SECOND ORDER HIDDEN MARKOV MODELS FOR SPEECH RECOGNITION

Brett Watson, Ah Chung Tsoi

Department of Electrical and Computer Engineering
University of Queensland

ABSTRACT - The application of Hidden Markov Models (HMM's) to speech research has yielded some of the best performing speech and speaker recognition systems. In this paper an extension to standard Markov models, second order HMM's, which allow dependence of transition probabilities on previous states as well as on the current state, incorporating in-context information on the hidden states, is presented. Both standard Baum-Welch re-estimation and discriminative alpha-network training techniques are presented.

INTRODUCTION AND TERMINOLOGY

When applying HMM's to speech recognition it is a standard practice to make a first order Markov assumption that at each observation time, t, a new state is entered based on the transition probability, which depends only upon the previous state. Another assumption is that the output at each time step depends only upon the state at that time, regardless of when and how the state was entered. In this paper the impact of altering the first assumption, so that the state which is entered at any time step, t, is dependent on the previous two states, at $t-1$ and $t-2$, is examined. This is referred to as the second order assumption.

Each state in a standard HMM has an associated initial probability π_i, such that:

$$\pi_i = Pr(s_1 = i)$$

where s_t is the state of the model at time t.

First order transition probabilities, a_{ij}, determine the probability of a transition between states i and j:

$$a_{ij} = Pr(s_t = j | s_{t-1} = i)$$

For the second order Markov models which are developed here, these first order transition probabilities apply only to the first transition in the state sequence.

Second order transition probabilities a_{ijk}, determine the probability of a transition to state k, given that we have had a transition from state i to state j:

$$a_{ijk} = Pr(s_t = k | s_{t-1} = j, s_{t-2} = i)$$

Each state has an associated output probability function $b_j(o_t)$, which is the probability of observing o_t, when in state j at time t:

$$b_j(o_t) = Pr(o_t | s_t = j)$$

There are T observations in the complete observation sequence, which is denoted by $O = O_1^T$. The corresponding state sequence followed by the model is $S = s_1, s_2, \ldots, s_T$.

We shall refer to the complete set of parameters which characterises the HMM as $\lambda = \{\pi_i, a_{ij}, a_{ijk}, b_j\}$. Using the terminology presented so far we can calculate the probability of a state sequence, given the model, and the joint probability of the state and observation sequences, given the model, as follows.

$$
\begin{aligned}
Pr(S|\lambda) &= \pi_{s_1} a_{s_1 s_2} a_{s_1 s_2 s_3} a_{s_2 s_3 s_4} \cdots a_{s_{T-2} s_{T-1} s_T} \\
&= \pi_{s_1} a_{s_1 s_2} \prod_{t=1}^{T-2} a_{s_t s_{t+1} s_{t+2}} \\
Pr(O,S|\lambda) &= Pr(S|\lambda) \prod_{t=1}^{T} b_{s_t}(o_t)
\end{aligned}
$$

In the following sections, algorithms for training these models are developed.

SECOND ORDER BAUM-WELCH RE-ESTIMATION

The Baum-Welch re-estimation process involves taking an initial estimate for the HMM parameters, and applying update equations to the model parameters in order to increase the likelihood of the training observation given the model (Huang, Ariki & Jack, 1990). We define a function, Q, such that:

$$
Q(\lambda, \overline{\lambda}) = \frac{1}{Pr(O|\lambda)} \sum_{all\ S} Pr(O,S|\lambda) log Pr(O,S|\overline{\lambda}) \tag{1}
$$

$Q(\lambda, \overline{\lambda})$ forms an auxiliary function such that

$$
Q(\lambda, \overline{\lambda}) \geq Q(\lambda, \lambda) \Rightarrow Pr(O|\overline{\lambda}) \geq Pr(O|\lambda) \tag{2}
$$

(and the inequality is strict unless $Pr(O|\overline{\lambda}) = Pr(O|\lambda)$).

