Towards a Maximum Entropy Method for Estimating HMM Parameters

Christian J. Walder, Peter J. Kootsookos and Brian C. Lovell
Intelligent Real-Time Imaging and Sensing Group,
School of Information Technology and Electrical Engineering,
The University of Queensland, Brisbane, Queensland 4072, Australia.
{walder, lovell}@itee.uq.edu.au

Abstract

Training a Hidden Markov Model (HMM) to maximise the probability of a given sequence can result in over-fitting. That is, the model represents the training sequence well, but fails to generalise. In this paper, we present a possible solution to this problem, which is to maximise a linear combination of the likelihood of the training data, and the entropy of the model. We derive the necessary equations for gradient based maximisation of this combined term. The performance of the system is then evaluated in comparison with three other algorithms, on a classification task using synthetic data. The results indicate that the method is potentially useful. The main problem with the method is the computational intractability of the entropy calculation.

1 Introduction

In recent years, the HMM has become one of the main tools for spatio-temporal pattern recognition, especially in the area of speech recognition [8]. In 1983, Levinson, Rabiner and Sondhi described a method of estimating HMM parameters from multiple training sequences in the maximum likelihood sense, via a special case of the expectation-maximisation algorithm. This method, known as the Baum-Welch algorithm, has been widely used, however it is well known that it is susceptible to the problem of “over fitting”.

Several attempts have been made to deal with the over-fitting problem. In 1998, Brand described an effective method involving maximum likelihood parameter estimation, but with the additional constraint of an “entropic prior” [1]. That is, an a priori assumption was made regarding the probability distribution of the HMM parameters themselves.

Recently, Davis et al have explored [2] the possibility of using parameter averaging, as suggested by Mackay in 1997 [6]. This method involves training a separate HMM for each training sequence, and then averaging the parameters of the resulting HMMs. The reported results indicate that the averaging method offers an improvement over the basic Baum-Welch algorithm.

This paper presents another method of HMM parameter estimation which is intended to overcome the over-fitting problem. The method herein was mentioned by Brand in 1998, with reference to combinatorial optimisation problems [1], however the approach does not appear to have been investigated for the task of HMM parameter estimation.

2 Background Theory

2.1 HMM Preliminaries and Notation

An HMM can be described as a probabilistic function of a Markov Chain. For the case of HMMs with discrete outputs and discrete states, we can assume that the underlying Markov Chain has N states, $q_1, q_2, \ldots, q_N$. Such a Markov chain can be specified in terms of an initial state distribution vector, $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$, and a state transition probability matrix, $A = [a_{ij}], 1 \leq i, j \leq N$. Here, π_i is the probability of q_i at time time $t = 0$, and a_{ij} is the probability of transiting to state q_j given that the current state is q_i, that is $a_{ij} = p(q_j \text{ at time t + 1 } | q_i \text{ at time t})$. In the previous expression and the remainder of the paper, $p(x)$ is to be taken as the probability of occurrence of event x.

Each of the Markov states have an associated random process which provides a probabilistic mapping to the output of the HMM, which is drawn from an alphabet V of M possible outputs, $v_1, v_2, \ldots, v_M$. These probabilistic mappings from hidden state to observed output can be collectively specified by another stochastic matrix $B = [b_{jk}]$ (the “observer matrix”) in which for $1 \leq j \leq N$ and $1 \leq k \leq M$, b_{jk} is the probability of observing symbol v_k given that the current state is q_j, that is, $b_{jk} = p(v_k \text{ at time t } | q_j \text{ at time t})$.

