STAT 542: Statistical Learning

Hidden Markov Model

Ruoqing Zhu, Ph.D. <rqzhu@illinois.edu>

Course Website: https://teazrq.github.io/stat542/

April 25, 2022

Department of Statistics
University of Illinois at Urbana-Champaign

Outline

- The Dishonest Casino Example
- Hidden Markov Models

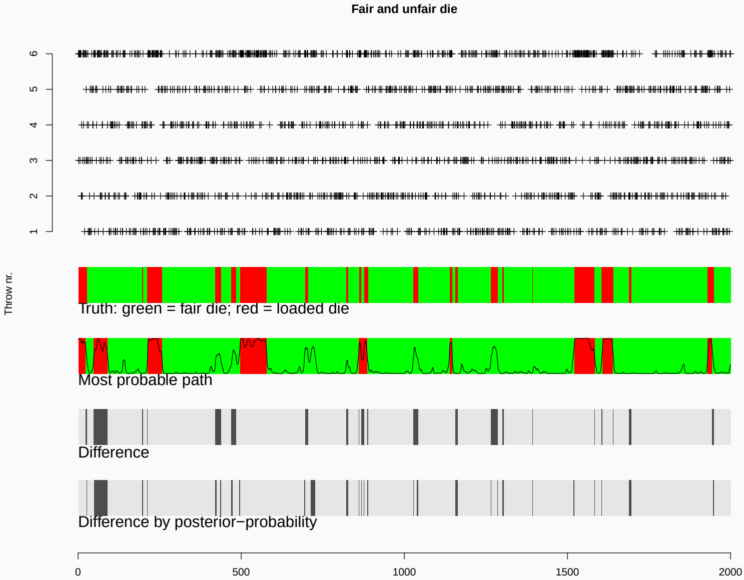
The Dishonest Casino Example

- An example taken from Durbin et. al. (1999).
- A dishonest casino uses two dice, one of them is fair and the other one is loaded.

Face/Prob	"1"	"2"	"3"	"4"	"5"	"6"
Fair Die	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$
Loaded Die	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{2}$

- The observer doesn't know which die is actually taken (the state is hidden), but the sequence of throws (observations) can be used to infer which die (state) was used.

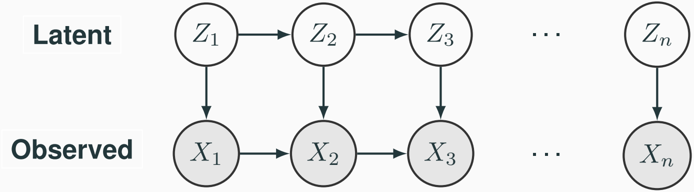
The Dishonest Casino Example

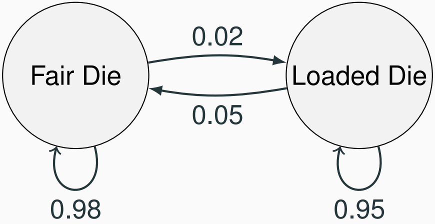


Hidden Markov Model

- Consider a Hidden Markov Model (HMM) for $(\mathbf{Z}, \mathbf{X}) = (Z_1, \ldots, Z_n, X_1, \ldots, X_n)$ where X_i's are observed (face of a dice) and Z_i's are hidden (fair or loaded). Let's assume that both $\mathbf{Z}$ and $\mathbf{X}$ are discrete random variables, taking m_z and m_x possible values, respectively. So the HMM is parameterized by $\theta = (w, A, B)$ where
  - $w_{m_z \times 1}$: distribution for Z_1, an initial stage.
  - $A_{m_z \times m_z}$: the transition probability matrix from Z_t to Z_{t+1}.
  - $B_{m_z \times m_x}$: the probability matrix (the emission distribution) for observing X_t under each hidden stage Z_t.
- The behavior of a HMM is fully determined by the three probabilities w, A, and B, and implicitly m_z and m_x.

Hidden Markov Model





Elements of a HMM

- For the Dishonest Casino Example, we have
  - $m_z = 2$, $m_x = 6$
  - $A = \begin{bmatrix} 0.98 & 0.02 \\ 0.05 & 0.95 \end{bmatrix}$
  - $B = \begin{bmatrix} \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{2} \end{bmatrix}$
  - $w = \left(\frac{1}{2}, \frac{1}{2}\right)$, equal probabilities if no strong prior believe
- We can calculate the probabilities of the observed data based on any given parameter value.

