Hidden Markov Models:
Theory, Implementation, and Extensions

Kyle Bradbury

December 17, 2007

1 Introduction

The ability to model phenomena that are encountered is a vital part of the scientific community as a whole. Many phenomena are not directly observable, however often some result or effect of that phenomena provide a means by which it may be analyzed and measured. For example, it would be difficult to know conclusively if a person is angry, however, a raised voice, frowning facial features, and increases heart rate may all indicate the underlying reality. When such phenomena occur in a sequentially then often a natural model for such a situation is a Hidden Markov Model (HMM).

In order to better understand the model and how it may be applied, first consider an illustrative example. In a 2002 medical study, [1], it was found that decreased exposure to sunlight, such as in the wintertime, has a negative effect on a person's mood, and may contribute to depression during the months of least sunlight, a condition known as seasonal affective disorder (SAD). Now consider a hypothetical male test subject, for a new study in which significant simplifying assumptions will be applied for illustrative purposes. This subject can experience three levels of sunlight: (1) high, (2) medium, and (3) low exposure. Each day the subject must report his mood rated on a scale of 1 to 9, where 1 is depressed and 9 is happy.

The mood of the subject is known, however the amount of sunlight exposure is unknown. At the beginning of the study, the subject filled out a survey claiming that he is typically happier when he experiences large quantities of sunlight, and similarly he is depressed when there are low quantities of sunlight. However, he also stated that when there is an intermediate amount of sunlight, his mood varies more with situations in his life, and there is more variability. With this prior knowledge, those conducting the study can draw inferences about this future actions. However, since there are always those events in life which can affect a person's mood drastically (such as a birth or a death in a family), the mood may not always correspond directly to the amount of sunlight exposure.

This situation is an ideal scenario for applying an HMM. There is an underlying state, which the amount of sunlight the subject receives. This quantity is unknown to the investigators of the study, so instead they use the information they asked him for: his mood; this is referred to as the observation. A "simulation" of the subject's mood is shown in figure 1, and it can be seen that although, for the most part, the observation agrees with the subjects entrance survey, there are a few surprises, such as the spike in mood while he's experiencing low exposure to sunlight. This leads to uncertainty in the model, but this too can be accounted for by incorporating an observation probability: given that he is in a certain state, an associated probability of being in a certain mood is estimated, which can be inferred based on his past experiences. Another way to account for uncertainty in the overall situation is by determining the likelihood of experiencing low sun exposure or medium sun exposure given that he is currently experiencing high sun exposure. This is a measure of the state transition probability. Similarly, there must be a certain probability that the first state he'll be in is low, medium, or high exposure, this is the initial state probability.

The mathematics behind these concepts were first conceived in the late 1960's through the work of Baum et al. in a series of articles beginning with [2]. Rabiner made that work more accessible through his 1989 tutorial, [3], wherein the theory and implementation of HMM's were eloquently presented, with a particular focus on applications to automatic speech recognition. Since then a multitude of publications have explored the varied applications of HMMs.

The basic HMM has discrete states transition probabilities, and discrete observation probabilities. This concept is readily extended to allow for continuous observation probabilities, but this process requires considering the numerical stability of the process.