2.2 Entropy of a Random Variable

Consider a random variable X, with N discrete outcomes, $x_1, x_2, \ldots, x_N$. The "information" of outcome x_i is [4]:

$$I(x_i) \triangleq -\log p(X = x_i)$$

The entropy of X is the expected information [4]:

$$H(X) \triangleq -\sum_{i=1}^{N} p(X = x_i) \log(p(X = x_i))$$

2.3 Entropy of an HMM

From the equation of Section 2.2, the entropy of a sequence of length T produced by HMM λ can be written as:

$$H(\lambda, T) = -\sum_{\forall O \in \tilde{O}_T} p(O|\lambda) \log p(O|\lambda) \qquad (1)$$

Where $\tilde{O}_T$ is the set of all sequences of length T that can be produced by λ. For a symbol alphabet size M, $|\tilde{O}_T| = M^T$, so the computation in equation 1 is intractable for large T.

2.4 Maximum Entropy Parameter Estimation

Let X be a random variable taking on values $x_1, x_2, \ldots, x_K$, with an unknown probability mass function (*pmf*), $p_k = p(X = x_k)$. Suppose we would like to estimate the *pmf* of X given only the expected value of some function $g(X)$ of X:

$$\sum_{k=1}^{K} g(x_k) p_k = c \qquad (2)$$

For example if $g(X) = X$ then c is the mean of X. Since Equation 2 does not, in general, specify the *pmf* of X uniquely, we must apply further constraints in order to solve for the p_k. One additional constraint that is commonly applied [4, 3] is that of "maximum entropy". That is, we seek the *pmf* that maximises the entropy subject to the constraint in Equation 2. This is intuitively appealing, since the maximum entropy solution is that which satisfies our known constraints, while asserting as little as possible about the nature of the underlying *pmf*. The maximum entropy parameter estimation can be set up as an optimisation problem and in some cases solved using classical methods such as Lagrange multipliers [4, 3].

3 Maximum Entropy HMM Parameter Estimation

In its most fundamental form, HMM parameter estimation proceeds as follows [5]. First of all, the source which is to be modelled is sampled one or more times, to provide "training data" for the parameter estimation. An HMM topology is then chosen, and an HMM is initialised randomly within the chosen topology. The HMM parameters are then adjusted so as to maximise the likelihood of the HMM producing the training sequence(s). This is known as "maximum likelihood" parameter estimation. For the case of HMMs, maximum likelihood parameter estimation is in itself a difficult problem: in general only locally optimal solutions can been found. A well known problem with the maximum likelihood approach is that of "overfitting" to the training data. That is, the model fits the training sequences *too* well, thereby failing to generalise.

In 1998, Brand proposed a means of dealing with the overfitting problem [1]. The method is essentially Bayesian inference with an "entropic prior". That is, maximum a posteriori (MAP) parameter estimation using an a priori distribution over parameter space. Formally, the method seeks the parameter set θ which maximises the posterior

$$p_e(\theta|x) \propto p(x|\theta) p_e(\theta) \qquad (3)$$

where x is the observed (training) data, and $p_e(\theta)$ is the entropic prior:

$$p_e(\theta) \propto e^{-H(\theta)} \qquad (4)$$

where $H(\theta)$ is the entropy of the model. A detailed explanation of the method can be found in [1].

The method proposed in this paper is similar to that of Brand [1], in that the generality of the model (as measured by its entropy) is accounted for during the training process, however the knowledge of model entropy is used in in a different way. Following the same approach as the classical maximum entropy method, we would like the model to have high entropy as well as to match our knowledge of the data. This leads to the following idea: rather than maximising the probability of the training sequences, maximise a linear combination of the likelihood and the model entropy. Formally, we seek to maximise the following "objective function":

$$C = b \log p(O|\lambda) + (1 - b) H(\lambda, T) \qquad (5)$$

Where $b \in [0, 1]$, the "balancing parameter", is the free parameter that sets the desired "generality" of the model, and O is our training sequence. For example, $b = 1$ results in normal maximum likelihood learning, whereas $b = 0$ ignores the training data and maximises the entropy of the model.

In equation 5, $\log p(O|\lambda)$ is maximised rather than $p(O|\lambda)$ to ensure that we are comparing equivalent units of likelihood and entropy – log of probability is information, and entropy is expected information, so the units are comparable. If the $\log$ is to base 2, then the units are "bits", while a natural log has units "nats".