Now

$$
log Pr(O,S|\overline{\lambda}) = log \overline{\pi}_{s_1} + \log \overline{a}_{s_1 s_2} + \sum_{t=1}^{T-2} log \overline{a}_{s_t s_{t+1} s_{t+2}} + \sum_{t=1}^{T} log \overline{b}_{s_t}(o_t) \tag{3}
$$

$$
\begin{aligned}
Q(\lambda, \overline{\lambda}) &= \sum_i \frac{Pr(O, s_1 = i|\lambda)}{Pr(O|\lambda)} log \overline{\pi}_i + \\
&\sum_i \sum_j \frac{Pr(s_1 = i, s_2 = j, O|\lambda)}{Pr(O|\lambda)} log \overline{a}_{ij} + \\
&\sum_i \sum_j \sum_k \frac{\sum_{t=1}^{T-2} Pr(s_t = i, s_{t+1} = j, s_{t+2} = k, O|\lambda)}{Pr(O|\lambda)} log \overline{a}_{ijk} + \\
&\sum_i \frac{\sum_{t \in o_t = v_k} Pr(s_t = i, O|\lambda)}{Pr(O|\lambda)} log \overline{b}_j(k)
\end{aligned}
$$

(Assuming a discrete output distribution with output classes v_k.)

Therefore from (1) to maximise $Q(\lambda, \overline{\lambda})$ we must choose our re-estimates so that:

$$
\overline{\pi}_i = \frac{Pr(O, s_1 = i|\lambda)}{\sum_i Pr(O, s_1 = i|\lambda)} \tag{4}
$$

$$
\overline{a}_{ij} = \frac{Pr(s_1 = i, s_2 = j, O|\lambda)}{\sum_j Pr(s_1 = i, s_2 = j, O|\lambda)} \tag{5}
$$

$$
\overline{a}_{ijk} = \frac{\sum_{t=1}^{T-2} Pr(s_t = i, s_{t+1} = j, s_{t+2} = k, O|\lambda)}{\sum_{t=1}^{T-2} \sum_k Pr(s_t = i, s_{t+1} = j, s_{t+2} = k, O|\lambda)} \tag{6}
$$

$$\bar{b}_j(k) = \frac{\sum_{t,o_t=v_k} Pr(s_t = i, O|\lambda)}{\sum_t Pr(s_t = i, O|\lambda)} \quad (7)$$

$$(8)$$

To simplify the calculation of these re-estimates we define forward and backward probabilities, α, and β.

$$\alpha_t(i) = Pr(s_t = i, O_1^t|\lambda) \quad (9)$$
$$\alpha_t(i,j) = Pr(s_{t-1} = i, s_t = j, O_1^t|\lambda) \quad (10)$$
$$\beta_t(i) = Pr(O_{t+1}^T|s_t = i, \lambda) \quad (11)$$

These α are calculated recursively as follows:

$$\alpha_1(i) = \pi_i b_i(o_1) \quad (12)$$
$$\alpha_2(i,j) = \alpha_1(i) a_{ij} b_j(o_2) \quad (13)$$
$$\alpha_2(j) = \sum_i \alpha_1(i) a_{ij} b_j(o_2) \quad (14)$$
$$\alpha_t(j,k) = \sum_i \alpha_{t-1}(i,j) a_{ijk} b_k(o_t),\ 3 \le t \le T \quad (15)$$
$$\alpha_t(k) = \sum_j \alpha_t(j,k),\ 3 \le t \le T \quad (16)$$

While the backward probabilities, β, are approximated by using the first order transition probabilities:

$$\beta_T(i) = \begin{cases} 1, \text{ for the final state of the model} \\ 0, \text{ otherwise} \end{cases} \quad (17)$$
$$\beta_t(j) = \sum_i a_{ji} b_i(o_{t+1}) \beta_{t+1}(i) \quad (18)$$

The re-estimates for the model parameters, in terms of the forward and backward probabilities are then:

$$\bar{\pi}_i = \frac{\alpha_1(i)\beta_1(i)}{\sum_i \alpha_1(i)\beta_1(i)} \quad (19)$$
$$\bar{a}_{ij} = \frac{\alpha_1(i) a_{ij} b_j(o_2) \beta_2(j)}{\sum_j \alpha_1(i) a_{ij} b_j(o_2) \beta_2(j)} \quad (20)$$
$$\bar{a}_{ijk} = \frac{\sum_{t=1}^{T-2} \alpha_{t+1}(i,j) a_{ijk} b_k(o_{t+2}) \beta_{t+2}(k)}{\sum_{t=1}^{T-2} \sum_k \alpha_{t+1}(i,j) a_{ijk} b_k(o_{t+2}) \beta_{t+2}(k)} \quad (21)$$
$$\bar{b}_j(k) = \frac{\sum_{t=1, o_t=v_k}^{T} \alpha_t(i)\beta_t(i)}{\sum_{t=1}^{T} \alpha_t(i)\beta_t(i)} \quad (22)$$