Modeling the data

- How to model the data and detect the underlying states (which die was used)?
- The underlying states $\{Z_t, t = 1, \ldots, n\}$ is a markov chain, that satisfies the following assumptions:
- The memoryless assumption:

$$\mathbf{p}(Z_t | Z_{t-1}, \ldots, Z_1) = \mathbf{p}(Z_t | Z_{t-1})$$

- The stationary assumption:

$$\mathbf{p}(Z_t | Z_{t-1}) = \mathbf{p}(Z_2 | Z_1), \text{ for } t = 2, \ldots, n$$

Modeling the data

- The log-likelihood on the observed data is given by

$$\begin{aligned}\log\left[\mathbf{p}(\mathbf{x}|\boldsymbol{\theta})\right] &= \log\left[\sum_{\mathbf{z}}\mathbf{p}(\mathbf{x},\mathbf{z}|\boldsymbol{\theta})\right] \\ &= \log\left[\sum_{\mathbf{z}}\mathbf{p}(\mathbf{z}|\boldsymbol{\theta})\mathbf{p}(\mathbf{x}|\mathbf{z},\boldsymbol{\theta})\right]\end{aligned}$$

  which is very hard to optimize due to the summation inside the log (generally not convex).

- Note: here $\mathbf{x}$ and $\mathbf{z}$ are the observed vectors of the sequences $\mathbf{X}$ and $\mathbf{Z}$, respectively.

- The the Baum-Welch algorithm is developed to solve this problem. It uses the EM algorithm and the forward-backward algorithm.

EM algorithm (discrete):

- E-step: Under the current value of $\boldsymbol{\theta}$, denoted as $\boldsymbol{\theta}^{(k)}$, find $\mathbf{p}(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^{(k)})$, the distribution of the unobserved variables given the observed data and $\boldsymbol{\theta}^{(k)}$. Then calculate:

$$
\begin{aligned}
g(\theta) &= \mathbb{E}_{\mathbf{Z}|\mathbf{x},\boldsymbol{\theta}^{(k)}} \log \mathbf{p}(\mathbf{x}, \mathbf{Z}|\boldsymbol{\theta}) \\
&= \sum_{\mathbf{z}} \mathbf{p}(\mathbf{Z} = \mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^{(k)}) \log \mathbf{p}(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})
\end{aligned}
$$

- M-step: Re-estimate the parameter $\boldsymbol{\theta}$ to maximize $g(\boldsymbol{\theta})$:

$$
\boldsymbol{\theta}^{(k+1)} = \arg\max_{\boldsymbol{\theta}} g(\boldsymbol{\theta})
$$

- How to calculate $\mathbf{p}(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^{(k)})$ for our HMM problem?

EM algorithm

- E-step: we first write out the log-likelihood of the complete data:

$$
\begin{aligned}
&\log \mathbf{p}(\mathbf{z}, \mathbf{x}|\boldsymbol{\theta}) \\
= \; &\log \left\{ \mathbf{p}(z_1) \prod_{t=1}^{n-1} \mathbf{p}(z_{t+1}|z_t, \cdots, z_1, \boldsymbol{\theta}) \prod_{t=1}^{n} \mathbf{p}(x_t|z_t, \cdots, z_1, \boldsymbol{\theta}) \right\}
\end{aligned}
$$

- Recall the memoryless and stationary assumptions, this can be simplified into

$$
\begin{aligned}
&\log \mathbf{p}(\mathbf{z}, \mathbf{x}|\boldsymbol{\theta}) \\
= \; &\log w(z_1) + \sum_{t=1}^{n-1} \log A(z_t, z_{t+1}) + \sum_{t=1}^{n} \log B(z_t, x_t)
\end{aligned}
$$

- Note that $\boldsymbol{\theta} = (w, A, B)$

EM algorithm

- We then try to integrate it over all possible values of $\mathbf{Z}$, based on a current iteration value $\theta^{(k)}$:

$$\mathbb{E}_{\mathbf{Z}|\mathbf{x},\boldsymbol{\theta}^{(k)}} \log \mathbf{p}(\mathbf{x}, \mathbf{Z}|\boldsymbol{\theta})$$

