EE365: Hidden Markov Models

Hidden Markov Models

The Viterbi Algorithm

Hidden Markov Models

Hidden Markov models

$$x_{t+1} = f_t(x_t, w_t)$$
$$y_t = h_t(x_t, z_t)$$

- called a *hidden Markov model* or HMM
- the states of the Markov Chain are not measurable (hence *hidden*)
- instead, we see $y_0, y_1, \ldots$
- y_t is a *noisy measurement* of x_t
- many applications: bioinformatics, communications, recognition of speech, handwriting, and gestures

Hidden Markov models

$$
\begin{aligned}
x_{t+1} &= f_t(x_t, w_t) \\
y_t &= h_t(x_t, z_t)
\end{aligned}
$$

- $x_0, w_0, w_1, \ldots, z_0, z_1, \ldots$ are independent
- hence the *state sequence* $x_0, x_1, \ldots$ is Markov
- w_t is *process noise* or *disturbance*
- z_t is *measurement noise*

Hidden Markov Models

order the variables as

$$x_0, y_0, x_1, y_1, x_2, \ldots$$

and apply the chain rule

$$\begin{aligned}
\mathbf{Prob}(y_t, x_t, \ldots, y_0, x_0) = & \, \mathbf{Prob}(y_t \mid x_t, y_{t-1}, x_{t-1}, \ldots, y_0, x_0) \\
& \, \mathbf{Prob}(x_t \mid y_{t-1}, x_{t-1}, \ldots, y_0, x_0) \\
& \, \mathbf{Prob}(y_{t-1} \mid x_{t-1}, y_{t-2}, x_{t-2}, \ldots, y_0, x_0) \\
& \qquad \vdots \\
& \, \mathbf{Prob}(x_1 \mid y_0, x_0) \\
& \, \mathbf{Prob}(y_0 \mid x_0) \\
& \, \mathbf{Prob}(x_0)
\end{aligned}$$

Hidden Markov Models

then we have

$$\mathbf{Prob}(y_0, \dots, y_t, x_0, \dots, x_t)$$
$$= Q_t(x_t, y_t) P_t(x_{t-1}, x_t) Q_{t-1}(x_{t-1}, y_{t-1}) \dots Q_0(x_0, y_0) \pi(x_0)$$

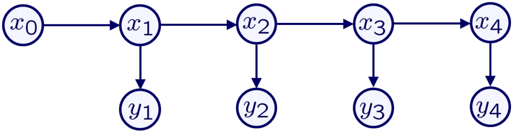
- $Q_t(x_t, y_t) = \mathbf{Prob}(y_t \mid x_t) = \mathbf{Prob}(y_t \mid x_t, y_{t-1}, x_{t-1}, \dots, y_0, x_0)$
- $P_t(x_{t-1}, x_t) = \mathbf{Prob}(x_t \mid x_{t-1}) = \mathbf{Prob}(x_t \mid y_{t-1}, x_{t-1}, \dots, y_0, x_0)$
- $\pi(x_0) = \mathbf{Prob}(x_0)$

Time-invariant case

$$
\begin{aligned}
x_{t+1} &= f(x_t, w_t) \\
y_t &= h(x_t, z_t)
\end{aligned}
$$

- $x_0, x_1, \ldots \in \mathcal{X}$ is a Markov chain with
  - transition probabilities $P_{ij} = \mathbf{Prob}(x_{t+1} = j \mid x_t = i)$
  - initial distribution $\pi_j = \mathbf{Prob}(x_0 = j)$
- $y_0, y_1, \ldots \in \mathcal{Y}$ is a set of measurements related to x_t by conditional probabilities $Q_{ik} = \mathbf{Prob}(y_t = k \mid x_t = i)$