SECOND ORDER ALPHA-NETWORK DISCRIMINATIVE TRAINING

Bridle's alpha-networks (Bridle, 1990) apply a gradient descent training procedure (which corresponds to back-propagation in time, used in neural network training) to the discriminative training of HMM's. Suppose we have a set, W, of word, or sub-word, models. Each model outputs a score, L_w for the likelihood of an observation sequence given that model. Normalising these likelihoods across all models:

$$P_w = \frac{L_w}{\sum_{v \in W} L_v} \tag{23}$$

We discriminatively train these models by minimising

$$J = -logP_c, \text{ where } c \in W \text{ is the correct class} \tag{24}$$

Minimising J maximises the mutual information, between the observation sequence and the correct model (Young, 1991).

For each parameter, ϕ, of the HMM we calculate updates using the formula:

$$\Delta\phi^{(n)} = \eta \frac{\partial J}{\partial \phi^{(n-1)}} \tag{25}$$

where n is the iteration step index, and η is the learning rate. As is common in neural network training a momentum term, ζ, may also be included:

$$\Delta\phi^{(n)} = \eta \frac{\partial J}{\partial \phi^{(n-1)}} + \zeta \Delta\phi^{(n-1)} \tag{26}$$

Therefore, to calculate updates for the first-order transition probabilities, a_{ij}, with respect to a single word class, w:

$$\frac{\partial J}{\partial a_{ij}} = \frac{\partial J}{\partial L_w} \frac{\partial L_w}{\partial a_{ij}} \tag{27}$$

Also

$$\frac{\partial J}{\partial L_w} = \frac{P_w - \delta_{cw}}{L_w} \tag{28}$$

$$\frac{\partial L_w}{\partial \alpha_t(j)} = \beta_t(j) \tag{29}$$

$$\frac{\partial L_w}{\partial a_{ij}} = \frac{\partial L_w}{\partial \alpha_2(j)} \frac{\partial \alpha_2(j)}{\partial a_{ij}} \tag{30}$$

$$= \beta_2(j)\alpha_1(i)b_j(o_2) \tag{31}$$

To enforce the constraints on the a_{ij}'s that $0 \leq a_{ij} \leq 1, \forall i,j$ and $\sum_j a_{ij} = 1, \forall i$ a set of unconstrained variables $\{A_{ij}\}$ are introduced, and we apply the transformation

$$a_{ij} = \frac{e^{A_{ij}}}{\sum_l e^{A_{il}}} \tag{32}$$

Then

$$\frac{\partial a_{il}}{\partial A_{ij}} = a_{il}(\delta_{jl} - a_{ij}) \tag{33}$$

So

$$\frac{\partial L_w}{\partial A_{ij}} = \sum_l \frac{\partial L_w}{\partial a_{il}} \frac{\partial a_{il}}{\partial A_{ij}} \tag{34}$$

$$= \sum_l \beta_2(j)\alpha_1(i)b_j(o_2)a_{il}(\delta_{jl} - a_{ij}) \tag{35}$$

$$\frac{\partial J}{\partial A_{ij}} = \frac{(P_w - \delta_{cw})}{L_w} \sum_l \beta_2(j)\alpha_1(i)b_j(o_2)a_{il}(\delta_{jl} - a_{ij}) \tag{36}$$

In a similar way for the second order transition probabilities we find

$$\frac{\partial J}{\partial a_{ijk}} = \frac{\partial J}{\partial L_w} \frac{\partial L_w}{\partial a_{ijk}} \tag{37}$$

$$\frac{\partial L_w}{\partial a_{ijk}} = \sum_{t=3}^{T} \frac{\partial L_w}{\partial \alpha_t(k)} \frac{\partial \alpha_t(k)}{\partial a_{ijk}} \tag{38}$$

$$\frac{\partial \alpha_t(k)}{\partial a_{ijk}} = \alpha_{t-1}(i,j)b_k(o_t) \tag{39}$$

$$\frac{\partial L_w}{\partial a_{ijk}} = \sum_{t=3}^{T} \beta_t(k)\alpha_{t-1}(i,j)b_k(o_t) \tag{40}$$