- To calculate this expectation, we need the conditional distribution of $Z|X, \boldsymbol{\theta}^{(k)}$, which is just the conditional expectations:

$$\begin{aligned} \gamma_t(i,j) &= \mathbf{p}(Z_t = i, Z_{t+1} = j|\mathbf{x}, \boldsymbol{\theta}^{(k)}) \\ \gamma_t(i) &= \mathbf{p}(Z_t = i|\mathbf{x}, \boldsymbol{\theta}^{(k)}) \end{aligned}$$

EM algorithm

- Suppose we already have the γ_t values, the E-step is:

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{Z}|\mathbf{x},\boldsymbol{\theta}^{(k)}} \log p(\mathbf{x}, \mathbf{Z}|\boldsymbol{\theta}) \\
&= \mathbb{E}_{\mathbf{Z}|\mathbf{x},\boldsymbol{\theta}^{(k)}} \left[\log w(Z_1) + \sum_{t=1}^{n-1} \log A(Z_t, Z_{t+1}) + \sum_{t=1}^{n} \log B(Z_t, x_t) \right] \\
&= \sum_{i=1}^{m_z} \gamma_1(i) \log w(i) + \sum_{t=1}^{n-1} \sum_{i,j=1}^{m_z} \gamma_t(i,j) \log A(i,j) \\
&\qquad + \sum_{t=1}^{n} \sum_{i}^{m_z} \gamma_t(i) \log B(i, x_t)
\end{aligned}
$$

- If we can compute each $\gamma_t(i,j)$ and $\gamma_t(i)$, this step is done.

EM algorithm

- At the M-step, we update the parameters $\theta = (w, A, B)$

$$
\begin{aligned}
w^{(k+1)}(i) &= \gamma_1(i), && i = 1, \dots, m_z; \\
A^{(k+1)}(i,j) &= \frac{\sum_{t=1}^{n-1} \gamma_t(i,j)}{\sum_{j'} \sum_{t=1}^{n-1} \gamma_t(i,j')}, && i, j = 1, \dots, m_z; \\
B^{(k+1)}(i,l) &= \frac{\sum_{t: x_t = l} \gamma_t(i)}{\sum_{t=1}^{n} \gamma_t(i)}, && i = 1, \dots, m_z,\ l = 1, \dots, m_x.
\end{aligned}
$$

- They are just the MLE estimators obtained by pooling and averaging a particular transition/emission event.
- For example, $B(i,l)$ is observing $X = l$ if the state is $Z = i$, so we go through all events with $Z = i$ in the entire chain, and average the events where $X = l$ is observed to get the probability.

Forward-Backward Algorithm

- In both steps, we need to calculate the conditional probabilities

$$\gamma_t(i,j) = \mathbf{p}(Z_t = i, Z_{t+1} = j | \mathbf{x}, \theta^{(k)}),$$

  which is the conditional probability of transiting from state i to state j at time point t given all the observed data $\mathbf{x}$, and

$$\gamma_t(i) = \mathbf{p}(Z_t = i | \mathbf{x}, \theta^{(k)}),$$

  the conditional probability of being at state i at time point t given all the observed data $\mathbf{x}$.

- We will use a forward-backward algorithm to calculate this.

Forward-Backward Algorithm

- With no risk of ambiguity, we will omit $\theta^{(k)}$ from the notation, i.e., $\mathbf{p}$ is always $\mathbf{p}_{\boldsymbol{\theta}^{(k)}}$
- For $\gamma_t(i,j)$, by Bayes' theorem, we have

$$
\begin{aligned}
\gamma_t(i,j) &= \mathbf{p}(Z_t = i, Z_{t+1} = j | \mathbf{x}) \\
&\propto \mathbf{p}(\mathbf{x}_{1:t}, Z_t = i, Z_{t+1} = j, x_{t+1}, \mathbf{x}_{(t+2):n}) \\
&= \underbrace{\mathbf{p}(\mathbf{x}_{1:t}, Z_t = i)}_{\alpha_t(i)} \times \underbrace{\mathbf{p}(Z_{t+1} = j | Z_t = i)}_{A(i,j)} \\
&\quad \times \underbrace{\mathbf{p}(x_{t+1} | Z_{t+1} = j)}_{B(j,x_{t+1})} \times \underbrace{\mathbf{p}(\mathbf{x}_{(t+2):n} | Z_{t+1} = j)}_{\beta_{t+1}(j)} \\
&\triangleq \alpha_t(i) A(i,j) B(j, x_{t+1}) \beta_{t+1}(j)
\end{aligned}
$$

