11 Hidden Markov Models (HMMs)

11.1 Model and Problem Description

We consider here a somewhat different model, which evolves on a discrete (finite) state space. The following elements describe the model:

1. A *finite* state space: $\mathcal{X} = \{1, \ldots, M\}$ with random states $X_k \in \mathcal{X}$.
2. The *state dynamics* is described by a time-homogeneous Markov chain:
$$p(X_{k+1} = j | X_k = i) = a(j|i) \equiv A_{ij}$$
with initial conditions $\pi_0 = (p(X_0 = i))_{i \in \mathcal{X}}$.
3. Measurements $Y_k \in \mathcal{Y}$ are obtained, and depend on the current state only:
$$p(Y_k = y | x_0^k, y_0^k) = p(Y_k = y | x_k)\,.$$
Define $b(y|x) = p(Y_k = y | X_k = x)$. The measurement space may be discrete, in which case $b(\cdot|x)$ is a probability mass function (pmf); or it may be continuous, and then $b(\cdot|x)$ will denote a probability density function (pdf).

The quantities (A, b, π_0) are the *natural model parameters.* In general, we have some model parameters θ on which the above depend.

It is usually assumed that the Markov chain X_n is ergodic (irreducible and non-periodic). The output process Y_n can be shown to inherit basic properties (stationarity, ergodicity, mixing) from the state process.

The basic computational problems for the HMM model are:

1. Output sequence probabilities: Compute $p(y_0^n)$.
2. State estimation: Given the measurements y_0^n, estimate X_0^n.
3. System identification/learning: Given y_0^n, estimate the model parameters θ.

The first two items assume *known* model (A, b, π_0). In the third the goal is to estimate the model, including the 'hidden' state dynamics.

Note that $p(y_0^n)$ is the likelihood function, that will be used for identification when the model parameters are unknown.

11.2 Output-Sequence Probabilities

The following computations which involve a given state sequence are easy:

$$p(x_0^n) = \prod_{k=0}^{n} p(x_k|x_0^{k-1})$$

(where $p(x_0|x_0^{-1}) \stackrel{\triangle}{=} \pi(x_0)$),

$$p(y_0^n|x_0^n) = \prod_{k=0}^{n} p(y_k|x_k)$$

$$p(y_0^n, x_0^n) = \prod_{k=0}^{n} p(x_k|x_{k-1})p(y_k|x_k)\,.$$

To compute $p(y_0^n)$, we can use

$$p(y_0^n) = \sum_{x_0^n \in \mathcal{X}^n} p(y_0^n, x_0^n)$$

However, as the number of possible sequences x_0^n is exponential, this direct computation becomes unfeasible unless n is small. This computation can be done much more efficiently using either forward or backward recursions.

Forward recursion: Compute $p(x_k, y_0^k)$ recursively as follows:

$$p(x_{k+1}, y_0^{k+1}) = b(y_{k+1}|x_{k+1}) \sum_{x_k \in \mathcal{X}} p(x_k, y_0^k)a(x_{k+1}|x_k)$$

with $p(x_0, y_0) = b(y_0|x_0)\pi(x_0)$.

We then obtain

$$p(y_0^n) = \sum_{x_n} p(x_n, y_0^n)\,.$$

This requires $O(nM^2)$ operations, as opposed to $O(nM^n)$ for direct computation.

Backward recursion: We can similarly compute $p(y_{k+1}^n|x_k)$ as follows:

$$p(y_k^n|x_{k-1}) = \sum_{x_k} p(y_{k+1}^n|x_k)a(x_k|x_{k-1})b(y_k|x_k)$$

starting from $p(y_{n+1}^n|x_n) \triangleq 1$. Here we have $p(y_0^n) \equiv p(y_0^n|x_{-1})$.

Posterior state distribution: Another quantity of interest is the conditional ('smoothed') state distribution $p(x_k|y_0^n)$. The forward and backward recursions may be combined to obtain this quantity. First,

$$p(x_k, y_0^n) = p(x_k, y_0^k, y_{k+1}^n) = p(x_k, y_0^k)p(y_{k+1}^n|x_k)\,.$$

This follows from the conditional independence of Y_0^k and Y_{k+1}^n given x_k. We may now compute $p(x_k|y_0^n) = p(x_k, y_0^n)/p(y_0^n)$.

