Hidden Markov Models

An Approach for Estimating the Unknown

Gustav Hedemark, August Heegaard, Laust Rask & Magnus Wied

6th of March 2024

Introduction

Ever since their formulation in the 1960s, hidden Markov models (HMMs) have been implemented in a broad variety of fields to great success. As the name indicates, HMMs are an alteration of the classical Markov model, where some parameters or distributions are estimated, although the underlying states cannot be directly observed. Consequently, these states are called "hidden" states and may only be estimated by using observed values. An early application and use of a HMM was within the field of speech recognition.[1] HMMs have also been found to be of great use within bioinformatics, where they have contributed significantly to the protein structure problem. As will be discussed later, Bystroff et al. used a HMM with great success by correlating the local sequence of a protein with the structure.[2]
In the following sections, a theoretical background for HMMs will be provided first, along with some of the most crucial algorithms. Thereafter, the application of HMMs by Bystroff et al. for protein structure prediction is presented, followed by a short conclusion.

Theory

The main difference when moving from a classical Markov chain to a HMM is that all states are not observable. The underlying stochastic processes are hidden but can be inferred from another set of processes that produce a sequence of observations $\mathbf{Q} = \{q_1, q_2, ..., q_T\}$ where T is the number of time-steps.[1] HMMs consist of N hidden states $\mathbf{S} = \{s_1, s_2, ..., s_N\}$ each of which has an initial probability $\mathbf{\Pi} = \{\pi_1, \pi_2, ..., \pi_N\}$ and K observables $\mathbf{O} = \{o_1, o_2, ..., o_K\}$. Between all states a State Transition Matrix $\mathbf{A}$ of size $N \times N$ is constructed, stating the probability of going from one state to a subsequent state given a time-step t. Between each state and each observable, a State Emission Matrix $\mathbf{B}$ of size $N \times K$ is constructed, this describes the one-way transition probability between the current state and the specific observable. Both matrices are seen in Figure 1 which also shows a schematic of the build-up of the model.

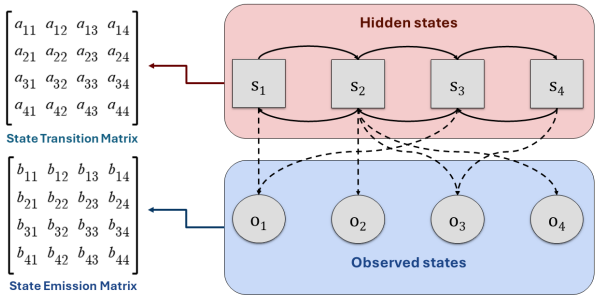

Figure 1: Overview of a HMM, where solid lines represent transition probabilities and dashed lines represent emission probabilities.

Baum-Welch Algorithm

A natural question to ask is how one can infer the values of the transition, emission and initialisation matrices for a HMM. The Baum-Welch algorithm finds a local maximum likelihood using the iterative Expectation-Maximisation Algorithm from some initial parameters.[3] The first step is calculating the expectation values of the model given an observed sequence ($\mathbf{Q}$) and the current model parameters (λ). This is done using the Forward and Backward Algorithms. The Forward Algorithm calculates the probability ($\alpha_i(t)$) of seeing the observed sequence (up to a point t) and being in state i at that time, given the model parameters. The Backward Algorithm calculates the probability ($\beta_i(t)$) of seeing the rest of the sequence (from a point t) to the end of the sequence, given that you start in state i and the model parameters. The values $\gamma_i(t)$ is the expected value of being in a specific state at a time t and $\xi_t(i,j)$ is the expected value of transitioning between two states at a time t.

$$\gamma_i(t) = P(q_t = i|q, \lambda) = \frac{\alpha_i(t)\beta_i(t)}{\sum_j^N \alpha_j(t)\beta_j(t)} \tag{1}$$

$$\begin{aligned} \xi_i j(t) &= P(q_t = i, q_{t+1} = j|Q, \lambda) \\ &= \frac{\alpha_i(t)a_{ij}b_j(q_{t+1})\beta_j(t+1)}{\sum_{k=1}^N \sum_{w=1}^N \alpha_k(t)a_{kw}b_w(q_{t+1})\beta_k(t+1)} \end{aligned} \tag{2}$$