In the next section we begin describing how the model can be optimised according to the objective function above. Before proceeding, however, it is worth making a few comments regarding the b parameter of Equation 5. The inclusion of the parameter can be justified by the following argument. In an extremely "data poor" training problem in which we have only one training sequence, it may be possible to find a deterministic (zero entropy) HMM that fits the data perfectly in the maximum likelihood sense, however this would obviously be of no value for either regression or classification tasks. By applying domain knowledge, it may be possible to sensibly choose b such that a useful model is obtained. The necessity for the free parameter is a symptom of the inherent difficulties of all inductive inference tasks – as is well known, logical induction is a flawed process, and one that requires the assumption of some prior knowledge in order to reach a conclusion [7].

4 Gradient Descent Equations

To maximise the objective function in equation 5, we can perform gradient descent w.r.t. C. To do this we need the partial derivative of C w.r.t. an arbitrary HMM parameter, θ. This follows directly from equation 5:

$$\frac{\partial C}{\partial \theta} = \frac{b}{p(O|\lambda)}\frac{\partial p(O|\lambda)}{\partial \theta} + (1-b)\frac{\partial H(\lambda, T)}{\partial \theta}$$

We now proceed, in a top-down approach, to relate the above expression back to all of the specific HMM model parameters.

4.1 Partial Derivatives of HMM Entropy with respect to Model Parameters

Taking the partial derivatives of equation 1 with respect to an arbitrary parameter θ we get,

$$\frac{\partial H(\lambda, T)}{\partial \theta} = -\sum_{\forall O \in \tilde{O}_T} \frac{\partial p(O|\lambda)}{\partial \theta}(1 + \log p(O|\lambda)) \quad (6)$$

Next we need the expressions for $\frac{\partial p(O|\lambda)}{\partial \theta}$.

4.2 Partial Derivatives of Likelihood with respect to HMM Parameters

This is our own derivation of the partial derivatives of likelihood with respect to HMM parameters. An alternative derivation is available in [5]. From [5] we have the probability in terms of the forward variable, $\alpha_t(n)$:

$$p(O|\lambda) = \sum_{n=1}^{N} \alpha_T(n) \quad (7)$$

$$\alpha_{t+1}(n) = \sum_{m=1}^{N} \alpha_t(m) a_{mn} b_n(O_{t+1}) \quad (8)$$

$$\alpha_1(n) = \pi_n b_n(O_1) \quad (9)$$

Where O_t is the t-th observation symbol in our training sequence, O. From equation 7 we get:

$$\frac{\partial p(O|\lambda)}{\partial a_{ij}} = \sum_{n=1}^{N} \frac{\partial \alpha_T(n)}{\partial a_{ij}}$$

$$\frac{\partial p(O|\lambda)}{\partial b_j(k)} = \sum_{n=1}^{N} \frac{\partial \alpha_T(n)}{\partial b_j(k)}$$

$$\frac{\partial p(O|\lambda)}{\partial \pi_i} = \sum_{n=1}^{N} \frac{\partial \alpha_T(n)}{\partial \pi_i}$$

From equations 8 and 9 we get, for a_{ij}:

$$\frac{\partial \alpha_{t+1}(n)}{\partial a_{ij}} = \alpha_t(i) b_j(O_{t+1}) + \sum_{m=1}^{N} a_{mn} b_n(O_{t+1}) \frac{\partial \alpha_t(m)}{\partial a_{ij}}$$

$$\frac{\partial \alpha_1(n)}{\partial a_{ij}} = 0$$

for $b_j(k)$,

$$\frac{\partial \alpha_{t+1}(n)}{\partial b_j(k)} = \sum_{m=1}^{N} (\delta(v_k, O_{t+1})\delta(j,n) a_{mn} \alpha_t(m) +$$

$$b_n(O_{t+1}) a_{mn} \frac{\partial \alpha_t(m)}{\partial b_j(k)})$$

$$\frac{\partial \alpha_1(n)}{\partial b_j(k)} = \pi_n \delta(O_1, v_k) \delta(j, n)$$

and for π_i:

$$\frac{\partial \alpha_{t+1}(n)}{\partial \pi_i} = \sum_{m=1}^{N} a_{mn} b_n(O_{t+1}) \frac{\partial \alpha_t(m)}{\partial \pi_i}$$

