Class notes
Hidden Markov Models

Amir Navot and Lavi Shpigelman

Winter 2003

1 Introduction

Assume you are given the following problem. A real-world process produces a sequence of observable symbols. Our job is to guess the state of the process from the observation sequence. For example, assume you are given a set of sensor readings from a robot while it moves around in a room. Here, we might like to guess the robot's real position from the sensor readings. In general, the states and observations might be continuous. We will discuss only the case when both are discrete and finite. In the robot example we may choose to assume that it moves on a 100x100 grid and that the sensor readings can take on a finite set of values.

We assume that the relation between state and observation is probabilistic. Lets assume that there are N possible states, M possible sensor values and T time steps. In order to describe the probabilities of the model one needs to specify the joint probability of every possible pair of state sequence and observation sequence. This would require specifying $N^T \times M^T$ values which is simply not possible for any reasonable problem size. For example, in the robot problem, with a 10×10 position grid, 100 sensor values and 10 time steps we get 10^{40} values. Obviously we need a simplified model.

The simplifying assumptions that are made in HMM are that the state sequence is Markov, i.e. that the future of the process is independent of its past given its current state. Also we assume that the observation at time t is independent of anything else given the current process state. An additional assumption that is usually made is that the process is stationary, i.e. that the probabilities are not dependent of the specific time. These assumptions are approximately correct in many real-world situations.

2 The Model

- A random variable is denoted by a capital letter (e.g. X) and a specific value (instance) by the corresponding small case (e.g. x).

- A random time series (a.k.a. process) is also denoted by the corresponding bold-face letter (e.g. $\mathbf{X}$ is the same as $\{X_t\}_{t=1}^{T}$)
- Similarly, an instantiation of a random time series is also denoted by the corresponding small-case bold-face letter (e.g. $\mathbf{x}$ is the same as $\{x_t\}_{t=1}^{T}$)

Definition 1 (Markov Process) *A process $\mathbf{X}$ is a Markov Process if $p(X_{t+1}|X_{1\dots t}) = p(X_{t+1}|X_t)$. In words, The future is independent of the past given the present.*

Definition 2 (Hidden Markov Model) *A pair of processes $\{\mathbf{X}, \mathbf{Y}\}$ is constitutes a Hidden Markov Model (HMM) if:*

- $\mathbf{X}$ *is a Markov Process.*
- $p(Y_t|Y_1, \dots, Y_T, X) = p(Y_t|X_t)$

We will usually assume that $\mathbf{X}$ is hidden and $\mathbf{Y}$ is observed. From the definition we can see that the process is defined by $p(X_t|X_{t-1})$, $p(Y_t|X_t)$ and $p(X_1)$. We also assume that the process is *stationary*, that is, the probabilities are not time dependent:

$$\begin{aligned} p(X_t = j|X_{t-1} = i) &= A(i,j) \\ p(Y_t = j|X_t = i) &= B(i,j) \\ p(X_1 = i) &= q_i \end{aligned}$$

This means that we can represent the process by the set of parameters $\lambda = \{A, B, q\}$ where A and B are matrices and q is a vector.

There are many independences which are not explicitly given by the model definition but can be derived by standard probability axioms. For example

Statement 1 *For an HMM $p(\mathbf{X}, \mathbf{Y}) = p(X_1) \prod_t p(X_t|X_{t-1}) p(Y_t|X_t)$*

Proof:

$$\begin{aligned} p(\mathbf{X}, \mathbf{Y}) &= p(\mathbf{X}) p(\mathbf{Y}|\mathbf{X}) \\ &= p(\mathbf{X}) p(Y_1|\mathbf{X}) p(Y_2|\mathbf{X}) \dots p(Y_T|\mathbf{X}) \\ &= p(\mathbf{X}) \prod_t p(Y_t|X_t) \\ &= p(X_1) p(X_2|X_1) p(X_3|X_2, X_1) \dots p(X_T|X_{1\dots T-1}) \prod_t p(Y_t|X_t) \\ &= p(X_1) \prod_t p(X_t|X_{t-1}) p(Y_t|X_t) \end{aligned}$$