Furthermore, the pairwise state distribution (that will be required later) can be computed as

$$p(x_{k-1}, x_k|y_0^n) = \frac{1}{C}p(x_{k-1}, y_0^{k-1})p(y_{k+1}^n|x_k)a(x_k|x_{k-1})b(y_k|x_k)$$

where $C = \sum_{x_k, x_{k-1}}\{\text{numerator}\}$ is the normalization constant.

11.3 MAP State Estimation

We now wish to find an estimate for the state sequence x_0^n, given the measurement sequence y_0^n. The primary concept here is the MAP estimator.

Single-state estimator:

For each $0 \le k \le n$, we can simply estimate x_k as

$$\hat{x}_k = \arg\max_{x_k} p(x_k, y_0^n)$$

where the latter probability was computed above.

State Sequence Estimation: The Viterbi Algorithm

We are usually interested in estimating the entire state sequence x_0^n. Not that this is *not* the same as the combined single-state estimates $\{\hat{x}_k\}_{k=0}^n$, that might yield unlikely (even 0-probability) sequences. We therefore consider

$$\hat{x}_0^n = \arg\max_{x_0^n} p(x_0^n, y_0^n)$$

where

$$p(x_0^n, y_0^n) = \prod_{k=0}^{n} p(x_k|x_{k-1})p(y_k|x_k)\,.$$

The Viterbi algorithm gives an iterative solution to this problem, which is a particular case of the Dynamic Programming algorithm.

Define the joint log-likelihood function $L_n(x_0^n) \stackrel{\triangle}{=} \log p(x_0^n, y_0^n)$. Then

$$L_n(x_0^n) = c_0(x_0) + \sum_{k=1}^{n} c_k(x_{k-1}, x_k)$$

where $c_k(x_{k-1}, x_k) = \log(p(x_k|x_{k-1})p(y_k|x_k))$. That is,

$$c_k(i,j) = \log(a(j|i)b(y_k|j)) \text{ for } k \ge 1, \text{ and } c_0(j) = \log(\pi_0(j)b(y_0|j))\,.$$

Obviously, $L_k(x_0^k) = L_{k-1}(x_0^{k-1}) + c_k(x_{k-1}, x_k)$.

Let

$$v_k(x_k) = \max_{x_0^{k-1}} L_k(x_0^k)\,, \quad x_k \in \mathcal{X}$$

It is easily verified that

$$v_k(j) = \max_i \{v_{k-1}(i) + c_k(i,j)\}\,, \quad j \in \mathcal{X}$$

and $\max_{x_0^n} L_n(x_0^n) = \max_j v_n(j)$. The maximizing state sequence is now obtained as

$$\begin{aligned} \hat{x}_n &= \arg\max_j v_n(j)\,, \\ \hat{x}_k &= \arg\max_i \{v_k(i) + c_{k+1}(i, \hat{x}_{k+1})\} \end{aligned}$$

11.4 Joint Parameter and State Estimation

Consider the problem of estimating the natural model parameters: $\theta = (A, b, \pi_0)$. When the state sequence is observed, this is an easy task. However, when only the measurements are available, it becomes considerably harder.

The basic estimator here is the MLE:

$$\hat{\theta} = \max_{\theta \in \Theta} p(y_0^n | \theta)$$

where Θ is the set of feasible parameters.

Some basic observations:

- It is hard to compute (and optimize) $p(y_0^n|\theta)$.
- However, it is "easy" to compute $p(y_0^n, x_0^n|\theta)$. Unfortunately, x_0^n is unknown.

To exploit the last observation, we can use the following two-step iterative scheme:

1. Given some estimate $\hat{\theta}_m$, compute $p(x_0^n|y_0^n, \hat{\theta}_m) \equiv \hat{p}(x_0^n)$.
2. Using $\hat{p}(x_0^m)$, get an improved estimate $\hat{\theta}_{m+1}$.

The resulting algorithm is known as the Baum algorithm (1966). It is a special case of the EM algorithm (1977).