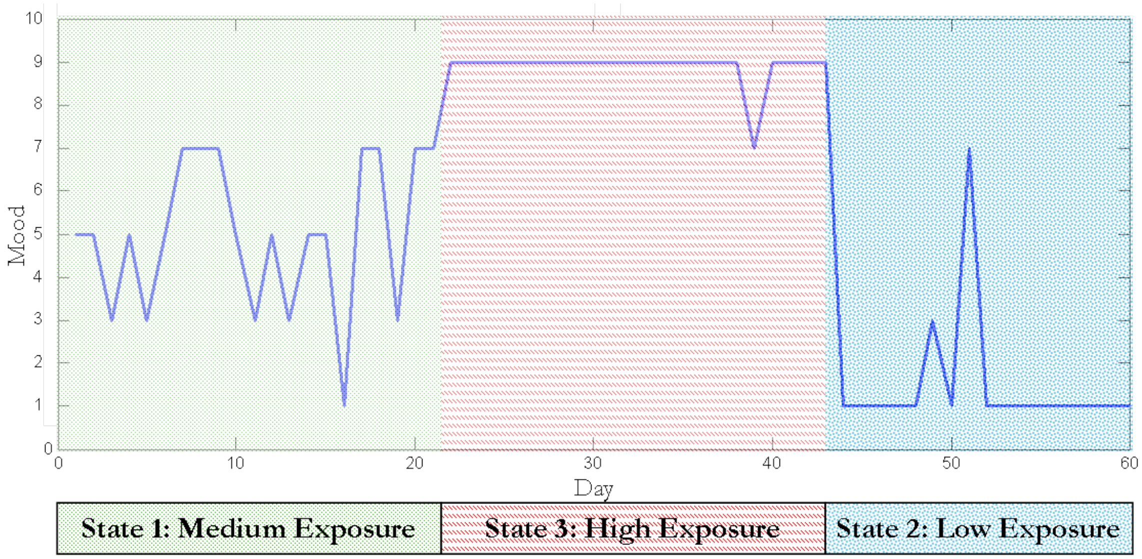

Figure 1: Example of the observation sequence of a simple HMM

These models can be extended to incorporate multiple observation sequences, for increased accuracy of initial state probabilities and state transition probabilities. Lastly, the entire HMM framework can be reworked in order to allow for explicit state duration densities. But before these more involved topics are addressed the basic model will be discussed in detail.

2 The Basic Model

Understanding the notation behind an HMM is critical to understanding its implementation and uses. The following section and the appendix are meant to provide insight into the nomenclature. For consistency, the notation used in [3] will be adopted.

2.1 Notational Note

In it's simplest form, an HMM is composed of a set of unobservable states, each of which has an initial state probability, state transition probabilities, and an observation probability density. State i is denoted S_i, where $i = 1, 2, \ldots N$, and N is the total number of states (work, the mountains, and the grocery store). An observation sequence (such as that the example shown in figure 1) of length T is represented by $O = \{O_1, O_2 \ldots O_T\}$ where the O_t's are individual observations. Each observation may be a scalar as in the example, or a vector. Each observation can take on a discrete number of observation symbol values, ν_k's, where ν_k can be a quantitative numerical value (20, 40, 60, 80, etc.,), or a qualitative description (slow, medium, fast, etc.). At a given time, t, the true underlying state is referred to as q_t, which can be any state S_i. A complete listing of notation can be found in the appendix.

2.2 The Model Explained

The model, λ, is composed of three components $\{A, B, \pi\}$. A is a matrix of state transition probabilities $\{a_{ij}\}$, which is the probability of being in state i at the current time step, t, and state j at next time step, $t+1$, or mathematically, this is $P(q_{t+1} = S_j | q_t = S_i | \lambda)$. An example involving Ian could be the probability that Ian will be heading to work at the next time step, given that he's currently heading to the mountains. B is the observation probability density, which is the probability of making a particular observation, O_t, given the current state is state i. This is notated as $b_i(O_t)$, which is represented as $P(O_t | q_t = S_i, \lambda)$. Following the example this could be the probability of Ian driving $O_t = 50$ mph, given that he was heading to the grocery store. The final component of the basic model is the vector π, which is vector of length N, each component, π_i, representing the probability of the first state being state i, $P(q_1 = S_i | \lambda)$.

3 Learning from the Model

With the model laid out, what can be learned by applying it? In [3], three problems are presented that can be addressed by HMMs. First, given the model, what is the probability of observing a specific observation sequence, $P(O|\lambda)$? Second, given a model and an observation sequence, what is the most probable underlying state sequence? Third, how can the parameters of the model be chosen in order to maximize $P(O|\lambda)$.

