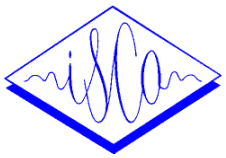


6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

ADJACENT NODE CONTINUOUS-STATE HMM'S

Carl Quillen

BBN Systems and Technologies, Cambridge MA, 02138
cquillen@bbn.com

10.21437/ICSLP.2000-105

ABSTRACT

This paper explores properties of a family of Continuous state hidden Markov models (CSHMM's) that are proposed for use in acoustic modeling. These models can be viewed as applying a smoothing to ordinary HMM's in order to make estimates of transition and observation probabilities more robust by sharing data between adjacent state nodes. They may be trained by EM so that all parameters properly reflect the applied smoothing. The amount of smoothing may be trained as well, and the model reverts to the ordinary HMM in the limit as the smoothing parameters reach zero. Thus this technique may be employed selectively only in areas in the model where training data is sparse. This paper formulates EM-training for one variant of these models, and explores their performance when applied to constructing ergodic CSHMM models of speech and to a phoneme recognition task on the same data. The ergodic CSHMM did not improve performance over the HMM, but the phoneme CSHMM's model the data with higher likelihood than the equivalent HMM's, and have superior recognition accuracy.

1. INTRODUCTION

Hidden Markov Models (HMM's) that are commonly employed in speech recognition have a state-space consisting of discrete states that are statistically independent given a fixed time t. A discrete state-space is by no means fundamental or necessary. HMM's where the state and observation probabilities are modeled by Gaussians are in widespread use, and are commonly referred to as Kalman filters. They suggest a filtering point of view, where the hidden state provides a "denoised" version of the observed audio data, and quite naturally they have been employed for speech enhancement.[3] Kalman filters are quite useful for modeling continuous trajectories, and they have been used in this capacity by a number of researchers[1],[2], but this approach usually relies on an a-priori segmentation of speech provided by another (usually HMM based) recognition system to make the search task feasible.

This paper describes and explores a family of non-parametric continuous state models that remain fairly close in spirit to traditional HMM's, and can be applied essentially without change in existing HMM based systems. These models also possess a piecewise-linear state which can be employed to model trajectories, but they can be used without the need for pre-segmenting the data, and are only moderately more expensive than standard HMM's.

2. $\alpha - \beta$ RECURSIONS FOR A CSHMM

Suppose you a have an observation sequence $\bar{x} = (x_1 \ldots x_T)$ for discrete times $t = 1 \ldots T$ (Most likely the individual x_i would be vectors). Suppose we have a state vector $\bar{s} = (s_1 \ldots s_T)$. Then using the usual notation $\alpha_t(s_t) = P(x_1 \ldots x_t, s_t)$, and $\beta_t(s_t) = P(x_{t+1} \ldots x_T | s_t)$, and using the usual Markovian dependence assumptions of the HMM,

$$P(\bar{x}, \ _t) = \alpha_t(s_t)\beta_t(s_t).$$

We can write the α and β recursions using integrals as

$$\alpha_t(s_t) = b(x_t|s_t) \int \alpha_{t-1}(s_{t-1}) P(s_t|s_{t-1}) ds_{t-1} \quad (1)$$

$$\alpha_1(s_1) = b(x_1|s_1)P(s_1) \quad (2)$$

$$\beta_{t-1}(s_t) = \int \beta_t(s_t) P(s_t|s_{t-1}) b(x_t|s_t) ds_t \quad (3)$$

$$\beta_T(s_T) = 1. \quad (4)$$

Here $b(x_t|s_t)$ denotes the observation probability of x_t. The key practical question that occurs when implementing these recursions as integrals is keeping the number of terms from growing exponentially as the recurrences progress. In the Kalman filter case usually $b(x_t|s_t)$ is a Gaussian, α and β are Gaussian mixtures, and $P(s_t|s_{t-1})$ is a Gaussian. Products of these Gaussians become sums of the exponents, and the resulting number of terms does not grow.