Transforming the a_{ijk}'s to enforce the constraints that $0 \leq a_{ijk} \leq 1, \forall i,j,k$, and $\sum_k a_{ijk} = 1, \forall k$:

$$a_{ijk} = \frac{e^{A_{ijk}}}{\sum_l e^{A_{ijl}}} \tag{41}$$

$$\frac{\partial a_{ijl}}{\partial A_{ijk}} = a_{ijl}(\delta_{kl} - a_{ijk}) \tag{42}$$

So

$$\frac{\partial L_w}{\partial A_{ijk}} = \sum_l \frac{\partial L_w}{\partial a_{ijl}} \frac{\partial a_{ijl}}{\partial A_{ijl}} \tag{43}$$

$$= \sum_l \sum_{t=3}^{T} \beta_t(k)\alpha_{t-1}(i,j)b_k(o_t)a_{ijl}(\delta_{kl} - a_{ijk}) \tag{44}$$

$$\frac{\partial J}{\partial A_{ijk}} = \frac{\partial J}{\partial L_w} \frac{\partial L_w}{\partial A_{ijk}} \tag{45}$$

$$= \frac{(P_w - \delta_{cw})}{L_w} \sum_l \sum_{t=3}^{T} \beta_t(k)\alpha_{t-1}(i,j)b_k(o_t)a_{ijl}(\delta_{kl} - a_{ijk}) \tag{46}$$

Similarly, to determine updates for the parameters of the output distributions (the means and variances of continuous distributions, for example), the partial derivatives of the parameters with respect to J must be calculated, and then the update equation can be applied.

PRACTICAL IMPLEMENTATION

The Baum-Welch training algorithm presented gives a local maximum of the likelihood function. It is important to start with good initial estimates, in order to approach the global maximum. The following procedure is proposed to achieve estimates for the second order HMM's:

1. Use a k-means procedure to achieve initial estimates for a first order Markov model for the training data.
2. Train the first order Markov model using standard Baum-Welch re-estimation.
3. Use the first order model as an initial estimate for the parameters of the second order model.
4. Train the second order HMM using the Baum-Welch training procedure presented in this paper.
5. Discriminatively train the models using the second order alpha-network training procedure.

This procedure has been applied to a plosive (/b/,/d/, /g/, /p/) recognition task, by creating a five state forward-transition only model for each phoneme. A minor increase in performance was observed. It is proposed that the technique may be most useful when applied to ergodic HMM's where there is a much greater range of possible state sequences. Plosives, and other phonemes are also frequently too short to benefit substantially from the increased history in the models.

CONCLUSIONS

The second-order HMM's developed in this paper utilise transition probabilities depending on the previous two states, rather than a single state, which is the norm for HMM's. In this way the models should be capable of employing the additional information about previous state occupations to improve recognition accuracy. The approach presented should be easily extended to higher order models.

A number of modifications to HMM's in order to give them greater modelling power have previously been proposed. In (Brown, 1987) it is suggested that it would be desirable to alter the output-independence assumption to reflect the fact that the way an observation at time $t-1$ differs from the mean of the output distribution influences the way an observation at time t differs from the output distribution for the model at time t. Incorporating such a modification in the HMM leads to squaring the number of parameters of the output distributions (Brown, 1987). The approach suggested here, however, increases the number of transition probabilities for an n-state model by n^3 for a fully ergodic model, and by considerably fewer for standard forward-transition only models, where many transition probabilities are set to zero. Since the number of states typically used in HMM's is usually significantly smaller than the number of parameters in the output distributions, the second-order HMM's should be computationally feasible in most instances.

REFERENCES

Bridle, J.S. (1990), *Alpha-Nets: A Recurrent 'Neural' Network Architecture with a Hidden Markov Model Interpretation* , Speech Communication 9, 83-92.

Brown, P.F. (1987), *The Acoustic-Modeling Problem in Automatic Speech Recognition*, Carnegie Mellon University, PhD Thesis.

Huang, X.D., Ariki, Y. & Jack, M.A. (1990), *Hidden Markov Models for Speech Recognition*, Edinburgh University Press.

Young, S.J. (1990) *Competitive Training: A Connectionist Approach to the Discriminative Training of Hidden Markov Models* , IEE Proceedings-1 Volume 138, No. 1, 61-68.