MSA101/MVE187 2020 Lecture 11.1 Hidden Markov Models

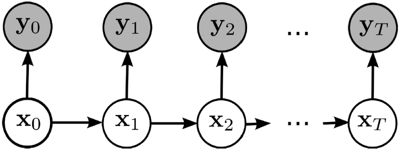
Petter Mostad

Chalmers University

September 30, 2020

Hidden Markov Models (HMM)

- Many types of data have a *sequential* nature: Time data, DNA data,. . .
- A common approach is to assume a *hidden* state of nature, changing like a Markov chain, with *observed* data depending on the hidden state.
- The model can be drawn as a Bayesian network:

  This is the general structure; there may also be a dependency of y_i on y_{i-1}.
- The y_i's are generally *observed*, the x_i's are *hidden*, the direction is often *time*.
- Examples: Visual interpretation for self-driving cars. Finding genes in DNA sequences.

Toy example

In this lecture we will work with a simple toy example of an HMM:

- The hidden variables $x_1, \ldots, x_N$ have possible values $1, \ldots, M$, and transition probabilities in the chain are (initially):

$$x_i \text{ given } x_{i-1} \text{ is } \begin{cases} \text{with prob. } 1/3: & x_{i-1}+1 \text{ if possible, otherwise } x_{i-1}. \\ \text{with prob. } 1/3: & x_{i-1}. \\ \text{with prob. } 1/3: & x_{i-1}-1 \text{ if possible, otherwise } x_{i-1}. \end{cases}$$

- The observed variables y_i are Poisson distributed with expectations given by the x_i:
- See the R code for simulated examples where we assume that $x_0 = 1$.

Inference for HMMs

We will in this lecture look at three types of inference connected to HMMs:

- Find the marginal density $\pi(x_i \mid y_1 \ldots, y_N)$ for each i. The Forward-Backward algorithm.
- Assume the x_i have finite sets of possible values. Find the sequence $x_1, \ldots, x_N$ of values such that

  $$\pi(x_1, \ldots, x_N \mid y_1, \ldots, y_N)$$

  is maximized. This is the Viterbi algorithm.
- Assume the x_i have finite sets of possible values. Assume that the transition probabilities of the Markov chain are unknown. Find the values for these maximizing their posterior given observations of $y_1, \ldots, y_N$. This is the Baum-Welsh algorithm.
- NOTE: The forward-backward algorithm is formulated in terms of an HMM, but the same ideas can be generalized to a Bayesian network of any shape, becoming a "message passing" algorithm.

MSA101/MVE187 2020 Lecture 11.2
The Forward-Backward algorithm

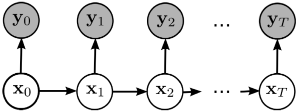
Petter Mostad

Chalmers University

September 30, 2020

The Forward-Backward algorithm

Message passing applied to a *Hidden Markov Model*.



Objective: Compute the marginal posterior distribution of every x_i given data $y_0, \dots, y_T$: Use $\pi(x_i \mid y_0 \dots, y_T) \propto \pi(y_{i+1}, \dots, y_T \mid x_i)\pi(x_i \mid y_0, \dots, y_i)$ and

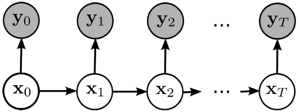
1. Forward: For $i = 0, \dots, T$ compute $\pi(x_i \mid y_0, \dots, y_i)$ using

$$\begin{aligned}\pi(x_i \mid y_0, \dots, y_i) &\propto \pi(y_i \mid x_i)\pi(x_i \mid y_0, \dots, y_{i-1}) \\ &= \pi(y_i \mid x_i) \int \pi(x_i \mid x_{i-1})\pi(x_{i-1} \mid y_0, \dots, y_{i-1})\, dx_{i-1}\end{aligned}$$

2. Backward: For $i = T-1, \dots, 0$ compute $\pi(y_{i+1}, \dots, y_T \mid x_i)$ using

$$\pi(y_{i+1}, \dots, y_T \mid x_i) = \int \pi(y_{i+2}, \dots, y_T \mid x_{i+1})\pi(y_{i+1} \mid x_{i+1})\pi(x_{i+1} \mid x_i)\, dx_{i+1}$$

