Estimating the Entropy of Hidden Markov Models

Erik Louie
elouie@andrew.cmu.edu

Abstract

Hidden Markov Models (HMMs) are widely used for discrete time series and unsupervised learning. The Baum-Welch has long since been the widely used algorithm to estimate the underlying HMM of an observed sequence. Recent research has shown that spectral methods are applicable to estimating the HMM efficiently. In this paper, we consider using the entropy of solution sequences to a given observed sequence as a measure of estimator efficiency. We survey methods of estimating and calculating entropy assuming HMMs. We then examine empirical results on their effectiveness.

1 Introduction

Hidden Markov Models (HMMs) are widely used for discrete time series and unsupervised learning. They are commonly used for text and speech recognition, genome modeling, and image processing. The predominant heuristic for estimating an HMM is the Baum-Welch algorithm, which uses an iterative EM algorithm to find local maxima [2]. More recently, efficient spectral learning algorithms have been proposed for learning HMMs.

Showing the effectiveness of algorithms has been a hard problem. For instance, research has shown that the HMM learning problem is not PAC-learnable in polynomial time [8]. Despite this, recent research has continued work on finding global bounds for risk, consistency, and entropy rates by placing reasonable assumptions on the data set, assumed model, or the algorithm used.

In this paper, we explore the use of entropy as a method of evaluating the effectiveness of an HMM learning algorithm. Entropy is a measure of the uncertainty of a random variable and mutual information is a measure of the divergence between two random variables [3]. We will calculate the entropy of the possible state sequence solutions for a given algorithm. Since we consider all the paths through the HMM states, we get an evaluation of the estimated HMM as a whole. To achieve this goal, we first survey the two primary methods of estimating an HMM. We then show an efficient method of calculating the entropy of state sequences for a given observation sequence in 3.2. As a brief aside, we show that the entropy can be computed efficiently during the Baum-Welch algorithm in 3.3. We finish off by comparing the entropy of the two algorithms in terms of the length of an observed sequence, the number of states, and number of observation symbols.

2 Hidden Markov Models

In this paper, we consider a hidden markov model to be a discrete-time homogenous Markov chain observed through a discrete-time memoryless invariate channel, such as that defined by [4], and specifically consider the case where there are finite states and observation symbols. Formally, a hidden Markov model is a 5-tuple (S, V, A, B, π) where

- $S = S_1, \ldots, S_N$ a set of N hidden states, which often represent physical aspects of a model such as coins in a model with coin tosses or word occurrence in a text model, where the state at time t is represented by q_t,

- $V = v_1, \ldots, v_M$ the set of M observable symbols, such as heads or tails or a block of bits representing words,
- $A = a_{ij}$ an $N \times N$ state probability transition matrix with transition probabilities $a_{ij} = P(q_t = S_i | q_{t-1} = S_j) \forall i, j \in 1, \ldots, N$,
- $B = b_j(k)$ distributions over the observed symbols where $b_j(k) = P(v_k | q_t = S_j)$, and
- $\pi = \pi_i$ the initial state probability distribution where the probability of each state is $\pi_i = P(q_1 = S_i)$.

We will assume S and V are known and use the simpler notation, $\lambda = (A, B, \pi)$, suggested by Rabiner [7].

3 Estimating entropy of an HMM

In this work, we are concerned with HMM where the states of the markov chain are unknown. Our interest is to estimate the entropy of the solution state sequence that correspond to a sequence of observed values assuming they are drawn according to an HMM. To estimate the entropy under HMM assumptions, we will first estimate the number of states, known as the order of the HMM, followed by estimating the parameter states, and then calculating the entropy for the HMM. For the initial estimation, we will use the well-known Baum-Welch algorithm to estimate the states and observable probability densities.

3.1 Estimating Parameters of the HMM

We will compare two algorithms performance in estimating the parameters. The first is the widely used Baum-Welch algorithm, as described by Rabiner, which is essentially an EM algorithm [7]. The second, recently discovered, algorithm is a variant on the spectral method, as described by Hsu et al [6].

3.1.1 The Baum-Welch Algorithm

The algorithm is a simple iterative expectation-maximization heuristic that utilizes a forward-backward procedure to calculate the variables necessary to use Bayes theorem to maximize the expected transitions and expected probability of seeing an observation.