There is another case where the number of terms will remain finite. This occurs when the function of $f(s_t)$ described by $\int f(s_t)P(s_t|s_{t-1})\, ds_t$ for any $f(s_t) \in L^2$ is a finite-dimensional projection, or example:

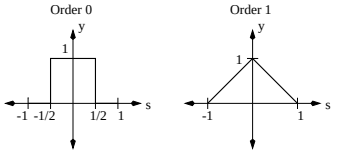
$$P(s_t|s_{t-1}) = \sum_{i,j} \frac{a_{ij}}{w_j} b_i(s_{t-1}) b_j(s_t) \quad (5)$$

$P(s_t|s_{t-1})$ is a probability distribution in s_t, and must satisfy the constraints

$$P(s_t|s_{t-1}) > 0 \text{ and } \int P(s_t|s_{t-1})\, ds_t = 1.$$

This can be achieved if $\{b_i(s)\}$is a set of positive basis functions in L^2, $w_i \doteq \int b_i(s)\, ds$, $\sum_i b_i(s) = 1$ and $\sum_j a_{ij} = 1\ \forall i$. Finding such basis functions is fairly easy. For one-dimensional state spaces, for eample the B-splines satisfy all the requirements.

Two different possible basis functions for a 1-dimensional state space are depicted below:



A basis set would be constructed by taking translates of the basic function, i.e. $b_i(s) = b_0(s - i)$. In the piecewise constant case, the basis functions have no overlap, and $P(s_t|s_{t-1})$ is exactly the same as for the discrete HMM. (The observation probability $b(x_t|s_t)$ could still be chosen to be somewhat differently, however.) The piecewise linear case corresponds to piecewise-linear interpolation between state nodes.

2.1. Centroid networks

We would like to take the filtering point of view and have a state space that corresponds to a de-noised version of the observation space. The approach taken here is to employ one-dimensional spline functions in a network in higher-dimensional observation space. This bears some explanation. Suppose you take training data and derive centroids from it by a k-means like process. Then quantize all the utterances in the training set. Each utterance will transition in time from centroid to centroid. Consider the centroids as nodes in a network, and the observed transitions as links connecting the nodes. An example is depicted in Figure 1.

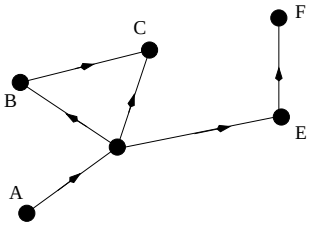


Figure 1: A network of transitions from VQ centroids

2.2. Order-1 basis on a centroid network

Figure 2a depicts the kind of basis element that would result from applying first order B-splines to a network like that in Figure 1. Basis elements here are indexed by the center node, so this basis element would be referred to as $b_D(s_t)$.

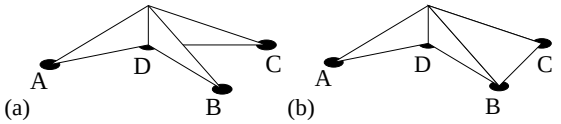


Figure 2: (a) A 1-D basis element for a network. (b) A Mixed dimensionality basis element.

The basis element just suggested only allows interactions between pairs of basis elements. This reflects the fact that the basis elements are one-dimensional. One can also have higher-dimensional elements that permit three-way or n-way interactions. The dimensionality of the resulting basis could even change from link to link. Figure 2b depicts this possibility.

In the remainder of this paper, we shall confine ourselves to piecewise-linear basis functions. Because it will be convienient to train the degree of interaction between nodes, the form of the basis function will be allowed to change from link to link. Figure 3 depicts the form functions might have on an individual link. An actual basis element would usually have support on multiple links, and would be pieced together using different sub-elements for the each link.

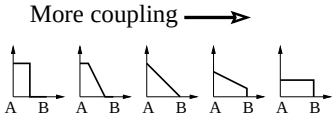


Figure 3: variable coupling between nodes A and B.

2.3. The observation distribution $b(x_t|s_t)$

The remaining question is what to employ for the observation distribution $b(x_t|s_t)$. There are many possibilities, a number of which permit practical implementation. Among the alternatives the following two suggest themselves:

1. Interpolated observation probabilities

$$b(x_t|s_t) = \sum_l b_l(s_t) N_l(x_t|\mu_l, \sigma_l). \tag{6}$$

   Where $N_l(x_t|\mu_l, \sigma_l)$ is a Gaussian with mean μ and variance σ. In this case we are just interpolating between Gaussians associated with adjacent state nodes. This alternative is explored in the remainder of this paper.