Forward-Backward Algorithm

- $\alpha_t(i) = \mathbf{p}(\mathbf{x}_{1:t}, Z_t = i)$ is the forward probability of observing $\mathbf{x}_{1:t}$ and having state i at time t;
- $\beta_{t+1}(j) = \mathbf{p}(\mathbf{x}_{(t+2):n} | Z_{t+1} = j)$ is the backward probability of observing $\mathbf{x}_{(t+2):n}$ given state j at time t.
- Note: $\alpha_t(i)$ is a joint probability, and $\beta_{t+1}(j)$ is a conditional probability.
- How to calculate $\alpha_t(i)$ and $\beta_{t+1}(j)$? We do this recursively starting from the two end points $t = 1$ and $t = n$.

Forward Probability

- For the first time point $t = 1$:

$$\alpha_1(i) = \mathbf{p}(x_1, Z_1 = i) = w(i)B(i, x_1),$$

- For each t, we can then calculate the next time point $\alpha_{t+1}(i)$ using the information of $\alpha_t(i)$:

$$\begin{aligned}
\alpha_{t+1}(i) &= \mathbf{p}(x_1, \ldots, x_{t+1}, Z_{t+1} = i) \\
&= \sum_j \mathbf{p}(x_1, \ldots, x_{t+1}, Z_t = j, Z_{t+1} = i) \\
&\quad \text{(exhaust all states of } Z_t \text{ in the previous } t\text{)} \\
&= \sum_j \mathbf{p}(\mathbf{x}_{1:t}, Z_t = j)\mathbf{p}(Z_{t+1} = i | Z_t = j)\mathbf{p}(x_{t+1} | Z_{t+1} = i) \\
&= \sum_j \alpha_t(j)A(j, i)B(i, x_{t+1})
\end{aligned}$$

Backward Probability

- For the last time point $t = n$, $\beta_n(i) = \mathbf{p}(\mathbf{x}_{n+1}|Z_n = i)$, but we don't have $\mathbf{x}_{n+1}$ (no information). Hence, to not inject any artificial information, we should let

$$\beta_n(i) = 1$$

meaning that we must observe $\mathbf{x}_{n+1}$ anyway at the last step.

Backward Probability

- Then we recursively calculate $\beta_{t-1}(i)$ in the previous state:

$$
\begin{aligned}
\beta_{t-1}(i) &= \mathbf{p}(x_t, \ldots, x_n | Z_{t-1} = i) \\
&= \sum_j \mathbf{p}(x_t, \ldots, x_n, Z_t = j | Z_{t-1} = i) \\
&\quad \text{(exhaust all states of } Z_t \text{ in the next } t\text{)} \\
&= \sum_j \mathbf{p}(Z_t = j | Z_{t-1} = i) \mathbf{p}(x_t | Z_t = j) \mathbf{p}(\mathbf{x}_{t+1:n} | Z_t = j) \\
&= \sum_j A(i,j) B(j, x_t) \beta_t(j)
\end{aligned}
$$

Forward-Backward Algorithm

- Finally, the conditional probability $\gamma_t(i,j)$ needs to be normalized by the marginal probability to be a proper distribution:

$$\gamma_t(i,j) = \frac{\alpha_t(i)A(i,j)B(j,x_{t+1})\beta_{t+1}(j)}{\sum_i \sum_j \alpha_t(i)A(i,j)B(j,x_{t+1})\beta_{t+1}(j)}$$

- Note that this is a joint probability, so we divide by the sum of all cases.
- This concludes the calculation of $\gamma_t(i,j)$.