3.1 Problem 1: $P(O|\lambda)$?

The simplest way of determining this probability is to determine all possible combinations of states and observations, incorporate the known probabilities and thereby calculate the probability. However, that technique requires an intractable integral over all states and possible observations. In [3] it was stated that the computation of $P(O|\lambda)$ using that method would require $2TN^T$ calculations. Since observation sequences of even moderate length would produce a computational nightmare, a more efficient technique is necessary, and such an algorithm was presented in [4] and [5], and is known as the Forward-Backward algorithm. There are two basic quantities in this algorithm, the first is the probability of observing the partial observation sequence $O_1O_2\dots O_t$ and being in state i at time t, which is known as the forward variable, $\alpha_t(i)$, defined as:

$$\alpha_t(i) \equiv P(O_1O_2\dots O_t|q_t = S_i, \lambda) \tag{1}$$

and is calculated through the following procedure from [3]:

Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N \tag{2}$$

Induction:

$$\begin{aligned}\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j)a_{ij}\right] b_j(O_{t+1}),\\ 1 \le t \le T-1\\ 1 \le j \le N\end{aligned} \tag{3}$$

Termination:

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_T(i) \tag{4}$$

The above solution to this problem only requires the forward variable α, however, in order to determine the optimal state sequence, $Q = q_1q_2\dots q_T$, given the observation sequence, O, the backward variable, β, will also be needed.

$$\beta_t(i) \equiv P(O_{t+1}O_{t+2}\dots O_T|q_t = S_i, \lambda) \tag{5}$$

which is calculated through the following procedure from [3]:

Initialization:

$$\beta_T(i) = 1, \qquad 1 \le i \le N \tag{6}$$

Induction:

$$\begin{aligned}\beta_t(i) = \sum_{j=1}^{N} a_{ij}b_j(O_{t+1})\beta_{t+1}(j),\\ t = T-1, T-2, \dots, 1\\ 1 \le i \le N\end{aligned} \tag{7}$$

Approachable derivations of the recursion relations for α and β can be found in [6].

3.2 Problem 2: "Best" State Sequence?

If there is a known observation sequence from which one hopes to determine the sequence of states that "best" explains the observation, there are a few meaningful ways that this can be done. As suggested in [3], one could choose to maximize the probability of the observation given the state, $P(O_t|q_t)$, at each time t, however this only solves the problem locally. The most common optimality criterion that is chosen is to find the sequence of states, $Q^* = \{q_1^* q_2^* \cdots q_1^T\}$, that is most probable given the observation sequence, or mathematically:

$$Q^* \equiv \underset{Q}{\text{argmax}}\{P(O|Q, \lambda)\} \tag{8}$$

The method for obtaining this solution is known as the Viterbi Algorithm. First presented in [7] as a method for decoding convolutional codes, it was further analyzed in [8], and summarized again in [3]. In order to present this algorithm, a new variable, $\delta_t(i)$ must be introduced, which is the partial sequence of states ending at time t in state $q_t = S_i$, that has the highest probability given the model:

$$\delta_t(i) \equiv \max_{q_1q_2\dots q_t-1} [P(q_1q_2\dots q_t = i, O_1O_2\dots O_t|\lambda] \tag{9}$$

With (9) and a state storage variable, $\psi_t(i)$, which will keep track of the state which, at time t, yields the maximum value of $\delta_t(i)$. The algorithm is as follows:

Initialization:

$$\delta_1(j) = \pi_j b_j(O_1), \quad 1 \le i \le N \tag{10}$$

$$\psi_i(j) = 0 \tag{11}$$

Recursion:

$$\delta_t(j) = \max_{1\leq i\leq N}[\delta_{t-1}(j)a_{ij}]b_j(O_t), \qquad 2 \leq t \leq T \quad 1 \leq j \leq N \tag{12}$$

$$\psi_t(j) = \operatorname*{argmax}_{1\leq i\leq N}[\delta_{t-1}(j)a_{ij}], \qquad 2 \leq t \leq T, \quad 1 \leq j \leq N \tag{13}$$

Termination:

$$P^* = \max_{1\leq i\leq N}[\delta_T(i)] \tag{14}$$

$$q_T^* = \operatorname*{argmax}_{1\leq i\leq N}[\delta_{t-1}(i)] \tag{15}$$

Determine the State Sequence:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1 \tag{16}$$

After applying this algorithm, $Q^* = q_1^* q_2^* \ldots q_T^*$, is the optimal state sequence according to the criteria of 9.