R example with the Forward-Backward algorithm



- The hidden chain $x_0 \to \cdots \to x_N$ is a random walk on the integers $\{1, \ldots, M\}$.
- The (prior) transition probabilities from x_i to x_{i+1} is to increase with 1 (if possible) with probability $1/3$, to decrease with 1 (if possible) with probability $1/3$, and otherwise stay put.
- We use the model $y_i \mid x_i \sim \text{Poisson}(x_i)$ and assume the y_i are observed.
- We use the Forward-Backward algorithm to find the marginal posterior probability for each x_i.

MSA101/MVE187 2020 Lecture 11.3
The Viterbi algorithm

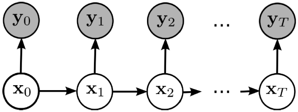
Petter Mostad

Chalmers University

September 30, 2020

The Viterbi algorithm

We consider an HMM where the x_i have a finite state space $\{1, \dots, M\}$:



Objective: Compute the vector $x_0, \dots, x_T$ which maximizes the posterior $\pi(x_0, \dots, x_T \mid y_0, \dots, y_T)$, i.e., maximizes $\pi(x_0, \dots, x_T, y_0, \dots, y_T)$.

- First formulation of an algorithm: Sequentially, for $i = 0, \dots, T$, compute and store
  - For each $j = 1, \dots, M$, the sequence $\hat{x}_0, \dots, \hat{x}_i$ maximizing $\pi(\hat{x}_0, \dots, \hat{x}_i, y_0, \dots, y_i)$ while $\hat{x}_i = j$.
  - For each $j = 1, \dots, M$, the value of the maximum above.
- Note that

  $$\pi(x_0, \dots, x_i, y_0, \dots, y_i) = \pi(x_0, \dots, x_{i-1}, y_0, \dots, y_{i-1}) \cdot \pi(x_i \mid x_{i-1}) \pi(y_i \mid x_i)$$

  Thus the results for stage i with $\hat{x}_i = j$ can be found by finding the $\hat{x}_{i-1}$ in $\{1, \dots, M\}$ maximizing

  $$\pi(\hat{x}_0, \dots, \hat{x}_{i-1}, y_0, \dots, y_{i-1}) \cdot \pi(x_i = j \mid \hat{x}_{i-1})$$

The Viterbi algorithm

- Thus results for the i'th step in the sequence can be computed by considering all combinations of values for x_i and x_{i-1} together with results from the $i-1$'th step.
- Improved and final formulation of the algorithm: For each i and j, you only need to store $\hat{x}_{i-1}$, not the whole sequence $\hat{x}_0, \ldots, \hat{x}_{i-1}, \hat{x}_i = j$. THEN: At any point, $(\hat{x}_1, \ldots, \hat{x}_i)$ can be reconstructed tracing backwards through stored information.
- Consider our toy example in R.

MSA101/MVE187 2020 Lecture 11.4
The Baum-Welsh algorithm

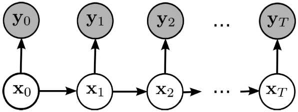
Petter Mostad

Chalmers University

September 30, 2020

The Baum-Welch algorithm

We now consider an HMM where all the x_i have a finite state spaces



but where some of the parameters of the distributions $\pi(X_0)$, $\pi(X_i \mid X_{i-1})$, and $\pi(Y_i \mid X_i)$ are unknown. Objective: Given fixed values for the y_i, find maximum likelihood estimates for the parameters in the model.

- Note: By adding nodes representing the unknown parameters, and assuming flat priors, the problem becomes that of computing the parameters maximizing the posterior, i.e., finding the MAP.
- Idea: Use the EM algorithm, with the values of the x_i as the augmented data.
- The E step of the EM algorithm is computed using (a small generalization of) the Forward-Backward algorithm.