Hidden Markov Model



- $x_t \in \mathcal{X} = \{1, 2, \ldots, n\}$ are the hidden states
- $y_t \in \mathcal{Y} = \{1, 2, \ldots, r\}$ are the measurements
- to specify a (time-invariant) HMM we only need
  - state transtion matrix $P_{ij} = \mathbf{Prob}(x_{t+1} = j \mid x_t = i)$
  - observation transition matrix $Q_{ik} = \mathbf{Prob}(y_t = k \mid x_t = i)$
  - initial state distribution $\pi_j = \mathbf{Prob}(x_0 = j)$
- can construct these from f, h

The Viterbi Algorithm

Maximum a posteriori state estimation

- time interval $[0, T]$
- we don't know the state sequence $x_0, \dots, x_T$, but we do know the measurements $y_0, \dots, y_T$ (and the probabilities P_{ij}, π_j, Q_{ik})
- so we will estimate $x_0, \dots, x_T$ based on the measurements $y_0, \dots, y_T$
- we would like to find the maximum a posteriori (MAP) estimate of $x_0, \dots, x_T$, denoted $\hat{x}_0, \dots, \hat{x}_T$, maximizes $\mathbf{Prob}(x_0, \dots, x_T \mid y_0, \dots, y_T)$
- in other words, find the *most likely* sequence of states given the measurements
- n^{T+1} possible sequences, so brute force consideration of all paths intractable

Maximum a posteriori state estimation

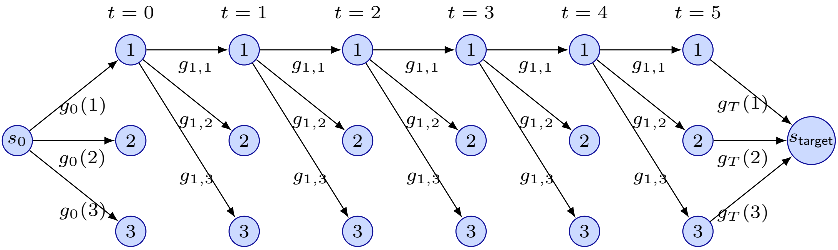
- same as maximizing (over $x_0, \ldots, x_T$)

$$
\begin{aligned}
&\mathbf{Prob}(x_0, \ldots, x_T)\, \mathbf{Prob}(y_0, \ldots, y_T \mid x_0, \ldots, x_T) \\
&\qquad = \left(\mathbf{Prob}(x_0) \prod_{t=0}^{T-1} \mathbf{Prob}(x_{t+1} \mid x_t) \right) \left(\prod_{t=0}^{T} \mathbf{Prob}(y_t \mid x_t) \right) \\
&\qquad = \pi_{x_0} \left(\prod_{t=0}^{T-1} P_{x_t, x_{t+1}} Q_{x_t, y_t} \right) Q_{x_T, y_T}
\end{aligned}
$$

- equivalently, minimize the negative logarithm

$$
\boxed{-\log \pi_{x_0} - \sum_{t=0}^{T-1} \log(P_{x_t, x_{t+1}} Q_{x_t, y_t}) - \log Q_{x_T, y_T}}
$$

MAP Markov state estimation a shortest path problem



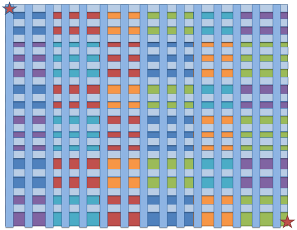
- vertices $x_t \in \mathcal{X} \times \{0, 1, \ldots, T\}$
- two additional vertices $\{s_0, s_{\text{target}}\}$
- edge cost $g(x_t, x_{t+1}) = -\log(P_{x_t, x_{t+1}} Q_{x_t, y_t})$
- edges $x_T \to s_{\text{target}}$ have terminal cost $g_T(x_T) = -\log Q_{x_T, y_T}$
- edges $s_0 \to x_0$ have initial cost $g_0(x_0) = -\log \pi_{x_0}$