The Baum Algorithm:

Recall that y_0^n is given. Define the log-likelihood function:

$$L(\theta) = \log p(y_0^n|\theta)$$

which we wish to maximize.

For a given estimate $\hat{\theta}_m$, define the following *auxiliary function*:

$$\begin{aligned} Q(\theta, \hat{\theta}_m) &= E\Big\{\log p(x_0^n, y_0^n|\theta) \mid y_0^n, \hat{\theta}_m\Big\} \\ &\equiv \sum_{x_0^n} p(x_0^n|y_0^n, \hat{\theta}_m)\ \log p(x_0^n, y_0^n|\theta)\,. \end{aligned}$$

This may be viewed as an "averaged" log-likelihood function.

The algorithm is, in principle:

(1) Expectation stage: given $\hat{\theta}_m$, compute $Q(\theta, \hat{\theta}_m)$ [using $p(x_0^n|y_0^n, \hat{\theta}_n)$].

(2) Maximization stage: $\hat{\theta}_{m+1} = \arg\max_\theta Q(\theta, \hat{\theta}_m)$.

In general, this algorithm increases the likelihood $L(\hat{\theta}_m)$ at each stage, as we show below. However, it can only find *local* maxima of $L(\theta)$.

The Re-estimation Formulas:

The process of obtaining $\hat{\theta}_{m+1}$ from $\hat{\theta}_m$ is often called re-estimation. Explicit formulas can be given in certain cases.

We start by computing $p(x_{t-1}, x_t | y_0^n, \hat{\theta}_m)$. This can be done using the backward/forward iteration (see section 11.2), with the model $\hat{\theta}_m$. Now

- $\hat{\pi}_0$ and $\hat{A}(j|i)$ are given by

$$(\hat{\pi}_0)_j = p(x_0 = j | y_0^n, \hat{\theta}_m)$$

$$\hat{a}(j|i) = \frac{\sum_{t=1}^n p(x_{t-1} = i, x_t = j | y_0^n, \hat{\theta}_m)}{\sum_{t=1}^n p(x_{t-1} = i | y_0^n, \hat{\theta}_m)}$$

- If Y is *discrete*, then

$$\hat{b}(y|i) = \frac{\sum_{t=0}^n p(x_t = i, y_t = y | y_0^n, \hat{\theta}_m)}{\sum_{t=0}^n p(x_t = i | y_0^n, \hat{\theta}_m)}$$

- If Y is Gaussian, with $(y_t | x_t = i) \sim \mathcal{N}(\mu_i, R_i)$, then

$$\hat{\mu}_i = \frac{\sum_{t=0}^n p(x_t = i | y_0^n, \hat{\theta}_m)\, y_t}{\sum_{t=0}^n p(x_n = i | y_0^n, \hat{\theta}_m)} = \text{“averaging” over } \ y_t \,.$$

$$\hat{R}_i = \text{similar averaging, with } y_t \text{ replaced by } (y_t - \hat{\mu}_i)(y_t - \hat{\mu}_i)^T \,.$$

11.5 The EM Algorithm

We briefly describe the EM algorithm in a general (abstract) setting, not restricted to HMMs.

The basic model:

θ – Unknown parameter, $\theta \in \Theta \subset \mathbb{R}^r$.

X – Hidden variable (or "state").

Y – Observation (noisy function of the state).

We assume that $p_\theta(x)$ and $p_\theta(y|x)$ are known. Hence we can compute (in principle)

$$p_\theta(y) = \int_x p_\theta(y|x) p_\theta(x) dx \,.$$

Define the log-likelihood function:

$$L(\theta) = \log p_\theta(y) \,.$$

Our goal is to compute the maximum likelihood estimator:

$$\hat{\theta} = \arg\max_{\theta \in \Theta} L(\theta) \,.$$

Since $L(\theta)$ is hard to maximize directly, we use the two step EM procedure.

a. The EM iteration

Recall that the measurement y is given. We start with some guess $\hat{\theta}_0$, and iterate for $m \geq 0$:

(1) E-step: Compute

$$\begin{aligned} Q(\theta, \hat{\theta}_m) &= E\Big(\log p_\theta(X, y) | Y = y, \hat{\theta}_m\Big) \\ &= \int_x \log p_\theta(x, y) dp_{\hat{\theta}_m}(x|y) \end{aligned}$$

