A MODIFIED BAUM-WELCH ALGORITHM FOR HIDDEN MARKOV MODELS WITH MULTIPLE OBSERVATION SPACES

Dr. Paul M. Baggenstoss

Naval Undersea Warfare Center
Newport RI, 02841
401-832-8240
p.m.baggenstoss@ieee.org

ABSTRACT

In this paper, a new algorithm based on the Baum-Welch algorithm for estimating the parameters of a hidden Markov model (HMM) is presented. It allows each state to be observed using a different set of features rather than relying on a common feature set. Each feature set is chosen to be a sufficient statistic for discrimination of the given state from a common "white-noise" state. Comparison of likelihood values is possible through the use of likelihood ratios. The new algorithm is the same in theory as the algorithm based on a common feature set, but without the necessity of estimating high-dimensional probability density functions (PDF's). A simulated data example is provided showing superior performance over the conventional HMM.

1. INTRODUCTION

Consider a hidden Markov model (HMM) for a process with N states numbered S_1 through S_N. Let the raw data be denoted $\mathbf{X}[t]$, for time steps $t = 1, 2, \ldots, T$. The parameters of the HMM, denoted Λ, comprise the state transition matrix $A = \{a_{ij}\}$, the state prior probabilities u_j, and the state observation densities $b_j(\mathbf{X})$, where i and j range from 1 to N. These parameters can be estimated from training data using the Baum-Welch algorithm [1], [2]. But, because $\mathbf{X}[t]$ is often of high dimension, it may be necessary to reduce the raw data to a set of features $\mathbf{z}[t] = T(\mathbf{X}[t])$. We then define a new HMM with the same A and u_j, but with observations $\mathbf{z}[t]$, $t = 1, 2, \ldots, T$ and the state densities $b_j(\mathbf{z})$ (we allow the argument of the density functions to imply the identity of the function, thus $b_j(\mathbf{X})$ and $b_j(\mathbf{z})$ are distinct). This is the approach used in speech processing today where $\mathbf{z}[t]$ are usually a set of cepstral coefficients. If $\mathbf{z}[t]$ is of low dimension, it is practical to apply probability density function (PDF) estimation methods such as Gaussian Mixtures to estimate the state observation densities. Such PDF estimatimation methods tend to give poor results above dimensions above about 5 to 10 [3] unless the features are exceptionally well-behaved (are close to independent or multivariate Gaussian). In human speech, it is doubtful that 5 to 10 features can capture all the relevant information in the data. Traditionally, the choices have been (1) use a smaller and insufficient features set, (2) use more features and suffer PDF estimation errors, or (3) apply methods of dimensionality reduction [4]. Such methods include linear subspace analysis [5], projection pursuit [6], or simply assuming the features are independent (a factorable PDF). All these methods involve assumptions that do not hold in general. We now present a new method based on sufficient statistics that is completely general and is based on the class-specific classifier [7]. Central to the classical Baum-Welch algorithm [1] is the ability to calculate $\gamma_t(j)$, the probability that state j is true at time t given the entire observation sequence $\mathbf{X}[1], \ldots, \mathbf{X}[T]$. These probabilities depend on the *forward* and *backward* probabilities, which in turn depend on the state likelihood functions $b_j(\mathbf{X})$. It is possible to re-formulate the problem to compute $\gamma_t(j)$ using low-dimensional PDF's. Consider an additional "common" state S_0 for which the raw data is pure independent Gaussian noise. Using the likelihood ratios

$$\frac{b_j(\mathbf{X})}{b_0(\mathbf{X})}, \quad 1 \le j \le N \tag{1}$$

in place of the likelihood functions $b_j(\mathbf{X})$ in the Baum-Welch algorithm, causes only a scaling change since the denominator is independent of j. Each of the ratios in (1) may be thought of as an optimal binary test or detector that can be re-written in terms of a state-dependent sufficient statistic, denoted $\mathbf{z}_j$, $j = 1, 2, \ldots, N$. Thus, we have

$$\frac{b_j(\mathbf{X})}{b_0(\mathbf{X})} = \frac{b_j(\mathbf{z}_j)}{b_0(\mathbf{z}_j)}, \quad 1 \le j \le N, \tag{2}$$

which follows from the sufficiency of $\mathbf{z}_j$ for state S_j vs. state S_0. Note that we use the same symbol $b_j(\cdot)$ to represent the density of $\mathbf{X}$ and $\mathbf{z}_j$, however they are distinct (the argument of the function should make it clear which function is implied). Since these tests do not involve the other states, they are simpler, and $\mathbf{z}_j$ can be individually lower in dimension than $\mathbf{z}$. Note that this requires *prior knowledge* about each state, i.e., that a given state is best distinguished from S_0 by a given statistic[1]. This *prior knowledge* is often available, but not used. It only requires some knowledge about the signal characteristics in each state, the same knowledge necessary for choosing features. Keep in mind that each