3.3 Problem 2: The "Best" Model?

A critically important process in the application of an HMM is to estimate model parameters based on previously sampled data. In terms of classification, this is the training process. Essentially, if only an observation sequence were provided and no other information about the process, how can the model parameters be estimated? Mathematically, this is a constrained optimization problem where one wishes to find the model, λ, that maximizes the likelihood of the observation sequence given the model $P(O|\lambda)$. This problem is constrained in the sense that all probability densities that are updated must integrate (or in this case, sum) to one. This is a problem is elegantly solved by the Baum-Welch Algorithm. First presented by Baum in a seminal paper in 1970, [9], this algorithm has been applied to an array of optimization problems. One special case of this algorithm is the Expectation Maximization, or EM, algorithm. In order to apply this process to the HMM parameter reestimation process, a few new quantities must be presented.

First, $\gamma_t(i)$ is defined as the probability of being in state S_i at time t given the observation sequence and the model, (17). From the definition of conditional probability, this can be written as (18).

$$\gamma_t(i) = P(q_t = S_i|O, \lambda) \tag{17}$$

$$= \frac{P(q_t = S_i, O|\lambda)}{P(O|\lambda)} \tag{18}$$

The numerator of (18) can be rewritten in terms of the forward and backward variables, and the the denominator is merely the sum over all N states.

$$P(q_t = S_i, O|\lambda) = P(O_1 \ldots O_t, q_t = S_i|\lambda) \cdot P(O_{t+1} \ldots O_T|q_t = S_i, \lambda) \tag{19}$$

$$= \alpha_t(i)\beta_t(i) \tag{20}$$

After these manipulations, one can arrive at an implementable definition that depends only on α and β:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{t} \alpha_t(j)\beta_t(j)} \tag{21}$$

Along with γ, it is desirable to have a quantity which represents the probability of transitioning from state i to state j at any time t. This quantity will be referred to as $\xi_t(i, j)$, and is enumerated in (22):

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j|O, \lambda) \tag{22}$$

$$= \frac{P(q_t = S_i, q_{t+1} = S_j, O|\lambda)}{P(O|\lambda)} \tag{23}$$

Which can be written in terms of the forward and backwards variables,

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\displaystyle\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)} \tag{24}$$

These two quantities are the key to the model parameter reestimation process. Since $\gamma_t(i)$ represents the probability of being in state i at time t, it also represents the probability of transitioning from state i at time $t+1$. Therefore, if $\gamma_t(i)$ were summed from t to $T-1$ (since there is no transition at time T), this would effectively yield the expected number of transitions from state i. Similarly, $\xi_t(i, j)$, if summed over the same interval, will estimate the expected number of transitions from state i to state j. Therefore, as shown in [3]:

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected transitions from } S_i \tag{25}$$

$$\sum_{t=1}^{T-1} \xi_t(i, j) = \text{expected transitions from } S_i \text{ to } S_j \tag{26}$$

From these quantities, the reestimation formulas can be determined, and will be the same as the equations from Baum-Welch. π_i is probability of being in state i at time $t = 1$, or the expected number of times in S_i at time $t = 1$, and is reestimated through (27).

$$\overline{\pi}_i = \gamma_1(i) \tag{27}$$

The state transition probabilities, a_{ij} are simply the ratio of the expected number of transitions from state i to state j and the expected number of transitions from state i, as shown in (28).

$$\overline{a}_{ij} = \frac{\displaystyle\sum_{t=1}^{T-1} \xi_t(i,j)}{\displaystyle\sum_{t=1}^{T-1} \gamma_t(i)} \tag{28}$$

Lastly, the state observation density, $b_i(O_t)$, is estimated as the number of times one observes a value ν_k while in state i divided by the number of time in state i, as presented in (29).