At each iteration, we will need to estimate a new HMM from the existing one. We first calculate two variables: a forward variable representing the probability of seeing a sequence of states until time t and the current observed symbol

$$\begin{aligned} \alpha_t(i) &= P(O_1 O_2 \ldots O_t, q_t = S_i | \lambda) \\ &= b_i(O_t) \sum_{j=1}^{N} \alpha_t(j) a_{ji}, \end{aligned}$$

where the last formula is calculated from the HMM from the previous iteration and $\alpha_1(i) = \pi_i b_i(O_1)$, and a backward variable representing the probability of seeing a sequence of states after time t and the current observed symbol

$$\begin{aligned} \beta_t(i) &= P(O_t O_{t+1} \ldots O_T | q_t = S_i, \lambda) \\ &= \sum_{j=1}^{N} \beta_{t+1}(j) b_j(O_t) \end{aligned}$$

where $\beta_T = 1$. Given these definitions, we can calculate the update variables needed for the update step of the iteration. We need two variables: $\xi_t(i, j)$, the probability of of transitioning from state S_i at time t to state S_j,

$$\begin{aligned} \xi_t(i, j) &= \frac{P(q_t = S_i, q_{t+1} = S_j, O | \lambda)}{P(O | \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} \beta_{t+1}(j)} \end{aligned}$$

and the probability of being in state S_i at time t is just $\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)$. Given these, we can now run the update step to create $\bar{\lambda}$:

$$\bar{\pi}_i = \gamma_i$$
$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$
$$\bar{b}_j(k) = \frac{\sum_{t=1, O_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

We then repeat until $\lambda = \bar{\lambda}$, repeating with $\bar{\lambda}$ otherwise.

3.1.2 Spectral Method for Learning an HMM

In the spectral method, we parameterize a model by $\hat{b}_1, \hat{b}_\infty$, and $\hat{B}_x \forall x \in [T]$, with a T-sized sample. As defined by Hsu et al., the algorithm has three steps:

1. Create empirical estimates $\hat{P}_1, \hat{P}_{2,1}, \hat{P}_{3,x,1} \forall x \in [T]$, where P_1 is the probability of a single observation, $P_{2,1}$ is the joint probability of seeing two observations in sequence, and $P_{3,x,1}$ is the probability of seeing three observations in sequence with the second being x.

2. Compute $\hat{U}$, the left singular vectors corresponding to the m largest singular values of the singular value decomposition of $\hat{P}_{2,1}$.

3. Compute the model parameters:
$$\hat{b}_1 = \hat{U}^T \hat{P}_1$$
$$\hat{b}_1 = (\hat{P}_{2,1}\hat{U})^+ \hat{P}_1$$
$$\hat{b}_x = \hat{U}^T \hat{P}_{3,x,1} (\hat{U}^T \hat{P}_{2,1})^+ \hat{P}_1$$

From his paper, he shows that we can retrieve the observation matrix by calculating the eigenvalues of $(U^T P_{3,x,1})(U_T P_3, 1)^+$, which correspond to the observation probabilities $O_{r,1}, \ldots, O_{r,m}$ for observation r. We retrieve O from the diagonal of the above observation probabilities. We can then get the initial transition matrix $\hat{\pi} = O^+ P_1$ and transition matrix $T = O^+ P_{2,1} (O^+)^T diag(\hat{\pi})^{-1}$. Thus, we obtain the HMM parameters from the spectral method.

3.2 Calculating the Entropy of an HMM

Hernando et al. have constructed an efficient algorithm to compute the entropy of the state parameters given a complete HMM for a specific sequence of observations [5]. This algorithm uses a dynamic programming approach similar to the Viterbi algorithm for finding the most probabilistic path of an HMM.

Let $q^t = q_1, q_2, \ldots, q_t$ be the sequence of states up to the t^{th} time state, $O^t = O_1, O_2, \ldots, O_t$ be the sequence of observed values up to the t^{th} time state, $c_t(j) = P(q_t = S_j | O^t, \lambda)$, and $H_t(j) = H(S^{t-1} | S_t = j, O^t = o^t)$. Hernando et al. have shown that the following recursive algorithm finds the entropy in $O(N^2 T)$ time:

1. Initialization, $\forall 1 \leq j \leq N$:

$$H_1(j) = 0$$
$$c_1(j) = \frac{\pi(j) b_j(O_1)}{\sum_{i=1}^{N} \pi(i) b_i(O_1)}$$

2. Recursive step, $\forall 1 \leq j \leq N, \forall 2 \leq t \leq T$:

$$c_t(j) = \frac{\sum_{i=1}^{N} c_{t-1} a_{ij} b_j(O_t)}{\sum_{k=1}^{N} \sum_{i=1}^{N} c_{t-1} a_{ik} b_j(O_t)}$$

$$P(q_{t-1} = S_i | q_t = S_j, O^t) = \frac{a_{ij} c_{t-1} i}{\sum_{k=1}^{N} a_{ik} c_{t-1} i}$$

$$H_t(j) = \sum_{i=1}^{N} (i) P(S_{t-1} = i | q_t = S_j, O^t) - \sum_{i=1}^{N} [P(q_{t-1} = S_i | q_t = S_j, O^t) \cdot \log P(q_{t-1} = S_i | q_t = S_j, O^t)]$$

3. Termination:

$$H(q^T | O^T) = \sum_{i=1}^{N} H_T(i) c_T(i) - \sum_{i=1}^{N} c_T(i) \log c_T(i)$$