□

Another example is:

Example 1 $p(X_{10}|X_8, X_7) = p(X_{10}|X_8)$

Proof:

$$\begin{aligned} p(X_{10}|X_8, X_7) &= \sum_{X_9} p(X_{10}, X_9|X_8, X_7) \\ &= \sum_{X_9} p(X_9|X_8, X_7) p(X_{10}|X_8, X_7, X_9) \\ &\overset{\text{Markovity}}{=} \sum_{X_9} p(X_9|X_8) p(X_{10}|X_8, X_9) \\ &= \sum_{X_9} p(X_{10}, X_9|X_8) \\ &= p(X_{10}|X_8) \end{aligned}$$

□

It is also true that $p(Y_{10}|X_8, X_7) = p(Y_{10}|X_8)$

Deriving independence relations by such algebraic manipulations is rather tiresome. To ease these derivations, one can (with some additional proofs) use graphical representations.

Definition 3 (The graph that corresponds to an HMM) *The undirected graph that corresponds to an HMM is a graph with* $2T$ *vertexes, one for each* X_t *and for each* Y_t *with an edge between any to consecutive* X_t *and* X_{t+1} *and between* X_t *and* Y_t*. See fig. 1 for an illustration.*

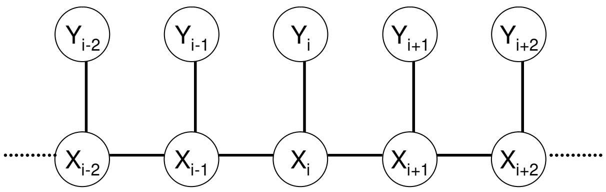

Figure 1: The Graph that corresponds to a Hidden Markov Process

Statement 2 *let* $A, B, C \subset \mathbf{X}, \mathbf{Y}$ *(i.e. each is a set of vertexes) then* $p(A|B, C) = p(A|B)$ *iff* B *separates* A *from* C *in the graph (i.e. any path from* A *to* C *passes through* B*).*

Proof: Specific case of the Hammersley Clifford theorem (which will be proved later on in the course).

□

3 The Three Problems of HMM

There are three key problems of interest that must be solved for HMMs to be useful. These are:

Likelihood: Given a series of observations $\mathbf{y}$ and a model λ, compute the likelihood $p(\mathbf{y}|\lambda)$.

Inference: Given a series of observations $\mathbf{y}$ and a model λ, compute the most likely series of hidden states $\mathbf{x}$.

Learning: Given a series of observations, learn best model λ.

3.1 Likelihood

Given a model $\lambda = \{A, B, q\}$, we can, in theory, compute the likelihood by

$$p(\mathbf{y}) = p(\mathbf{y}|\lambda) = \sum_{\mathbf{x}} p(\mathbf{x})p(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{x}} q_{x_1} \prod_{t=1}^{T} A(x_t|x_{t+1}) \prod_{t=1}^{T} B(x_t|y_t)$$

However, this computation is impossible in practice since its time complexity is $O(2T-1)N^T$ where N is the number of possible values for x_t. This problem can be overcome by a dynamic programming solution. Consider the forward variable $\alpha_t(i) = p(y_{1,\ldots,t}, x_t = i|\lambda)$, i.e. the probability of the partial observation sequence (until time t) and the state i at time t given the model λ. One can see that

$$p(\mathbf{y}) = \sum_i \alpha_T(i)$$

The α_ts can be found efficiently by the recursive procedure given in algorithm 1.