(2) M-step: $\hat{\theta}_{m+1} = \arg\max_\theta Q(\theta, \hat{\theta}_m)\,.$

Stop when $\|\hat{\theta}_{m+1} - \hat{\theta}_m\| \leq \epsilon$.

b. EM increases $L(\theta)$

We will show below that for every θ and $\hat{\theta}_m$,

$$L(\theta) - L(\hat{\theta}_m) \geq Q(\theta, \hat{\theta}_m) - Q(\hat{\theta}_m, \hat{\theta}_m)\,. \qquad (*)$$

Therefore, taking $\hat{\theta}_{m+1} = \arg\max_\theta Q(\theta, \hat{\theta}_m)$, we obtain

$$L(\hat{\theta}_{m+1}) \geq L(\hat{\theta}_m)$$

with equality only if $\hat{\theta}_{m+1} = \hat{\theta}_m$ (more precisely, if $\hat{\theta}_m$ is a maximizer of $Q(\theta, \hat{\theta}_m)$).

To simplify notation, let $\hat{E}_m(\cdot)$ denote expectation over X with respect to $p_{\hat{\theta}_m}(x|y)$. Then

$$Q(\theta, \hat{\theta}_m) = \hat{E}_m \left(\log p_\theta(X, y)\right)\,.$$

To establish $(*)$, note that

$$\begin{aligned} Q(\theta, \hat{\theta}_m) &= \hat{E}_m \log p_\theta(X, y) \\ &= \hat{E}_m \Big(\log p_\theta(X|y) + \log p_\theta(y) \Big) \\ &= \hat{E}_m \log p_\theta(X|y) + L(\theta)\,. \end{aligned}$$

Therefore:

$$Q(\theta, \hat{\theta}_m) - Q(\hat{\theta}_m, \hat{\theta}_m) = \hat{E}_m \left\{ \log \frac{p_\theta(X|y)}{p_{\hat{\theta}_m}(X|y)} \right\} + L(\theta) - L(\hat{\theta}_m)\,.$$

To show that $\hat{E}_m\{\dots\} \leq 0$, we use Jensen's inequality:

$$E(\log Z) \leq \log E(Z)\,.$$

Therefore:

$$\hat{E}_m\{\dots\} \le \log \hat{E}_m\left(\frac{p_\theta(X|y,\theta)}{p_{\hat{\theta}_m}(X|y)}\right) = \log \int_x \frac{p_\theta(x|y)}{p_{\hat{\theta}_m}(x|y)} p_{\hat{\theta}_m}(x|y)dx$$
$$= \log 1 = 0\,.$$

c. EM as a max-max procedure

An interesting interpretation of the EM can be obtained by looking at the function:

$$F(\theta, P_0) \triangleq \int_x \log\left(\frac{p_\theta(x,y)}{P_0(x)}\right) P_0(x)dx$$

where P_0 is some pdf in x.

It can be shown that:

$$\arg\max_{P_0} F(\theta, P_0) = p_\theta(x|y)\,.$$

Indeed,

$$F(\theta, P_0) = \int_x \log\left(\frac{p_\theta(x|y)}{P_0(x)}\right) P_0(x)dx + L(\theta)$$

and for any pair of distributions $q(x)$ and $p(x)$ we have that

$$\int_x \log\left(\frac{p(x)}{q(x)}\right) q(x)dx \le \int_x \left(1 - \frac{p(x)}{q(x)}\right) q(x)dx = 1 - 1 = 0$$

with equality for $q = p$. Therefore,

$$\max_{P_0} F(\theta, P_0) = L(\theta)\,,$$

and

$$\max_{P_0,\theta} F(\theta, F_0) = \max_\theta L(\theta)\,.$$

The EM algorithm can now be viewed as trying to maximize $F(\theta, P_0)$ by alternately maximizing in each argument, while keeping the other fixed:

(1) E-step: maximize $F(\hat{\theta}_m, P_0)$ in P_0, to get $\hat{P}_m = p_{\hat{\theta}_m}(x|y)$.