3.3 Efficiently calculating entropy during the Baum-Welch algorithm

Notice from the above calculations that $c_t(j)$ is the probability of the t^{th} state being S_j, which is $\gamma_t(j)$ from the Baum-Welch algorithm. Thus, we can apply the above calculation of entropy by first calculating the probability of the $t-1^{th}$ step being in state S_j using $\xi_t(i,j)$ and $\gamma_{t+1}(j)$:

$$P(q_{t-1} = S_i | q_t = S_j, O^t) = \frac{xi_t(i,j)}{\gamma_{t+1}(j)}$$

So, to calculate the entropy given the observation sequence during each iteration, we need only add an $O(NT)$ steps to calculate the entropy:

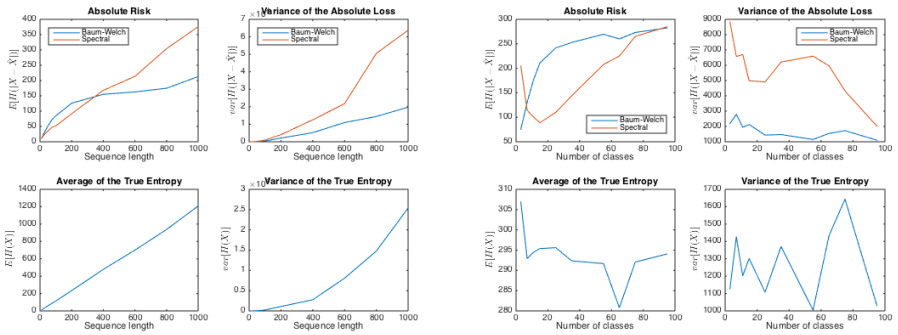
$$H_t(j) = \sum_{i=1}^{N} (i) \frac{\xi_t(i,j)}{\gamma_{t+1}(j)} - \sum_{i=1}^{N} [\frac{\xi_t(i,j)}{\gamma_{t+1}(j)} \cdot \log \frac{\xi_t(i,j)}{\gamma_{t+1}(j)}]$$

3.4 Empirical Results of Absolute Risk of the Entropy

Here, we show empirical results of the two methods of creating HMMs. We consider the risk of the absolute loss function of the entropy of the solution states between a given actual hidden markov model and the estimated markov model. We make a few simplifications in this procedure:

- The generating HMM has uniformly random transitions between states and uniformly random multinomial observation parameters.
- The number of states is known to the estimators.
- The sequence length is substantially larger than the number of states or observed symbols.
- The Baum-Welch algorithm gives up after a reasonable, finite number of steps. In this case, fifty.

We used the following procedure for retrieving the average estimated entropy:



(a) Variation over observation sequence length (b) Variation over observation symbol length

Figure 1: Empirical entropy estimation risk

1. Select an HMM HMM_{true} at random.
2. Create a sequence s from HMM_{true}.
3. Create a random guess for the HMM H_{guess}.
4. Feed H_{guess} into the Baum-Welch algorithm to retrieve HMM_{BM}.
5. Create $length(s) - 2$ triples from s, s_{triple}.
6. Feed s_{triple} into the spectral algorithm to retrieve HMM_{spec}.
7. Retrieve the entropy $H_{true}, H_{BM}, and H_{spec}$ from each of the HMMs.
8. Calculate the absolute difference.

We then iterate over different values of each of the length of an observed sequence, number of states in the HMM, and number of observed symbols in the HMM, keeping the others fixed.

Notice in both 1a and 1b that the Baum-Welch algorithm risk is logarithmic in the size of the parameter. However, In both cases, the spectral method produces a risk that is approximately linear in the parameters, while initially having lower risk. Thus, despite the fact that the Baum-Welch algorithm is known to find local maxima, it achieves a lower bound in the long term, empirically. However, Baum-Welch is computationally more complex than the spectral method and quickly increases in computation time in the number of states. So, while the spectral method has a moderate risk, it gains in time complexity.

4 Future Work

The results of this paper show the effectiveness of estimators with empirical results. In their paper, Hsu et al. show a global bound on the absolute risk of sequences of observed symbols using the spectral algorithm. This bound requires an assumption on the distance of mixture models. Recent work on EM theory has shown global bounds on the absolute risk of parameters given good initial guesses and the maximization function in the Baum-Welch algorithm is contractive [1].

References

[1] Sivaraman Balakrishnan, Martin J Wainwright, and Bin Yu. Statistical guarantees for the em algorithm: From population to sample-based analysis. *arXiv preprint arXiv:1408.2156*, 2014.

[2] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Statist.*, 41(1):164–171, 02 1970.

[3] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[4] Yariv Ephraim and Neri Merhav. Hidden markov processes. *Information Theory, IEEE Transactions on*, 48(6):1518–1569, 2002.

[5] Diego Hernando, Valentino Crespi, and George Cybenko. Efficient computation of the hidden markov model entropy for a given observation sequence. *Information Theory, IEEE Transactions on*, 51(7):2681–2685, 2005.

[6] Daniel Hsu, Sham M Kakade, and Tong Zhang. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.

[7] Lawrence Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[8] Sebastiaan A Terwijn. On the learnability of hidden markov models. In *Grammatical Inference: Algorithms and Applications*, pages 261–268. Springer, 2002.