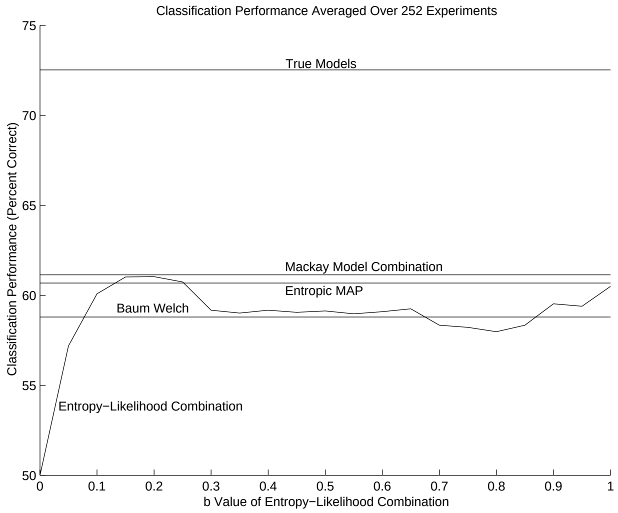

Figure 1. Mean classification performance for various model pairs.

$$\frac{\partial \alpha_1(n)}{\partial \pi_i} = b_i(O_1)\delta(i,n)$$

In the above, $\delta(x,y)$ is the Kronecker delta function:

$$\delta(x,y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The equations above provide a recursive means of computing the partial derivatives of probability, w.r.t. model parameters, for a given observation sequence. The time complexity of the calculation is linear w.r.t. T. This and the results of section 4.1 allow us to calculate $\frac{\partial H(\lambda,T)}{\partial \theta}$ for all parameters θ, ie. a_{ij}, $b_j(k)$ and π_i. Unfortunately the complexity is then exponential w.r.t. T, that is, the operation has time complexity of order $O(M^T)$.

Since we now have the partial derivatives of both entropy and likelihood with respect to all of the HMM parameters, we can calculate $\frac{\partial C}{\partial \theta}$ using Equation 4, and so we can maximise C using standard hill climbing/gradient descent based numerical optimisation.

5 Results

The performance of the method has been tested in a classification task with the following methodology. An HMM topology of two hidden states and two observation symbols was chosen, with a "feed forward" structure (upper triangular in the transition matrix). Two HMMs were then randomly initialised subject to the topology and structure constraints above, and a bias was placed on the long diagonal of the transition matrix by choosing uniformly random (in the range $[0, 1]$) transition probabilities, then adding 3 to the long diagonal and normalising each *pmf* to satisfy the stochasticity constraint. Structure was also added to the observer matrix by similarly biasing a single randomly chosen probability from the observation *pmf* of each state. From each of these two "true" or "generating" models, 5 training and 500 testing sequences of length 4 were randomly generated. The sequences were chosen to be so short, and the number of states so few, due to the exponential time complexity the entropy calculation (see Section 2.3).

The training set of each model was then used to estimate an HMM with the same topology as that of the initial models, using the following training algorithms: Baum-Welch maximum likelihood [5], Mackay model averaging [2], Brand's entropic prior [1], and finally the maximum entropy method presented in this paper. For the maximum entropy method, the b value of Equation 5 was varied from 0 to 1 with a step size of 0.05. Following this, the test set was then classified by all of the pairs of learnt models, and also by the "true" models. This entire procedure was repeated 252 times with different random seeds. The mean classification performance over the 252 trials is shown for each model pair in Figure 1.