$$\overline{b_i}(\nu_k) = \frac{\displaystyle\sum_{t \text{ s.t. } O_t = \nu_k} \gamma_t(i)}{\displaystyle\sum_{t=1}^{T-1} \gamma_t(i)} \tag{29}$$

Through the iterative reestimation of the model parameters using the above Baum-Welch update equations, it was shown in [5] that the likelihood of an observation sequence given the model is monotonically nondecreasing,

$$P(O|\overline{\lambda}) \geq P(O|\lambda) \tag{30}$$

So increased iterations of the reestimation process will only increase the likelihood, however, overtraining the parameters to a specific observation sequence is also always a possibility so caution should be exercised when using a large number of iterations.

4 Implementation

Now that the necessary formulations for a complete HMM framework have been presented, it is important to discuss some of the pitfalls of implementation. Consider again the example discussed in section 1. A parameter set for the HMM has been estimated for that situation which includes 3 states and 5 possible speeds, and the estimates are identical to the actual values. Now, using this model, observation sequences were generated of increasing length, and using each observation sequence the corresponding likelihood of the observation sequence given the model, $P(O|\lambda)$, is computed. Figure 2 shows the results of this example when the size of the observation sequence ranges from 1 to 5000.

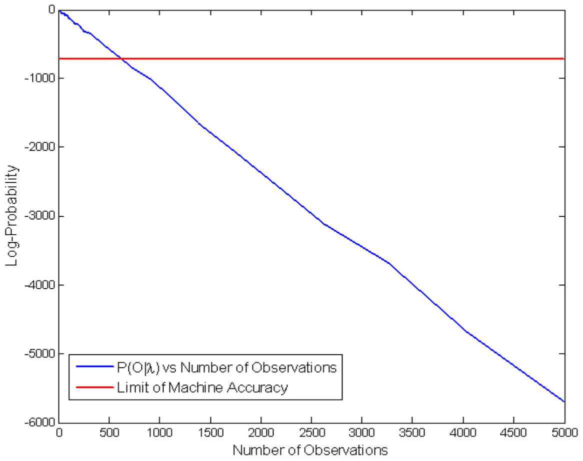

Figure 2: Numerical Instability of an HMM

The y-axis in this plot represents the log-likelihood of the observation, and the horizontal line denotes the limit of machine accuracy. Therefore, for this relatively simple scenario, with few states and a perfect estimate of the model parameters, observations of length greater than 600 will fall beyond machine accuracy, and will be set to zero. For more complex scenarios, machine accuracy could be reached much more quickly, so clearly careful consideration has to be made when implementing this framework.

There are two prominent approaches to this implementation problem. The first is the method of scaling, introduced in [10] and explained in detail in [3]. This method effectively alleviates problems associated with numerical stability by creating scaled versions of the forward and backward variables by introducing scaling coefficients which normalize the probabilities to a safe computation range, without sacrificing accuracy. However, this procedure is rather involved and requires significant changes to the parameter reestimation equations. Another technique to achieve a numerically stable implementation, presented in [11], is based on performing all computations in log-space. This means that addition and multiplication must be performed in such a way that only the logarithm of the value, not the value itself, is used. Mann suggests in [11] to create four functions to perform these calculations, each of which will be able to handle the value log(0) (reffered to herein as LOGZERO). Equation (31), below, is the only function that will ever take as an argument the actual non-logarithmic probabilities.

$$\mathrm{eln}(x) = \begin{cases} \ln(x), & x \neq 0 \\ \mathrm{LOGZERO}, & x = 0 \end{cases} \tag{31}$$

$$\mathrm{eexp}(x) = \begin{cases} \mathrm{e}^x, & x \neq 0 \\ 0, & x = 0 \end{cases} \tag{32}$$

The log-summation and log-product functions, equations (33) and (34), respectively, take only log-values as arguments instead of the values themselves. This prevents the use of the actual probabilities.