2. Trajectory observation probabilities

$$b(x_t|s_t) = N\left(x_t \left| \sum_l b_l(s_t)\mu_l, \sum_l b_l^0(s_t)\Sigma_l \right.\right). \tag{7}$$

   where $b_l^0(s_t)$ is the non-overlapping piecewise constant basis. Here the Gaussian mean depends continuously on the state and the variances vary discontinuously—they depend on the nearest node. All integrals required to carry out the $\alpha - \beta$ recursions and EM update can be carried out analytically, although the results can be quite messy. Space constraints do not permit further discussion of this model here.

It should be noted that there is no reason why the same basis elements $b_l(s_t)$ need be used for both $b(x_t|s_t)$ and $P(s_t|s_{t-1})$. One could for example have a piecewise constant elements with no overlap for $P(s_t|s_{t-1})$ and use elements of the family in figure 3 for $b(x_t|s_t)$. On can even conceive of having two different families of basis functions in $P(s_t|s_{t-1})$:

$$P(s_t|s_{t-1}) = \sum_{i,j} \frac{a_{ij}}{w_j} b_i^1(s_{t-1}) b_j^2(s_t).$$

2.4. Recurrences

We'll assume a piecewise linear basis $\{b_i\}$ used for both $b(x_t|s_t)$ and $P(s_t|s_{t-1})$ and the simple state dependence for the observation probability described in (6). The α recursion can be somewhat simplified if we define a probability γ so that $\gamma_t(s_t) = P(x_1 \ldots x_{t-1}, s_t)$, and $\alpha_t(s_t) = b(x_t|s_t)\gamma_t(s_t)$. γ satisfies a recursion very similar to the β recursion:

$$\gamma_{t+1}(s_{t+1}) = \int P(s_{t+1}|s_t)b(x_t|s_t)\gamma_t(s_t)ds_t \quad (8)$$

$$= \sum_k c_k(t+1)b_k(s_{t+1}). \quad (9)$$

The fact that $P(s_{t+1}|s_t)$ is a projection onto $\{b_k(s_{t+1})\}$ implies that we can write (8) as (9). Applying equation (6):

$$\gamma_{t+1}(s_{t+1}) = \sum_{i,j,k,l} N_l(x_t)c_k(t)\frac{a_{ij}}{w_j}b_j(s_{t+1}) < ilk >$$

Where $< ilk >$ refers to the integral

$$< ilk > \doteq \int b_i(s_t)b_l(s_t)b_k(s_t)\, ds_t. \quad (10)$$

which can be computed analytically given a particular choice of basis functions. Equation (10) specifices the amount of smoothing between nodes in the centroid network. Consider the case of piecewise linear basis functions. Suppose we consider a link between nodes A and B. Then i, l, k in (10) can each be either of A or B if this integral is to be nonzero along that link. There are eight possible sets of indices for basis functions that overlap on that link. For linear elements these eight cases reduce to two: $\int_0^1 x^3\, dx = \frac{1}{4}$ and $\int_0^1 x^2(1-x)\, dx = \frac{1}{12}$.

2.5. Generalized nearest-neighbor basis functions

It may be desirable to tune the amount of coupling between nodes desribed by equation (10). This can be acomplished by varying the form of the basis function, for example as suggested by figure 3. Confining ourselves to computing the integral along a single on-dimensional link $C_1 = \int_0^1 b(x)^3\, dx$ and $C_2 = \int_0^1 b(x)^2 b(1-x)\, dx$ characterize the only integrals that need to be computed, if we assume that the the basis function $b(x)$ satisfies $b(x) + b(1-x) = 1$. Assuming also that $b(x) \geq 0$, it then follows that $C_1 = \frac{1}{2} - 3C_2$, $1/8 \leq C_1 \leq 1/2$ and $0 \leq C_2 \leq 1/8$.

