and model order selection," *IEEE Trans. Signal Processing*, pp. 2240–2252, Oct 2001.