Algorithm 1 The Trellis (lattice) algorithm

1. $\alpha_1(j) = q_j B(j, \mathbf{y}_1), \quad 1 \leq j \leq N$
2. for $t = 1, 2, \ldots, T-1, \quad 1 \leq i \leq N$

$$\alpha_{t+1}(i) = B(i, y_{t+1}) \sum_{j=1}^{N} A(j,i)\alpha_t(j)$$

The proof of the algorithm is as follows

Proof:[of algorithm 1]

by definition of $\alpha_t(i)$ we see that $\alpha_1(i) = q(i)B(i, y_1)$.

induction step:

$$
\begin{aligned}
\alpha_{t+1}(i) &= p(y_{1,\ldots,t+1}, x_{t+1} = i) \\
&= \sum_j p(x_t = j, x_{t+1} = i, y_{1,\ldots,t+1}) \\
&= \sum_j p(x_{t+1} = i, y_{t+1} | x_t = j, y_{1,\ldots,t}) p(x_t = j, y_{1,\ldots,t}) \\
&\overset{\text{Markovity}}{=} \sum_j p(x_{t+1} = i, y_{t+1} | x_t = j) p(x_t = j, y_{1,\ldots,t}) \\
&= \sum_j p(y_{t+1} | x_{t+1} = i, x_t = j) p(x_{t+1} = i | x_t = j) p(x_t = j, y_{1,\ldots,t}) \\
&\overset{\text{Markovity}}{=} \sum_j p(y_{t+1} | x_{t+1} = i) p(x_{t+1} = i | x_t = j) p(x_t = j, y_{1,\ldots,t}) \\
&= \sum_j B(i, y_{t+1}) A(j, i) \alpha_t(j) \\
&= B(i, y_{t+1}) \sum_j A(j, i) \alpha_t(j)
\end{aligned}
$$

$\square$

Another equivalent way of computing the likelihood is by a backward procedure. Consider the backward variable $\beta_t(i) = p(y_{t+1,\ldots,T} | x_t = i, \lambda)$. i.e. the probability of the partial observation sequence from $t+1$ to the end, given state i at time t and the model λ. Then

$$
\begin{aligned}
p(\mathbf{y}) &= \sum_i p(y_{1,\ldots,T} | x_1 = i) p(x_1 = i) \\
&= \sum_i p(y_{2,\ldots,T} | x_1 = i, y_1) p(y_1 | x_1 = i) p(x_1 = i) \\
&= \sum_i p(y_{2,\ldots,T} | x_1 = i) p(y_1 | x_1 = i) p(x_1 = i) \\
&= \sum_i \beta_1(i) B(i, y_1) q(i)
\end{aligned}
$$

The $\beta_t(i)$ satisfies the following recursion:

1. $\beta_T(j) = 1, \quad 1 \le j \le N$
2. for $t = T-1, T-2, \ldots, 1, \quad 1 \le j \le N$

$$\beta_t(j) = \sum_{i=1}^{N} A(j, i) B(i, y_{t+1}) \beta_{t+1}(i)$$

The proof of this recursion is similar to the one for the αs and is left as an exercise.

A third way of computing the likelihood is by combination of $\alpha_t(i)$ and $\beta_t(i)$ for any time t:

$$\begin{aligned}
p(y) &= \sum_i p(y, x_t = i) \\
&= \sum_i p(y_{1,\ldots,t}, y_{t+1,\ldots,T}, x_t = i) \\
&= \sum_i p(y_{1,\ldots,t} | y_{t+1,\ldots,T}, x_t = i) p(y_{t+1,\ldots,T}, x_t = i) \\
&= \sum_i p(y_{1,\ldots,t} | x_t = i) p(y_{t+1,\ldots,T}, x_t = i) \\
&= \sum_i \alpha_t(i)\beta_t(i)
\end{aligned}$$

This is sometimes known as the forward-backward method.

3.2 Inference

Given a series of observations we wish to learn best model λ.There are several plausible optimality criteria for the 'best state sequence'. One is to choose the states which are individually most likely. This maximizes the expected number of correct individual states. i.e. $\max \sum_t p(x_t|\mathbf{y})$. This is sometimes referred to as the *Bit error rate*. Note that this maximization can be done by maximizing for each element separately. Another optimality criterion is to maximize the probability of the whole series. i.e. $\max p(\mathbf{x}|\mathbf{y})$. This is sometimes referred to as the *Word error rate*. The difference between these criteria can be large if the first method results in a transition $x_t \rightarrow x_{t+1}$ with low probability.