$$\begin{aligned} &\mathrm{elnsum}[\mathrm{eln}(x),\mathrm{eln}(y)] \\ &\quad = \begin{cases} \mathrm{eln}(x+y), & x \neq 0, \quad y \neq 0 \\ \mathrm{eln}(x), & y = 0 \\ \mathrm{eln}(y), & x = 0 \end{cases} \end{aligned} \tag{33}$$

$$\begin{aligned} &\mathrm{elnproduct}[\mathrm{eln}(x),\mathrm{eln}(y)] \\ &\quad = \begin{cases} \mathrm{eln}(x) + \mathrm{eln}(y), & x \neq 0, \quad y \neq 0 \\ \mathrm{LOGZERO}, & y = 0 \text{ or } x = 0 \end{cases} \end{aligned} \tag{34}$$

Now, the main idea that allows (33) to work is a convenient identity for logarithms, presented in [11], and is shown below in (35):

$$\begin{aligned} \ln(x+y) &= \ln(x+y) + \ln(x) - \ln(x) \\ &= \ln(x) + \ln(x+y) - \ln(x) \\ &= \ln(x) + \ln\left[\frac{x+y}{x}\right] \\ &= \ln(x) + \ln(1 + \frac{y}{x}) \\ &= \ln(x) + \ln(1 + e^{\ln(\frac{y}{x})}) \\ &= \ln(x) + \ln\left[1 + e^{\ln(y) - \ln(x)}\right] \end{aligned} \tag{35}$$

As long as x is larger than y, then this expression will remain within machine accuracy. Consider an example where $x = e^{100}$ and $y = e^{-100}$. In this case the results is $100 + \ln(1 + e^{-200})$ which will return 100, since e^{-200} is so small. However, if $y = e^{100}$ and $x = e^{-100}$, then the result will be $-100 + \ln(1 + e^{200})$, and e^{200} will overflow to infinity. Therefore, (33) should be implemented using the logarithmic relationship of (35).

The major benefit of this approach is that there is no need to adjust or expand any of the HMM theory, it merely is a tool for the ease of programming.

5 Extensions

With the theoretical foundation laid and implementational concerns under control, the basic discrete model can be extended in different ways which may be more appropriate on an application-by-application basis. The most notable extensions include:

- Replace discrete observation densities with continuous observation densities
- Incorporate multiple observation sequences into a single analysis
- Designate explicit state duration densities

5.1 Continuous Observation Densities

Although the discrete model is convenient because it is relatively simple to implement and apply, it is often a poor model of a continuous process. There is, however, an extension of the discrete model that allows for continuous observation densities. This concept was first introduced in [12], then extended to multivariate mixtures in [13], and summarized in [3]. The idea is to use a multivariate mixture model, such as a Gaussian mixture model, to estimate the continuous observations densities for each state. Essentially this requires summing a number of weighted Gaussian distributions to estimate the actual probability distribution. Such a mixture model, if allowed to contain a large number of terms, will be able to estimate any density function within any desired accuracy. However, using a mixture model instead of a discrete density requires some changes to the model. The new expression for the observation density is shown in equation (36).

Let $\mathbf{O} = [\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, \ldots, \mathbf{O}^{(K)}]$ represent a set of observation sequences, each of which is composed of a set of observations, where $\mathbf{O}^{(k)} = [O_1^{(k)} O_2^{(k)} \cdots O_{T_k}^{(k)}]$ is the kth observation sequence, and is of length T_k.

$$b_i(\mathbf{O}) = \sum_{m=1}^{M} c_{im} \mathfrak{N}(\mathbf{O}, \mu_{im}, \sigma_{im}), \quad 1 \leq i \leq N \tag{36}$$