This work was supported by Office of Naval Research (ONR-321US)

[1] It is *not* neccessary to know which state is true at any given time

state does not need a distinct set of statistics. The class-specific HMM is an extension of the conventional HMM. If each state has the same set of statistics, the class-specific HMM specializes to the conventional HMM. When designing a class-specific HMM, one could allocate a subset of states to each class of data. Each subset of states would require only the statistics apropriate for that class of data. In fact, an entirely different processing for feature extraction, tailored to that type of signal can be used. This can solve the problem that occurs in speech processing when short-duration plosives are forced to be analyzed using long analysis windows which are more appropriate for vowels. When compared to the standard feature-based HMM using a common feature set $\mathbf{z}$, the density of the numerators may be estimated using fewer training samples. Or, for a fixed number of training samples, more accuracy and robustness can be obtained. In addition, experience has shown that, difficulties encountered in initialization of the Baum-Welch algorithm are greatly diminished when this *prior knowledge* is used.

A few words about the denominator densities $b_0(\mathbf{z}_j)$ is necessary. These densities may be approximated from synthetic white noise or derived analytically. For data that significantly departs from S_0, all denominators can tend to zero. In this case, the denominator densities $b_0(\mathbf{z}_j)$ must be solved for analytically or with an approximation valid in the far tails. The task of finding solutions valid in the tails is helped by the fact that S_0 is characterized by pure *iid* Gaussian noise. In the computer simulations, we use exact formulas or approximations valid in the tails.

2. MATHEMATICAL RESULTS

Details of the derivation are to be published [8]. Here, we provide only the algorithm.

2.1. The Class-Specific HMM

2.1.1. The class-specific forward procedure

1. Initialization:

$$\alpha_1(j) = u_j \frac{b_j(\mathbf{z}_j[1])}{b_0(\mathbf{z}_j[1])}, \quad 1 \le j \le N \tag{3}$$

2. Induction (for $1 \le t \le T-1, \quad 1 \le j \le N$):

$$\alpha_{t+1}(j) \quad = \quad \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] \frac{b_j(\mathbf{z}_j[t+1])}{b_0(\mathbf{z}_j[t+1])} \tag{4}$$

3. Termination:

$$\frac{p(\mathbf{X}[1],\ldots,\mathbf{X}[T]|\Lambda)}{p(\mathbf{X}[1],\ldots,\mathbf{X}[T]|H_0)} = \sum_{i=1}^{N} \alpha_T(i), \tag{5}$$

where H_0 is the condition that state S_0 is true at every t.

2.1.2. The class-specific backward procedure

1. Initialization:

$$\beta_T(i) = 1 \tag{6}$$

2. Induction (for $t = T-1,\ldots 1, \quad 1 \le i \le N$):

$$\beta_t(j) \quad = \quad \sum_{i=1}^{N} a_{ji} \frac{b_i(z_i[t+1])}{b_0(z_i[t+1])} \beta_{t+1}(i). \tag{7}$$

2.1.3. HMM Reestimation formulas

Define $\gamma_t(j)$ as $p(\theta_t = j|\mathbf{X}[1],\ldots,\mathbf{X}[T])$. We have

$$\gamma_t(j) = (\alpha_t(j)\, \beta_t(j)) \Big/ \left(\sum_{i=1}^{N} \alpha_t(i)\, \beta_t(i)\right). \tag{8}$$

Let

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij} \dfrac{b_j(z_j[t+1])}{b_0(z_j[t+1])} \beta_{t+1}(j)}{\displaystyle\sum_{i=1}^{N}\sum_{m=1}^{N} \alpha_t(i)\, a_{im} \frac{b_m(z_m[t+1])}{b_0(z_m[t+1])} \beta_{t+1}(m)}, \tag{9}$$

The updated state priors are $\hat{u}_i = \gamma_1(i)$. The updated state transition matrix is

$$\hat{a}_{ij} = \left(\sum_{t=1}^{T-1} \xi_t(i,j)\right) \Big/ \sum_{t=1}^{T-1} \gamma_t(i). \tag{10}$$