3.2.1 Maximizing the bit error rate

The maximization is done by calculating $p(x_t = i|\mathbf{y})$ for each t and i and choosing the i with the highest probability for each t. This computation can be done efficiently using α and β defined in section 3.1 as follows:

$$\begin{aligned}
p(x_t = i|\mathbf{y}) &= \frac{1}{p(\mathbf{y})} p(x_t = i, \mathbf{y}) \\
&= \frac{1}{p(\mathbf{y})} p(x_t = i, y_{1\ldots t}) p(y_{t+1,\ldots,T} | x_t = i, y_{1\ldots t}) \\
&= \frac{1}{p(\mathbf{y})} p(x_t = i, y_{1\ldots t}) p(y_{t+1,\ldots,T} | x_t = i) \\
&= \frac{1}{p(\mathbf{y})} \alpha_t(i)\beta_t(i)
\end{aligned}$$

3.2.2 Maximizing the word error rate

The solution to this problem is very similar to the solution for the bit error rate but with a different recursion variable $\delta_t(i)$ which is similar to $\alpha_t(i)$

$$\delta_t(i) = \max_{x_{1\ldots t-1}} p(x_t = i, y_{1\ldots t})$$

The following recursion holds:

$$\begin{aligned}
\delta_t(i) &= \max_{x_{1\ldots t-1}} p(x_t = i, y_{1\ldots t}) \\
&= \max_{x_{1\ldots t-1}} p(x_t = i, y_{1\ldots t-1}) p(y_t|x_t = i, y_{a\ldots t-1}) \\
&= \max_{x_{1\ldots t-1}} p(x_t = i, y_{1\ldots t-1}) p(y_t|x_t = i) \\
&= p(y_t|x_t = i) \max_{x_{1\ldots t-1}} p(x_t = i, y_{1\ldots t-1}) \\
&= p(y_t|x_t = i) \max_{x_{1\ldots t-2}} \max_j p(x_{t-1} = j, y_{1\ldots t-1}) p(x_t = i|x_{t-1} = j) \\
&= p(y_t|x_t = i) \max_j p(x_t = i|x_{t-1} = j) \max_{x_{1\ldots t-2}} p(x_{t-1} = j, y_{1\ldots t-1}) \\
&= p(y_t|x_t = i) \max_j p(x_t = i|x_{t-1} = j) \delta_{t-1}(j) \\
&= B(i, y_t) \max_j A(j, i) \delta_{t-1}(j)
\end{aligned}$$

The initial condition is

$$\delta_1(i) = q_i B(i, y_1)$$

From δ's definition it is obvious that $max_i \delta_T(i)$ is the joint probability of the observation and the most likely state sequence. To find the most likely sequence we define another parameter that remembers the choices in each max stage.

$$\psi_t(i) = \arg\max_j \delta_{t-1}(j) A(j, i)$$

The most likely sequence is

$$\begin{aligned}
i_T^* &= \arg\max_i \delta_T(i) \\
i_t^* &= \psi_{t+1}(i_{t+1}^*) \quad t = T-1, \ldots, 1
\end{aligned}$$

This algorithm is known as the *Viterbi* algorithm

3.3 Learning

The problem at hand is that of estimating the model's parameters $\lambda = \{A, B, q\}$ from given samples. We discuss two settings, the *fully observed* and the *partially observed*. In the fully observed setting, both the observations $\mathbf{y}$ and the true corresponding state sequence $\mathbf{x}$ are known, while in the partially observed setting, only the observations $\mathbf{y}$ are given. In both cases, we find the maximum-likelihood estimator.