As one can see from (36), a observation density made up of a mixture model requires three additional components for each state, $1, 2, \ldots, N$ and each mixture component, $1, 2, \ldots, M$. These components are: a set of mixture coefficients, c_{im},

$$\bar{c}_{im} = \frac{\displaystyle\sum_{t=1}^{T} \gamma_t(i,m)}{\displaystyle\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(i,k)} \tag{37}$$

mean vectors, μ_{im},

$$\overline{\mu}_{im} = \frac{\sum_{t=1}^{T} \gamma_t(i,m) \cdot \mathbf{O}_t}{\sum_{t=1}^{T} \gamma_t(i,m)} \tag{38}$$

and covariance matrices, σ_{im}

$$\overline{\sigma}_{im} = \frac{\sum_{t=1}^{T} \gamma_t(i,m) \cdot (\mathbf{O}_t - \mu_{im})(\mathbf{O}_t - \mu_{jm})'}{\sum_{t=1}^{T} \gamma_t(i,m)} \tag{39}$$

The reestimation formulas from [13],[3] are shown above in (37) to (39). In this framework the the reestimation formula for the probability of being in state i at time t, equation (21), also has to be adjusted to take the continuous densities into account:

$$\gamma_t(i,m) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \cdot \frac{c_{im}\mathfrak{N}(\mathbf{O}_t, \mu_{im}, \sigma_{im})}{\sum_{k=1}^{M} c_{ik}\mathfrak{N}(\mathbf{O}_t, \mu_{ik}, \sigma_{ik})} \tag{40}$$

The reestimation formulas for the state transition probabilities and the initial state probabilities, remain unchanged. Now the model can be written as $\lambda = A, \pi, c, \mu, \sigma$. This extension has been extremely useful in speech recognition and landmine detection, to name a few examples.

5.2 Multiple Observation Sequences

Another concern that presents itself with HMMs is how to accurately estimate the state transition and initial state probabilities. First, consider the initial state probabilities. A single observation sequence provides one realization of the Markov process. That implies that regardless of the underlying initial state probabilities, in that one realization there was only one initial state. Since there was only one observation sequence, the reestimation process can only find one nonzero initial state probability. Similarly, in reestimating the state transition probabilities, if a single observation sequence is used, then unless that sequence is well-representative of the data as a whole, the estimated state transition probabilities may be biased and possibly might not even have observations from all the actual states that are a part of the underlying process. A logical solution to this problem is to use multiple observation sequences to estimate the model parameters.

Combining multiple observations into a single meaningful analysis, [10] and [3] presented a new maximization goal for the Baum-Welch updates,

$$P(O|\lambda) = \prod_{k=1}^{K} P(\mathbf{O}^{(k)}|\lambda) = \prod_{k=1}^{K} P_k \tag{41}$$

Using (41), the formula for the state transition probabilities can be derived as:

$$\overline{a}_{ij} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) a_{ij} b_j(O_{t+1}^{(k)}) \beta_{t+1}^k(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(j)} \tag{42}$$

Essentially this is an averaging process and the update equations for the other model parameters are adjusted in the same way, by summing over the product of the inverse of the likelihood of each observation and the numerator and the denominator, as in (42).

5.3 Explicit State Duration Densities

Another question can be asked: what is the probability of remaining in a given state for a certain duration, for example, n time steps. This is the probability of self transitioning $n-1$ tines, and not self-transitioning once:

$$a_{ii}^{n-1}(1 - a_{ii}) \tag{43}$$

It should be noted that (43) is a geometric distribution. Clearly for many real-world phenomena, a geometric distribution is inappropriate. Often a binomial, or discretized Gaussian may be more fitting. It is for this reason that the hidden semi-Markov model (HSMM) was created. An HSMM is essentially an HMM with explicit state duration densities. However, because the probability of transitioning to a new state depends on the time duration in the current state, then the process is no longer a pure first-order Markov chain. Due to the involved nature of the changes necessary to implement a HSMM, the formulations required for this model extension are not presented here, but the interested reader is encouraged to refer to [14], one of the early papers on this subject, or [3], which provides a more approachable summary of the theory.

6 Conclusion

HMMs have been successful analysis and classification tools in diverse applications from speech recognition, bioinformatics and genomics, to landmine detection. The extensions of the basic theory allow the flexibility to apply this model to discrete or continuous phenomena, and the ability to use multiple observations leads to increased accuracy in the estimation of model parameters. With research into the uses of these models still ongoing and new extensions being developed, there will undoubtedly be much more to look for in the coming years to expand on this theory.