The Baum-Welch algorithm: Example

For simplicity we assume each X_i can have values $1, \ldots, M$. Let

$$\theta = (q, p) = ((q_1, \ldots, q_M), (p_{11}, \ldots, p_{MM}))$$

be the parameters we want to estimate, where

$$\begin{aligned} q_j &= \Pr(X_0 = j) \\ p_{jk} &= \Pr(X_i = k \mid X_{i-1} = j) \end{aligned}$$

The full loglikelihood given θ becomes

$$\begin{aligned} & \log\left(\pi(x_0, \ldots, x_T, y_0, \ldots, y_T \mid \theta)\right) \\ = \; & \log\left(\pi(x_0 \mid \theta) \prod_{i=1}^{T} \pi(x_i \mid x_{i-1}, \theta) \prod_{i=0}^{T} \pi(y_i \mid x_i)\right) \\ = \; & \log \pi(x_0 \mid \theta) + \sum_{i=1}^{T} \log \pi(x_i \mid x_{i-1}, \theta) + \sum_{i=0}^{T} \log \pi(y_i \mid x_i) \\ = \; & C + \sum_{j=1}^{M} I(x_0 = j) \log q_j + \sum_{i=1}^{T} \sum_{j=1}^{M} \sum_{k=1}^{M} I(x_{i-1} = j) I(x_i = k) \log p_{jk} \end{aligned}$$

The Baum-Welch algorithm: Example continued

- In the E step, we would like to compute the expectation of the full loglikelihood under the distribution $\pi(x_0, \ldots, x_T \mid y_0, \ldots, y_T, \theta^{old})$ for some set of parameters θ^{old}.
- Thus we need to compute the expectations $\mathrm{E}\left[I(x_0 = j)\right]$ and $\mathrm{E}\left[I(x_{i-1} = j) I(x_i = k)\right]$ under this distribution.
- Fixing θ^{old}, we can use the Forward-Backward algorithm to compute the densities $\pi(x_i \mid y_0, \ldots, y_i)$ and $\pi(y_{i+1}, \ldots, y_T \mid x_i)$. Further we have that

$$
\begin{aligned}
& \pi(x_i, x_{i+1} \mid y_0, \ldots, y_T) \\
\propto \quad & \pi(y_{i+1}, \ldots, y_T \mid x_i, x_{i+1}) \pi(x_i, x_{i+1} \mid y_0, \ldots, y_i) \\
\propto \quad & \pi(y_{i+2}, \ldots, y_T \mid x_{i+1}) \pi(y_{i+1} \mid x_{i+1}) \pi(x_{i+1} \mid x_i) \pi(x_i \mid y_0, \ldots, y_i)
\end{aligned}
$$

making it possible to compute the joint posterior for x_i and x_{i+1} from these densities.

The Baum-Welch algorithm: Example continued

The algorithm can now be summed up as

- Choose starting parameters θ^{old}.
- Run the Forward-Backward algorithm on the Markov model with parameters θ^{old} to compute the numbers $\mathrm{E}\left[I(x_0 = j)\right]$ and $\mathrm{E}\left[I(x_{i-1} = j)I(x_i = k)\right]$.
- Find the θ maximizing the expected loglikelihood

$$\sum_{j=1}^{M} \mathrm{E}\left[I(x_0 = j)\right] \log q_j + \sum_{i=1}^{T} \sum_{j=1}^{M} \sum_{k=1}^{M} \mathrm{E}\left[I(x_{i-1} = j)I(x_i = k)\right] \log p_{jk}$$

  In fact, we get

$$\hat{q}_j = \mathrm{E}\left[I(x_0 = j)\right] \text{ and } \hat{p}_{jk} = \frac{\sum_{i=1}^{T} \mathrm{E}\left[I(x_{i-1} = j)I(x_i = k)\right]}{\sum_{k=1}^{M} \sum_{i=1}^{T} \mathrm{E}\left[I(x_{i-1} = j)I(x_i = k)\right]}$$

- Set $\theta^{old} = ((\hat{q}_1, \ldots, \hat{q}_M), (\hat{p}_{11}, \ldots, \hat{p}_{MM}))$ and iterate until convergence.