With the expected values (γ and ξ), the maximisation step is used to update the model parameters. As $\gamma_i(t)$ is simply the expected fraction in state i, the initial parameters are updated to equal $\gamma_i(t=1)$. The transmission probabilities are updated based on ξ and γ:

$$a_{ij} = \frac{\sum_t^{T-1} \xi_i j(t)}{\sum_t^{T-1} \gamma_i(t)} \tag{3}$$

Dividing by γ_t normalizes the probabilities. Lastly, the emission probabilities are updated based on $\gamma_i(t)$ and the observed sequence ($\mathbf{Q}$):

$$b_i(o_j) = \frac{\sum_t^{T} \gamma_i(t) \cdot 1(q_t = o_j)}{\sum_t^{T} \gamma_i(t)} \tag{4}$$

Here $1(q_t = o_j)$ represents a conditional function that is 1 if the condition is filled, and otherwise 0. Repeating this algorithm has been proven to find a local maximum likelihood for the model parameters.[3]

Viterbi Algorithm

The Viterbi algorithm describes the most likely sequence of states $\mathbf{X} = \{x_1, x_2, .., x_T\}$ through the hidden Markov chain given a specific observation sequence ($\mathbf{Q}$). It is done by utilising Bayesian statistics and finding the most likely path for each iteration using the previous state as the prior. To do this, one has to keep track of which argument maximises the most likely path. This is done by storing these values in an array ψ.

$$x_i(t) = \max_k(x_k(t-1) \cdot a_{k,i} \cdot b_i(q_t)) \tag{5}$$

$$\psi_i(t) = \arg\max_k(x_k(t-1) \cdot a_{k,i} \cdot b_i(q_t)) \tag{6}$$

$x_i(t)$ is the most likely hidden state i at time t and $\psi_i(t)$ is the most likely state at time $t-1$. After T time-steps one has the final state x_T of the most likely sequence $\mathbf{X}$, and by iterating backwards using the ψ array, the most likely sequence can be found.

$$x_T = \max_k(x_k(T)) \tag{7}$$

$$t = T, T-1, ..., 2: \quad x_{t-1} = \psi_{x_t}(t) \tag{8}$$

Here, the iterative process of this algorithm becomes apparent, with the need for storing ψ and then going backwards in order to get the most probable sequence.[4]

Application

As mentioned previously, HMMs have applications within the field of protein science, for example in protein folding prediction, as shown by Bystroff et al.[2]. In the paper the HMM, abbreviated as HMMSTR, seeks to construct protein structures from the local sequence of amino acids. The presented HMM is a branched merge of multiple Markov chains, where each of these subchains corresponds to a specific protein fragment, i.e. a short amino acid sequence with a well-defined structure, allowing it to associate certain structures with specific amino acid sequences. By carefully merging the chains, a greater HMM is obtained, where each hidden state path corresponds to a defined structure. Each hidden state on a path produces four values that describe the probability of observing a particular amino acid, the angle of the backbone, the structure, and the structural context, respectively. With this, provided a new amino acid sequence, possible local protein structures and angles may be estimated. Rather than utilising the Viterbi algorithm, HMMSTR uses a voting procedure, whereby the local structure at an amino acid is given by the greatest sum of probabilities for a certain structure. Although HMMSTR was at its inception a powerful tool, it was still limited by its discretisation of angles and structures, whereby information is lost. As such, alternative HMMs with continuous outputs have been developed since, as shown in the TorusDBN model, where angles are represented as points on a torus.[5] Others have done away with HMMs in preference for entirely different approaches, such as employing deep Markov models instead.[6]

Conclusion

We have briefly introduced the concept and theory of HMMs and two algorithms to determine the parameters of hidden states in a HMM and the most likely hidden state sequence (the Baum-Welch and Viterbi Algorithms). Furthermore, we presented a case where a HMM was implemented for predicting the protein structure from an amino acid sequence, showing the applicability of this method.

References

(1) L. Rabiner, *Proceedings of the IEEE*, 1989, **77**, 257–286.

(2) C. Bystroff, V. Thorsson and D. Baker, *Journal of Molecular Biology*, 2000, **301**, 173–190.

(3) L. E. Baum, T. Petrie, G. Soules and N. Weiss, *The Annals of Mathematical Statistics*, 1970, **41**, 164–171.

(4) A. Viterbi, *IEEE Transactions on Information Theory*, 1967, **13**, 260–269.

(5) W. Boomsma, K. V. Mardia, C. C. Taylor, J. Ferkinghoff-Borg, A. Krogh and T. Hamelryck, *Proceedings of the National Academy of Sciences*, 2008, **105**, 8932–8937.

(6) C. B. Thygesen, A. S. Al-Sibahi, C. S. Steenmanns, L. S. Moreta, A. B. Sørensen and T. Hamelryck, 2021, DOI: 10.1101/2021.06.22.449406.