References

[1] G. W. Lambert. Effects of sunlight and season on serotonin turnover in the brain. *Lancet*, 360(9348):1840–1842, December 2002.

[2] L. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *Annals of Mathematical Statistics*, 37(6):1554–1563, December 1966.

[3] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286. IEEE, February 1989.

[4] L. Baum and J. Egon. An inequality with applications to statistical estimation for probabilistic functions of a markov process and to a model for ecology. *Bulletin of American Mathematical Society*, 73:360–363, 1967.

[5] L. Baum and G. Sell. Growth functions for transformations on manifolds. *Pacific Journal of Mathematics*, 27(2):211–227, 1968.

[6] C. Bishop. *Pattern Recognition and Machine Learning.* Springer Science+Business Media, LLC, New York, 2006.

[7] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, IT-13(2):260–269, April 1967.

[8] Jr. G. David Fornet. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1973.

[9] G. Soules L. Baum, T. Petrie and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, February 1970.

[10] L. Rabiner S. Levinson and M. Sondhi. An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *The Bell System Technical Journal*, 62(4):1035–1074, 1983.

[11] T. Mann. Numerically stable hidden markov model implementation. February 2006.

[12] L. Liporace. Maximum likelihood estimation for multivariate observations of markov sources. *IEEE Transactions on Information Theory*, IT-28(5):729–734, September 1982.

[13] S. Levinson B. Juang and M. Sondhi. Maximum likelihood estimation for multivariate mixture observations of markov chains. *IEEE Transactions on Information Theory*, IT-32(2):307–309, March 1986.

[14] S. Levinson. Continuously variable duration hidden markov models for speech analysis. In *IEEE Conference on ICASSP '86*, volume 11, pages 1241–1244, April, 1986.

[15] P. Hart R. Duda and D. Stork. *Pattern Classification.* John Wiley & Sons, Inc., New York, second edition, 2001.

Appendix

Note: some definitions from [3].

$\lambda = \{A, B, \pi\}$, The model

$O = \{O_1 O_2 \dots O_t\}$, Observation sequence

$O_i =$ Individual observation vector

$S_i =$ State i

$q_t =$ Underlying state at time t

$\pi_i = P(q_1 = S_1)$, Initial state probability

$A = \{a_{ij}\}$, State Transition Probability Matrix

$a_{ij} = P(q_{t+1} = S_j | q_t = i, \lambda)$
The probability of transitioning to state j,
given the current state is state i

$B = \{b_j(O_t)\}$
Observation Probability Distribution Matrix

$b_j(O_t) = P(O_t | q_t = S_j)$
Probability of observing O_t,
given the current state is state j

$\alpha_t(i) = P(O_1 O_2 \dots O_t | q_t = S_i, \lambda)$
Probability of the partial observation
sequence from 1 to t, and
the state at time t is S_i

$\beta_t(i) = P(O_{t+1} O_{t+2} \dots O_T | q_t = S_i, \lambda)$
Probability of the partial observation
sequence from $t+1$ to T, given
the state at time t is S_i

$\gamma_t(i) = P(q_t = S_i | O, \lambda)$
Probability of being in state S_i at
time t, given the observation sequence

$\delta_t(i) = \max_{q_1 q_2 \dots q_t - 1} [P(q_1 q_2 \dots q_t = i, O_1 O_2 \dots O_t | \lambda]$
The highest probability along a single
path at time t which accounts for the
first t observations and ends in state S_i

$\psi_t(i) =$ The state at time t which
is along the path of greatest probability
that accounts for the first t
observations and ends in state S_i

$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$
Probability of being in state S_i at
time t and S_j at time $t+1$,
given the observation sequence

$\mathbf{O} = [\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, \dots, \mathbf{O}^{(K)}]$
Set of Observation Sequences

$\mathbf{O}^{(k)} = [O_1^{(k)} O_2^{(k)} \cdots O_{T_k}^{(k)}]$
The kth Observation Sequence