2.6. The β recursion

The β recursion is very similar to what we saw for $\gamma_t = \alpha_t / b(x_t|s_t)$.

$$\beta_{t-1}(s_t) = \int P(s_t|s_{t-1})b(x_t|s_t)\beta_t(s_t)\, ds_t \quad (11)$$

$$= \sum_k d_k(t-1)b_k(s_{t-1}). \quad (12)$$

Using (6) as before results in

$$\beta_{t-1}(s_{t-1}) = \sum_{i,j,k,l} N_l(x_t)d_k(t)\frac{a_{ij}}{w_j}b_i(s_{t-1}) < jlk > .$$

2.7. Normalization

Because the absolute likelihood of an utterance falls exponentially with the length of the utterance, α and β tend to rapidly to zero as their recurrences progress. Because of this it is easier to calculate $\hat{\alpha}$ and $\hat{\beta}$ instead of α and β, where

$\hat{\alpha}_t(s_t) = P(s_t|x_1 \ldots x_t) = \alpha_t / \int \alpha_t\, ds_t$ and $\hat{\beta}_t = \beta_t / \int \hat{\alpha}_t \beta_t\, ds_t$. We can also define $\hat{\gamma}$ in terms of $\hat{\alpha}$, so that $\hat{\gamma}_t b(x_t|s_t) = \hat{\alpha}_t$. The recurrences for $\hat{\alpha}, \hat{\beta}$ and $\hat{\gamma}$ are straightforward modifications of the ones for α, β and γ. See Minka[4] for details.

Below we'll have occasion to express $\hat{\gamma}$ and $\hat{\beta}$ in terms of their basis function expansions

$$\hat{\gamma}_t = \sum_i \hat{c}_i(t)b_i(s_t) \text{ and } \hat{\beta}_t = \sum_i \hat{d}_i(t)b_i(s_t).$$

3. EM UPDATE EQUATIONS

The proceeding recursions may be employed without difficulty in the standard EM algorithm, to optimize the model parameters. In the equations below, the form of the basis functions may vary from link to link. We assume below that each link between two nodes i and j is enumerated by a number ℓ. Expressing terms where we can by $\hat{\alpha}$ and $\hat{\beta}$, and optimizing subject to the constraints $\sum_j c_j(1) = 1$, $\sum_j a_{ij} = 1$, $0 \leq C_2(\ell) \leq 1/8$, $C_1(\ell) + 3C_2(\ell) = 1/2$ yields the following EM update equations:

$$\mu_l' = \sum_t \sum_{i,j} \hat{c}_i(t)\hat{d}_j(t)N_l(x_t) < ijl > x_t$$

$$\Sigma_l' = \sum_{t=1}^{T} \sum_{i,j} \hat{c}_i(t)\hat{d}_j(t)N_l(x_t) < ijl > (x_t - \mu_l)^2$$

$$c_i'(1) \propto \sum c_i(1)\hat{d}_i(1)N_l(x_1) < ijl >$$

$$a_{kl}' \propto \sum_{t=1}^{T-1} \frac{1}{e_t} \sum_{i,j,m,n} < kij > \hat{c}_i(t)N_j(x_t)a_{kl}$$
$$\hat{d}_m(t+1)N_n(x_{t+1}) < lmn >$$

$$C_1'(\ell) \propto \sum_{t=1}^{T} \sum_{i \in \text{nodes}(C_1(\ell))} \hat{c}_i(t)\hat{d}_i(t)N_i(x_1)C_1(\ell)$$

$$C_2'(\ell) \propto \sum_{t=1}^{T} \sum_{i,j,l \in X} \hat{c}_i(t)\hat{d}_j(t)N_l(x_t) < ijl >$$

where e_t is the normalizing constant $\hat{\alpha}_t = e_t\alpha_t$, and X is the set of $\{i, j, l\}$ such that $< ijl >= C_2(\ell)$. The constant of proportionality for c_i', a_{kl}', C_1' and C_2' is computed in each case so that the relevant constraints are satisfied.

These update equations are written out assuming one utterance, but they are normalized correctly so that the sum could continue across multiple utterances.

4. PRELIMINARY EXPERIMENTS

The theory described here was explored in two ways, first by constructing an ergodic HMMS's and CSHMM's, and then constructing phoneme modes using the ergodic HMM state probabilities as observations.

The ergodic HMM's are constructed by first carrying out k-means clustering of the training data. The k-means centroids are then used to provide Gaussians for a single-Gaussian/state ergodic HMM. The transition matrix is initialized uniformly. For the CSHMM variant, links between states are found by quantizing utterances in training by the k-means centroids. Transitions between one centroid and other are counted, and links between states created for the 5 most frequently observed transitions. Links are made symmetrical, so individual states may aquire more than 5 links.