Forward-Backward Algorithm

- Similarly, we can calculate $\gamma_t(i)$ using Bayes' Theorem

$$\begin{aligned}\gamma_t(i) &= \mathbf{p}(Z_t = i|\mathbf{x}) \\ &\propto \mathbf{p}(Z_t = i, \mathbf{x}) \\ &= \mathbf{p}(Z_t = i, \mathbf{x}_{1:t})\mathbf{p}(\mathbf{x}_{t+1:n}|Z_t = i, \mathbf{x}_{1:t}) \\ &= \mathbf{p}(Z_t = i, \mathbf{x}_{1:t})\mathbf{p}(\mathbf{x}_{t+1:n}|Z_t = i) \\ &= \alpha_t(i)\beta_t(i)\end{aligned}$$

- Hence, after normalization, this is a proper distribution

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_i \alpha_t(i)\beta_t(i)}$$

- This concludes all the components needed in the EM algorithm.

Inference on the Hidden States $\mathbf{Z}$

- There are several ways to find the "optimal" hidden state sequence $\mathbf{Z}$ given an observation sequence $\mathbf{x}$, depending on the definition of "optimality".
- One criterion is to choose the hidden states Z_t's that are individually most likely at the time point t, that is

$$\widehat{Z}_t^* = \arg\max_i \mathbf{p}(Z_t = i|\mathbf{x}, \widehat{\theta}) = \arg\max_i \gamma_t(i).$$

- Here we can plug in $\widehat{\theta}$ from the aforementioned EM algorithm and simply get the argmax.

Inference on the Hidden States $\mathbf{Z}$

- Such a solution is optimal in the sense that it maximizes the expected number of correct states (by choosing the most likely state for each t).
- However, the resulting sequence may not be the most likely one and it may not even be a valid sequence. For example, if Z has three states (instead of 2), and the transition probability $A(1, 3) = 0$ (impossible to transit from 1 to 3), but it is still possible to have

$$\widehat{Z}_t^* = 1 \quad \text{and} \quad \widehat{Z}_{t+1}^* = 3$$

Inference on the Hidden States $\mathbb{Z}$

- An alternative approach is to find the most likely single sequence, i.e.,

$$\widehat{\mathbf{Z}}^{*} = \arg\max_{\mathbf{z}_{1:n}} \mathbf{p}(\mathbf{Z} = \mathbf{z}|\mathbf{x}, \widehat{\theta}).$$

- Note that maximizing this is the same as maximizing

$$\mathbf{p}(\mathbf{Z} = \mathbf{z}, \mathbf{x}|\widehat{\theta}) \quad \text{w.r.t} \quad \mathbf{z}_{1:n}$$

- Solving this with brute force will cost 2^n number of tries, which is almost impossible. Hence, we need some sophisticated algorithm. A dynamic programming method called Viterbi algorithm was proposed for this.
- An example of dynamic programming: Fibonacci series

Inference on the Hidden States $\mathbb{Z}$

- Define a maximizing sequence $z_{1:t}$ up to time t, which ends with $z_t = i$. The associated probability is

$$\mu_t(i) = \max_{\mathbf{z}_{1:(t-1)}} \mathbf{p}(\mathbf{Z}_{1:(t-1)} = \mathbf{z}_{1:(t-1)}, Z_t = i, \mathbf{x}_{1:t} | \widehat{\theta})$$

  which is the highest probability along a single path from 1 to t that ends up at state $Z_t = i$, given the observed sequence and the parameter estimation.

Inference on the Hidden States $\mathbb{Z}$

- Realizing that (with $\widehat{\theta}$ omitted)

$$
\begin{aligned}
&\mu_{t+1}(i) \\
&= \max_{\mathbf{z}_{1:t}} \mathbf{p}(\mathbf{z}_{1:t}, Z_{t+1} = i, \mathbf{x}_{1:(t+1)} | \widehat{\theta}\,) \\
&= \max_{\mathbf{z}_{1:t}} \Big\{ \mathbf{p}(\mathbf{z}_{1:(t-1)}, Z_t = z_t, \mathbf{x}_{1:t} | \widehat{\theta}\,) \mathbf{p}(z_{t+1} | z_t, \widehat{\theta}\,) \mathbf{p}(x_{t+1} | z_{t+1}, \widehat{\theta}\,) \Big\} \\
&= \Big\{ \max_{j} \mu_t(j) A(j, i) \Big\} B(i, x_{t+1})
\end{aligned}
$$

- By induction, we can solve the entire sequence.