2.1.4. Gaussian mixture reestimation formulas

We assume the following Gaussian mixture representation for each $b_j(z_j)$:

$$b_j(z_j) = \sum_{k=1}^{M} c_{jk}\, \mathcal{N}(z_j, \boldsymbol{\mu}_{jk}, \mathbf{U}_{jk}), \quad 1 \le j \le N. \tag{11}$$

where

$$\mathcal{N}(z_j, \boldsymbol{\mu}, \mathbf{U}) \triangleq (2\pi)^{-P_j/2} |\mathbf{U}|^{-1/2} e^{\left\{-\frac{1}{2}(s-\boldsymbol{\mu})' \mathbf{U}^{-1} (s-\boldsymbol{\mu})\right\}},$$

and P_j is the dimension of z_j. We may let M be independent of j as long as M is sufficiently large. Let

$$\gamma_t(j,m) = \gamma_t(j) \left[\frac{c_{jm}\, \mathcal{N}(z_j[t], \boldsymbol{\mu}_{jm}, \mathbf{U}_{jm})}{b_j(z_j[t])}\right], \tag{12}$$

$$\hat{c}_{jm} = \left(\sum_{t=1}^{T} \gamma_t(j,m)\right) \Big/ \left(\sum_{t=1}^{T}\sum_{l=1}^{M} \gamma_t(j,l),\right) \tag{13}$$

$$\hat{\boldsymbol{\mu}}_{jm} = \left(\sum_{t=1}^{T} \gamma_t(j,m)\, z_j[t]\right) \Big/ \left(\sum_{t=1}^{T} \gamma_t(j,m)\right), \tag{14}$$

and

$$\hat{\mathbf{U}}_{jm} = \frac{\displaystyle\sum_{t=1}^{T} \gamma_t(j,m)\, (z_j[t] - \boldsymbol{\mu}_{jm})\, (z_j[t] - \boldsymbol{\mu}_{jm})'}{\displaystyle\sum_{t=1}^{T} \gamma_t(j,m)}. \tag{15}$$

2.2. Discussion

2.2.1. Relationship to conventional algorithm

The class-specific forward procedure terminates having computed the likelihood ratio (5), whereas the conventional algorithm computes only the numerator. The class-specific Baum-Welch algorithm maximizes (5) over Λ, which is equivalent to maximizing the numerator only. It is comforting to know that $\gamma_t(j)$, $\xi_t(i,j)$, $\hat{u}_i$, and $\hat{a}_{ij}$ will be identical to those estimated by the conventional approach if (2) is true. There is no correspondence, however, between the Gaussian mixture parameters of the two methods except that the densities they approximate obey (2).

3. COMPUTER SIMULATION

3.1. Simulation details

To test the Class-Specific HMM, we created a synthetic Markov model with known sufficient statistics for each state. Comparisons with the conventional approach was made by combining all the class-specific features into one common feature set. In this way, neither method has an unfair advantage with respect to features. The following state transition matrix and state priors were used:

$$\mathbf{A} = \begin{bmatrix} 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0.7 & 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ 0 & 0 & 0 & 0 & 0.7 & 0.3 \\ 0.1 & 0 & 0 & 0 & 0 & 0.9 \end{bmatrix},$$

and $u_j = 1/6$, for all j. For each t, a state is chosen randomly according to the Markov model. Then, a N-sample segment of a time series is created from the statistical model corresponding to the chosen state ($N = 256$). In our notation, we have $\mathbf{X}[t] = \{x_1[t], x_2[t], \ldots, x_N[t]\}$. Features are then computed from each segment and used as an observation vector of an HMM. The *common* state, S_0, is *iid* Gaussian noise of zero mean and unit variance, denoted N(0,1). For each model, a sufficient statistic is chosen for distinguishing the signal from H_0. These are denoted $\mathbf{z}_1$ through $\mathbf{z}_6$. Table 1 lists each signal description, the sufficient statistic, and the distribution under state S_0. The criteria for selecting these synthetic signals were (a) the signals are easy to describe and produce, (b) a sufficient or approximately sufficient statistic was known for distinguishing each state from S_0, (c) these statistics had a known density under state S_0, (d) the signals and statistics were diverse and statistically dependent.