EM training begins with two passes of training of the transition probabilities with the Gaussians held fixed. Two more stages up EM training then update all parameters in the model, and then 6 more stages are applied with just the variances held fixed. Then Gaussians with > 200 frames of training data are split, and these new Gaussians added with links to their source Gaussians. Three more passes of EM training are then used. Finally 3 passes of EM employed using held out training data to provide a final update of link weights.

Eight hours of Call-Friend Farsi data was used to train the resulting 3000-state ergodic HMM's and CSHMM's. The same data was then decoded with the ergodic models. The top-50 Gaussian likelihoods were used as observations for forward-backward training of 30 phoneme models using utterance-level segmentations. Five-state HMM's and five-node CSHMMS's were created. The phoneme CSHMM models used links between adjacent state nodes. Link-strengths were frozen until the 14th EM iteration, and then were updated until the 20th iteration.

The phoneme performance for HMM observations can be seen in figures 4 and 5. Likelihoods remain lower for the phoneme CSHMM's until the link weights are trained, at which point the weights tend to zero, and the result is effectively an ordinary HMM that performs better than a conventionally trained one. The continuous-state methodology appears to have avoided a local minimum. Even before the link weights are trained, the CSHMM has slightly better error rates than the standard HMM. This is particularly promising given that adequate training data was available for all the models trained, and that model-smoothing was't necessary. One would expect the CSHMM to help more in sparse training data situations.

Figure 6 dipicts phoneme error rates for the ergodic CSHMM case. Again, CSHMM phoneme models are better than the HMM models, but absolute performance levels are worse. Link weights in the ergodic CSHMM are very close to zero in this case, so the training has effectively produced an inferior HMM than the usual training would. One might conjecture that the procedure for picking links in ergodic training was inadequate, and that a better procedure will be necessary—for example using Viterbi decoding to find likely transitions, or high-magnitude elements of a transition matrix.

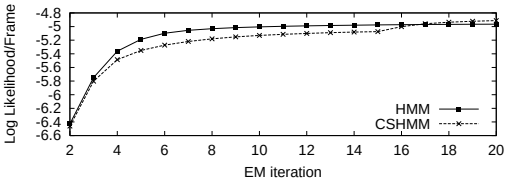


Figure 4: Log-likelihoods for phoneme models, HMM observations

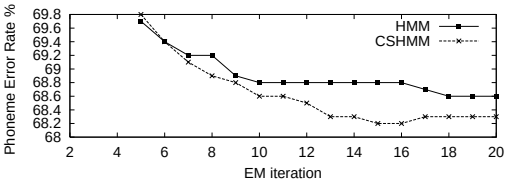


Figure 5: Phoneme error rates for HMM observations

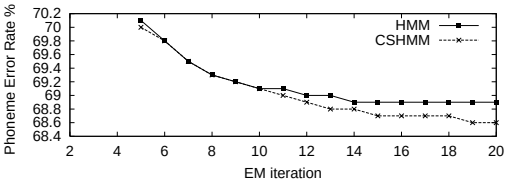


Figure 6: Phoneme error rates for CSHMM observations

5. CONCLUSION

The continuous state formalism presented provides a very general way of smoothing state statistics in an HMM. It can be viewed as accomplishing this by partially tying states together. Because the degree of tying is tunable and trainable, it can be used in a variety of interesting ways: to smooth in poorly trained parts of a model, to prevent locking on too quickly to a local minimum during early EM-iterations, and for backing-off from finer to coarser models. For example one might try tying states from a triphone model to a related biphone or monophone model with low levels of interaction to make the triphone estimates more robust. While the superiority of these models isn't conclusively demonstrated here, the attractions of this flexible theoretical framework have hopefully been made apparent.

6. REFERENCES

1. Vassilios Digalakis. Boston University Ph.D. thesis, 1992.
2. Jeff Ma and Li Deng, "Bayesian Decoding Strategy for Conversational Speech Recognition Using a Constrained Nonlinear State-Space Model for Vocal-Tract-Resonance Dynamics", IEEE Trans. Sp. Audio Processing. 1999.
3. M. Gabrea, E. Grivel, and M. Najim, "Single Microphone Kalman Filter-Based Noise Canceller", IEEE Signal Processing Letters, Vol. 6 No. 3, March 1999.
4. Thomas P. Minka, "From Hidden Markov Models to Dynamical systems", ftp://vismod.www.media.mit.edu/pub/tpminka/papers/minka-lds-tut.ps.gz, 1999