6 Discussion

Before considering the information presented in Figure 1, it should first be noted that the amount of data used to construct the curves is somewhat insufficient. For example, the sign test shows that the hypothesis "Mackay Model Combination is no better or worse than Brand's entropic MAP" is correct to the significance level $p \leq 0.796$, however the true models are significantly better than all others, and the entropic MAP and Mackay model averaging methods are better than Baum Welch at the 90% significance level. With these considerations of statistical significance in mind, we proceed to discuss those features Figure 1 that are likely to be significant.

The first thing to notice is that there are indeed improvements to be made over the Baum Welch algorithm. Next, we see that the "Entropy-Likelihood Combination" method is no better or worse than random for $b = 0$ – this is to be expected since $b = 0$ corresponds to pure entropy maximisation, which gives an HMM equivalent to the independent sampling of a random variable with uniform *pmf*. As b increases, so does the performance of the maximum entropy models, until $b = 0.2$. This may seem to be a surprisingly small value for optimum b, but this is partly due to the fact that in our implementation of the training method, the log-likelihood term of Equation 4 is in fact the sum of the log-likelihood for each of the five training sequences used in the test. This results in the likelihood term effectively being increased in magnitude by a factor of five. Our conjecture, however, is that the optimal value for b is a function of the entropy of the generating HMM (or more generally, the entropy of the generating source, which in most practical applications will not be an HMM). If this conjecture is correct, then it may well be possible to determine the correct value of b for a given application, based on the statistics of known sequences.

The main problem with the algorithm in its current form is the computational intractability of the entropy calculation. Unfortunately, it is unclear whether an efficient calculation exists. It may be possible to use an ad-hoc function that is similar to entropy, however it is unclear whether this will be effective. To illustrate some of the difficulties, imagine that our ad-hoc "approximation" of $H(\lambda)$ is $H_A(\lambda) + H_B(\lambda)$, where $H_A(\lambda)$ is the entropy of the states given the transition matrix and initial state *pmf*, and $H_B(\lambda)$ is the sum of the entropies of the observer matrix *pmf*s. Now consider the pathological case in which all of the observer *pmf*s have zero entropy except one, then by varying only the transition matrix, the maximum entropy HMM is obtained when the transition matrix always transitions (with probability 1) to the state with the non-zero observation *pmf* – that is, when $H_A(\lambda) = 0$! Nonetheless, there may exist a function that performs well, for example the "variance" of an HMM as defined in [9].

7 Future Work

Some possibilities for the continuation of the work are the following:

- Attempt to find an efficient calculation for $H(\lambda)$. Failing that, prove the hardness of the problem.
- Examine the performance of the system using various easily calculated ad-hoc alternatives to $H(\lambda)$.
- Investigate the relationship between the optimal b value (of Equation 5) and the entropy of the generating source.

References

[1] Matthew Brand. Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation*, 11(5):1155–1182, 1999.

[2] Richard I. A. Davis, Brian C. Lovell, and Terry Caelli. Improved estimation of hidden markov model parameters from multiple observation sequences. *Proceedings International Conference on Pattern Recognition*, pages 168–171, August 2002.

[3] E.T. Jaynes. *Papers on Probability, Statistics and Statistical Physics*. Kluwer Academic Publishers, 1989.

[4] Alberto Leon-Garcia. *Probability and Random Processes for Electrical Engineering*. Addison Wesley, 2nd edition, May 1994.

[5] S.E. Levinson, L.R. Rabiner, and M.M. Sondhi. An introducton to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *Bell System Technical Journal*, 62:1035–1074, April 1983.

[6] D.J.C. Mackay. Ensemble learning for hidden markov models. *Technical Report, Cavendish Laboratory, University of Cambridge*, 1997.

[7] K.R. Popper. *The Logic of Scientific Discovery*. Hutchinson, 1968.

[8] L.R. Rabiner and B.H.Juang. *Fundamentals of Speech Recognition*. Prentice Hall, New Jersey, 1993.

[9] Roy L. Streit. The moments of matched and mismatched hidden markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(4), April 1990.