To illustrate the strength of the Class-Specific algorithm, it was compared with two classical implementations. The classical implementations used a 7-dimensional feature vector consisting of all features used by the Class-Specific (CS) approach. Two variations of the classical approach were created by assuming the Gaussian Mixtures had (1) a general covariance matrix ("classical" or CL approach) or (2) a diagonal structure which assumes all the features are statistically independent ("independence assumption" or IA approach). We will refer to these three methods by the abbreviations CS, CL, and IA.

State 1
Description: An impulse of duration 2 samples occurring on samples 1 and 2 of the time-series with additive Gaussian noise of variance 1. **Model**: $x_t = 2(\delta[t-1] + \delta[t-2]) + n_t$, where n_t is distributed $\mathcal{N}(0,1)$. **Sufficient statistic**: $\mathbf{z}_1 = x_1 + x_2$. **log-PDF**: $\log b_0(\mathbf{z}_1) = -0.5\log(4\pi) - 0.25\mathbf{z}_1^2$.
State 2
Description: Same as state 1, but on samples 2 and 3. **Model**: $x_t = 2(\delta[t-2] + \delta[t-3]) + n_t$, where n_t is distributed $\mathcal{N}(0,1)$. **Sufficient statistic**: $\mathbf{z}_2 = x_2 + x_3$. **log-PDF**: $\log b_0(\mathbf{z}_2) = -0.5\log(4\pi) - 0.25\mathbf{z}_2^2$.
State 3
Description: Signal type 3 is *iid* Gaussian noise of zero mean and variance 1.7. **Model**: $x_t = n_t$, where n_t is distributed $\mathcal{N}(0,1.7)$. **Sufficient statistic**: $\mathbf{z}_3 = \log\left\{\sum_{t=1}^N x_t^2\right\}$. **log-PDF**: $\log b_0(\mathbf{z}_3) = -\log\Gamma(N/2) - (N/2)\log(2) + (N/2)\,\mathbf{z}_3 - \exp(\mathbf{z}_3)/2$.
State 4 and 5
Description: Signal types 4 and 5 are closely-spaced sinewawes in Gaussian noise. **Model**: $x_t = n_t + a\sin(\omega_k t + \phi)$, $k = 4,5$, where $a = 0.4$, $\omega_4 = 0.100$, and $\omega_5 = 0.101$ radians per sample. **Sufficient statistic**: $\mathbf{z}_k = \log\left\{\left|\sum_{t=1}^N x_t e^{-j\omega_k t}\right|^2\right\}$ $k = 4,5$. **log-PDF**: $\log b_0(\mathbf{z}_k) = -\log N - \frac{\exp\{\mathbf{z}_k\}}{N} + \mathbf{z}_k$ $k = 4,5$.
State 6
Description: Signal type 6 is autoregressive noise of variance 1. **Model**: $x_t = [-a_1 x_{t-1} - a_2 x_{t-2} + n_t]\ \alpha$, where n_t is distributed N(0,1), $a_1 = -0.75$, $a_2 = 0.78$, and $\alpha = 0.5675$ (α is chosen such that the variance of x_t is 1). **Approximate sufficient statistic**: The first and second lags of the normalized autocorrelation estimates computed circularly. $\mathbf{z}_6 = \{\tilde{r}_1, \tilde{r}_2\}$ where $\tilde{r}_k = \hat{r}_k/\hat{r}_0$, and $\hat{r}_k = \frac{1}{N}\sum_{i=1}^N x_i x_{(i-k) \bmod N}$. **log-PDF**: Details provided in [8].

Table 1: Description of each signal type, sufficient statistics (SS), and distribution of the SS under state S_0.

The multiple record implementation of the HMM described in Section V.(B.) of Rabiner [1] (page 273) was used. All training and testing data was organized into *records* with a constant length of 99 data segments per record ($T_r = 99$). Each segment consisted of $N = 256$ time samples. Features were computed on each segment (a segment is associated with a time-step or observation of the HMM). In the experiment, algorithm performance was determined as a function of R, the number of training records. In the experiments, we used R=1,2,5,10,20,40,80, 160, and 320 records. To measure algorithm performance of all implementations, the appropriate version of the Baum-Welch algorithm was first run to convergence on the available training data. Next, the Viterbi algorithm [2] was used to determine the most likely state sequence on a separate pool of 640 records of testing data. The number of state errors divided by the

[2]The algorithm in Rabiner [1], pp 263-264, appropriately modified for the CS method by substituting likelihood ratios $b_j(\mathbf{z}_j[t])/b_0(\mathbf{z}_j[t])$ in place of $b_j(X[t])$.

number of opportunities (640 times 99) was used as a performance metric. For more accurate determination of algorithm performance, sixteen independent trials were made at each value of R.