(2) M-step: maximize $F(\theta, \hat{P}_m)$ to get $\hat{\theta}_{m+1}$. This is the same as maximizing $Q(\theta, \hat{\theta}_m)$, since $F(\theta, \hat{P}_m) = Q(\theta, \hat{\theta}_m) - C$, where C does not depend on θ.

d. Example: Re-estimation for exponential families

Suppose that $p(x)$ and $p(y|x)$ depend on different parameters. That is

$$p_\theta(x, y) \equiv p_\theta(x) p_\theta(y|x) = p_\lambda(x) p_\mu(y|x)$$

where $\theta = (\lambda, \mu) \in \Theta_1 \times \Theta_2$.

For HMMs, indeed we have $\lambda = (\pi_0, A)$ and $\mu = b$.

It follows that

$$\begin{aligned} Q(\theta, \hat{\theta}_m) &= \hat{E}_m(\log p_\theta(X, y)) \\ &= \hat{E}_m(\log p_\lambda(X))) + \hat{E}_m(\log p_\mu(y|x)) \\ &\triangleq Q_1(\lambda, \hat{\theta}_m) + Q_2(\mu, \hat{\theta}_m) \end{aligned}$$

and

$$\max_\theta Q(\theta, \hat{\theta}_m) = \max_\lambda Q_1(\lambda, \hat{\theta}_m) + \max_\mu Q_2(\mu, \hat{\theta}_m) \, .$$

Consider the first term. Assume that $p_\lambda(x)$ is an exponential family of distributions, namely

$$\begin{aligned} p_\lambda(x) &= \frac{1}{\alpha(\lambda)} \beta(x) \exp\Big[\sum_{i=1}^{s} c_i(\lambda) T_i(x)\Big] \\ &= \beta(x) \exp\Big[c(\lambda)' T(x) - \log \alpha(\lambda)\Big] \, , \qquad \lambda \in \mathbb{R}^d \, . \end{aligned}$$

This includes most distributions of interest, including Gaussian, Poisson, Binomial, Uniform and more. The vector $T(x)$ is the *sufficient statistic* of that family.

We the have

$$\begin{aligned} \arg\max_\lambda Q_1(\lambda, \hat{\theta}_m) &= \arg\max_\lambda \Big\{ \hat{E}_m[c(\lambda)' T(x)] - \log \alpha(\lambda) \Big\} \\ &= \arg\max_\lambda \Big\{ c(\lambda)' \hat{T}_{m+1} - \log \alpha(\lambda) \Big\} \end{aligned}$$

where $\hat{T}_{m+1} = \hat{E}_m(T(X))$.

We can therefore compute $\hat{\lambda}_{m+1}$ as follows:

1. E-step: Compute $\hat{T}_{m+1} = \hat{E}_m(T(X))$.

2. M-step: $\hat{\lambda}_{m+1} = \arg\max_\lambda \{c(\lambda)'\hat{T}_{m+1} - \log \alpha(\lambda)\}$

References: Pointers to HMM and EM literature

HMMs: A primer on HMMs in the context or speach recognition:

- L.R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proc. IEEE*, vol. 64, pp. 532–556, 1989.

Several textbooks have chapters on HMMs. In the context of speech-oriented applications, we mention:

- L.R. Rabiner and B. Juang, *Fundamentals of Speech Recognition,* Prentice-Hall, 1993.
- F. Jelinek, *Statistical Methods for Speech Recognition,* MIT Press, 1999.

A comprehensive recent overview can be found in the survey paper:

- Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Trans. Inform. Theory,* vol. 48, no. 6, pp. 1518-1569, June 2002.

EM: The EM is mentioned in most textbooks on statistical parameter estimation and machine learning. A simple introduction paper:

- T.K. Moon, "The Expectation-Maximization algorithm," *IEEE Signal Processing Magazine,* November 1996, pp. 47–60.

A comprehensive treatment can be found in

- G.J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions,* Wiley, 1997.

EM+Kalman filtering: See the following paper and references therein.

- L. Deng and X. Shen, "Maximum likelihood in statistical estimation of dynamic systems: Decomposition algorithm and simulation results," *Signal Processing,* vol. 57, 1997, pp. 65–79.