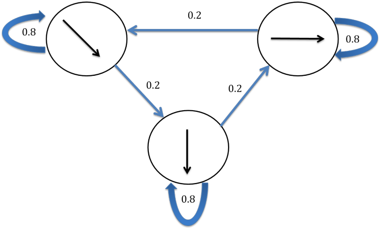
Viterbi algorithm

an efficient method for MAP estimation of Markov state sequence:

- use Bellman-Ford to find shortest path from s_0 to s_{target}
- the resulting sequence of states is the MAP estimate

Example: Grid sensors





- intruder starts from top-left corner
- direction of motion determined by state of Markov chain
- 40 by 40 grid, $d = 3$ directions, $|\mathcal{X}| = 4800$ possible states
- laser sensors detect crossing odd rows and columns
- 20 vertical sensors, 20 horizontal sensors, 441 possible measurements
- sensors detect intruder with probability 0.3

Example: Grid sensors

dynamics are

$$x_{t+1} = \phi(x_t + d(m_t))$$
$$m_{t+1} = (m_t + w_t) \text{ mod } 3$$

- directions are $d(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $d(1) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $d(2) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
- $w_0, w_1, \ldots$ are IID Bernoulli with $\mathbf{Prob}(w_t = 1) = 0.2$
- $\phi(x)$ clips components of x to $[1, 40]$

Example: Grid sensors

measurements are

$$y_{i,t} = \begin{cases} \lfloor (x_{i,t}+1)/2 \rfloor & \text{if } z_{i,t}=1 \text{ and } x_{i,t} \bmod 2 = 1 \\ 21 & \text{otherwise} \end{cases}$$

- at each time t we measure y_1 and y_2, functions of horizontal and vertical vehicle coordinates x_1 and x_2
- $z_{1,t}$ and $z_{2,t}$ are IID Bernoulli sequences with $\mathbf{Prob}(z_{i,t}=1)=0.3$

Viterbi updates

the Viterbi algorithm is

$$v_0(x) = -\log \pi_x \text{ for all } x$$

$$\textbf{for } t = 0, \ldots, T-1$$

$$\mu_t(x) = \operatorname*{argmin}_u \big(v_t(u) - \log(P_{ux} Q_{u,y_t})\big)$$
$$v_{t+1}(x) = \min_u \big(v_t(u) - \log(P_{ux} Q_{u,y_t})\big)$$

- at every step

$$v_t(x_t) - \log Q_{x_t,y_t} = -\log\big(\mathbf{Prob}(x_0, \ldots, x_t)\,\mathbf{Prob}(y_0, \ldots, y_t \mid x_0, \ldots, x_t)\big)$$

- the x_t that maximizes this quantity is $\hat{x}_t$, the MAP estimate given $y_0, \ldots, y_t$
- $\mu_t(x_{t+1})$ is the parent vertex of x_{t+1} along the shortest path

Viterbi computation

- simple implementation:
  - measure $y_0, \ldots, y_t$
  - compute $v_0, \ldots, v_t$
  - compute $\hat{x}_t$ by maximizing $v_t(x_t) - \log Q_{x_t, y_t}$
  - follow parent links: $\hat{x}_s = \mu_s(\hat{x}_{s+1})$ for $s = t-1, \ldots, 0$
- gives MAP estimate $\hat{x}_0, \ldots, \hat{x}_t$

Viterbi updates

- at time t, to compute $\hat{x}_0, \cdots \hat{x}_t$ we need v_t and $\mu_0, \ldots, \mu_{t-1}$
- these do not change over time, so we can reuse them at the next time-step
- this gives an *on-line* version of the Viterbi algorithm; at each time t
  - measure y_t
  - compute v_{t+1} and μ_t from v_t and y_t
  - compute $\hat{x}_t$ by maximizing $v_t(x_t) - \log Q_{x_t, y_t}$
  - follow parent links to find $\hat{x}_s = \mu_s(\hat{x}_{s+1})$ for $s = t-1, \ldots, 0$