3.1.1. Catastrophic errors and "assisted" training

The CL and IA approaches encountered severe problems with initialization. More often than not, a poor initialization resulted in the Baum-Welch algorithm finding the incorrect stationary point. This resulted in catastrophic errors. This appeared to be caused by an inability to distinguish two or more states. The number of catastrophic errors decreased as the number of training records increased. The CS method did not encounter any catastrophic errors. To study only the PDF estimation issue apart from any initialization issues, it was decided to "assist" the CL and IA approaches by providing a good initialization point. This initialization point was found by training separately on labeled data from each of the N states, then using these PDF's as the state PDF's with the known $\mathbf{u}$ and A. In short, the algorithm was initialized with the true parameter values. This "good" parameter set was used as the starting point for all trials of the CL and IA approaches. Doing this provides somewhat of a "lower bound" on error performance. The CS results are unassisted.

3.2. Main results

A plot of median error probability for the CL, CS, and IA methods is provided in Figure 1. The IA approach has a higher limiting error rate, due to the built-in assumption of independence, which is not valid. It has, however, nearly identical convergence behavior as the CS method. This is expected since the dimension of the PDF estimates is similar for the two approaches.

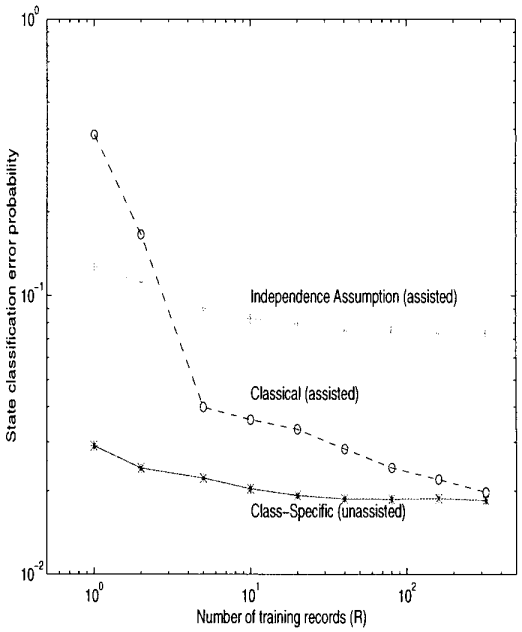

Figure 1: Probability of state classification error as a function of number of training records (R) for three methods - median performance of 16 trials. The CL and IA methods are assisted by providing a "good" initialization point.

4. CONCLUSIONS

A class-specific implementation of the Baum-Welch algorithm has been developed and verified to work in principle. A computer simulation used a 6-state HMM with six synthetic signal models. The class-specific feature dimensions were either 1 or 2. When the performance was measured in terms of state errors in the most likely state sequence, the class-specific method greatly outperformed the classic HMM approach which operated on a combined 7-dimensional feature set. The improved performance was due to the lower dimension of the feature spaces made possible by knowledge of the appropriate feature set for each state. It also outperformed a method that assumed the features were statistically independent. In this case, the improved performance comes from the fact that the class-specific approach does not compromise theoretical performance when reducing dimension.

5. 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] B. H. Juang, "Maximum likelihood estimation for mixture multivariate stochastic observations of Markov chains," *AT&T Technical Journal*, vol. 64, no. 6, pp. 1235–1249, 1985.

[3] D. W. Scott, *Multivariate Density Estimation*. Wiley, 1992.

[4] S. Aeberhard, D. Coomans, and O. de Vel, "Comparative analysis of statistical pattern recognition methods in high dimensional settings," *Pattern Recognition*, vol. 27, no. 8, pp. 1065–1077, 1994.

[5] H. Watanabe and S. Katagiri, "Discriminative subspace method for minimum error pattern recognition," in *Proc. 1995 IEEE Workshop on Neural Networks for Signal Processing*, pp. 77–86, 1995.

[6] P. J. Huber, "Projection pursuit," *Annals of Statistics*, vol. 13, no. 2, pp. 435–475, 1985.

[7] P. M. Baggenstoss, "Class-specific features in classification.," *IEEE Trans Signal Processing*, December 1999.

[8] P. M. Baggenstoss, "A modified Baum-Welch algorithm for hidden Markov models with multiple observation spaces.," *IEEE Trans Speech and Audio*, to appear 2000.