3.3.1 Fully observed

We are interested in finding

$$\hat{A}, \hat{B}, \hat{q} = \arg\max_{A,B,q} p(\mathbf{x}, \mathbf{y}; A, B, q)$$

We do so by using a variation principle. We start by writing the log-likelihood and a set of constraints

$$\begin{aligned}
\log p(\mathbf{x}, \mathbf{y}; A, B, q) &= \log q(x_1) + \sum_t \log A(x_{t-1}, x_t) + \sum_t \log B(x_t, y_t) \\
&= \log q(x_1) + \sum_{i,j} n(i,j) \log A(i,j) + \sum_{i,j} m(i,j) \log B(i,j)
\end{aligned}$$

where

$$\begin{aligned}
n(i,j) &= \#(x_{t-1} = i, x_t = j) \\
m(i,j) &= \#(x_t = i, y_t = j)
\end{aligned}$$

And the constraints are

$$\begin{aligned}
\sum_j A(i,j) &= 1 \quad \forall i \\
\sum_j B(i,j) &= 1 \quad \forall i \\
\sum_j q(j) &= 1
\end{aligned}$$

Applying the Lagrange Multipliers technique to solve for A

$$\mathcal{L} = const + \sum_{i,j} n(i,j) \log A(i,j) + \sum_i \lambda_i \left(\sum_j A(i,j) - 1 \right)$$

Deriving with respect to $A(i,j)$ and equating with 0 yields

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial A(i,j)} &= \frac{n(i,j)}{A(i,j)} + \lambda_i = 0 \\
\Rightarrow A(i,j) &= -\lambda_i n(i,j) \\
\stackrel{\text{normalization}}{\Longrightarrow} A(i,j) &= \frac{n(i,j)}{\sum_j n(i.j)}
\end{aligned}$$

similarly,

$$\begin{aligned}
B(i,j) &= \frac{m(i,j)}{\sum_j m(i,j)} \\
q(j) &= \frac{n_1(j)}{\sum_j n_1(j)}
\end{aligned}$$

Note that $n(i,j)$ and $m(i,j)$ are sufficient statistics for the model.

3.3.2 Partially observed

In the partially observed setting, the state sequences are unknown. Therefore we can not count $m(i,j)$ and $n(i,j)$. Instead, we treat them as random variables. The relevant likelihood which we would like to maximize is

$$\hat{A}, \hat{B}, \hat{q} = \arg\max_{A,B,q} p(\mathbf{y}; A, B, q)$$

To fill in the missing (hidden) data, we use the EM (Expectation Maximization) algorithm. The algorithm consists of iterations between two steps:

In the first step, termed the EXPECTATION (E) step, the expected $\mathbf{x}$ is found, given the current model (found in the next step).

In the second MAXIMIZATION (M) step, the maximum-likelihood model is re-estimated using the currently expected $\mathbf{x}$ (found in the previous step).

To find the expected $\mathbf{x}$ we define the following variables:

$$\begin{aligned} \gamma_t(i) &= p(\mathbf{x}_t = i | \mathbf{y}, \lambda) \\ \xi_t(i,j) &= p(x_t = i, x_t + 1 = j | \mathbf{y}, \lambda) \end{aligned}$$

From the definition of $\alpha_t(i)$ and $\beta_t(i)$ given in sec. 3.1 it is clear that their values are given by:

$$\begin{aligned} \gamma_t(i) &= \frac{\alpha_t(i)\beta_t(i)}{p(\mathbf{y}|\lambda)} \\ \xi_t(i,j) &= \frac{\alpha_t(i)A(i,j)B(j, y_{t+1}\beta_{t+1}(i)}{p(\mathbf{y}|\lambda)} \end{aligned}$$

The expectation is calculated by summing over t of γ and ξ and the maximization step consists of assigning the expectancies to A, B and q:

$$\begin{aligned} q(i) &= \gamma_1(i) \\ A(i,j) &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \\ B(i,k) &= \frac{\sum_{t=1, y_t=k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \end{aligned}$$

The following theorem gives justification for the usage of EM:

Theorem 1

$$p(\mathbf{y}; A^{(i)}, B^{(i)}, q^{(i)}) \geq p\left(\mathbf{y}; A^{(i-1)}, B^{(i-1)}, q^{(i-1)}\right)$$

The proof will be given later in the course.