Machine learning: lecture 17

Tommi S. Jaakkola
MIT AI Lab
tommi@ai.mit.edu

Topics

- Hidden markov models
  - posterior probabilities over states
  - the EM algorithm
  - viterbi (dynamic programming)

Hidden Markov models: review

A hidden Markov model (HMM) is model where we generate a sequence of outputs in addition to the Markov state sequence

$$
\begin{array}{ccccccc}
s_0 & \rightarrow & s_1 & \rightarrow & s_2 & \rightarrow & \ldots \\
\downarrow & & \downarrow & & \downarrow & & \\
\mathbf{x}_0 & & \mathbf{x}_1 & & \mathbf{x}_2 & &
\end{array}
$$

- To fully specify an HMM, we need to know
  1. the number of states m
  2. the initial state distribution $P_0(s_0)$
  3. the hidden state transition probabilities $P_1(s_{t+1}|s_t)$
  4. the output distribution $P_o(\mathbf{x}_t|s_t)$ (discrete or continuous)

HMM problems: review

- There are several problems we have to solve

1. How do we evaluate the probability that our model generated the observation sequence $\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\}$?
   - forward-backward algorithm
2. How do we uncover the most likely hidden state sequence corresponding to these observations?
   - dynamic programming
3. How do we adapt the parameters of the HMM to better account for the observations?
   - the EM-algorithm

Recursive computation: review

$$\begin{array}{ccccccccc} s_0 & \longrightarrow & s_1 & \longrightarrow & s_2 & \longrightarrow \ldots \longrightarrow & s_{n-1} & \longrightarrow & s_n \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ \mathbf{x}_0 & & \mathbf{x}_1 & & \mathbf{x}_2 & & \mathbf{x}_{n-1} & & \mathbf{x}_n \end{array}$$

- Forward (predictive) probabilities $\alpha_t(i)$:

$$\begin{aligned} \alpha_t(i) &= P(\mathbf{x}_0, \ldots, \mathbf{x}_t, s_t = i) \\ \frac{\alpha_t(i)}{\sum_j \alpha_t(j)} &= P(s_t = i | \mathbf{x}_0, \ldots, \mathbf{x}_t) \end{aligned}$$

- Backward (diagnostic) propabilities $\beta_t(i)$:

$$\beta_t(i) = P(\mathbf{x}_{t+1}, \ldots, \mathbf{x}_n | s_t = i)$$

(evidence about the current state from future observations)

Recursive computation: review

$$\begin{array}{ccccccccc} s_0 & \longrightarrow & s_1 & \longrightarrow & s_2 & \longrightarrow \ldots \longrightarrow & s_{n-1} & \longrightarrow & s_n \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ \mathbf{x}_0 & & \mathbf{x}_1 & & \mathbf{x}_2 & & \mathbf{x}_{n-1} & & \mathbf{x}_n \end{array}$$

Forward recursion:

$$\begin{aligned} \alpha_0(i) &= P_0(s_0 = i)\, P_o(\mathbf{x}_0 | s_0 = i) \\ \alpha_t(i) &= \sum_j \alpha_{t-1}(j)\, P_1(s_t = i | s_{t-1} = j) P_o(\mathbf{x}_t | s_t = i) \end{aligned}$$

Backward recursion:

$$\begin{aligned} \beta_n(i) &= 1 \\ \beta_{t-1}(i) &= \sum_j P_1(s_t = j | s_{t-1} = i) P_o(\mathbf{x}_t | s_t = j)\, \beta_t(j) \end{aligned}$$

Uses of forward/backward probabilities

- Complementary forward/backward probabilities

$$\begin{aligned}\alpha_t(i) &= P(\mathbf{x}_0, \ldots, \mathbf{x}_t, s_t = i) \\ \beta_t(i) &= P(\mathbf{x}_{t+1}, \ldots, \mathbf{x}_n | s_t = i)\end{aligned}$$

permit us to evaluate various (posterior) probabilities

First, we can evaluate the probability of the observation sequence:

$$\begin{aligned}P(\mathbf{x}_0, \ldots, \mathbf{x}_n) = \sum_i P(\mathbf{x}_0, \ldots, \mathbf{x}_n, s_t = i) \\ &= \sum_i P(\mathbf{x}_0, \ldots, \mathbf{x}_t, s_t = i) P(\mathbf{x}_{t+1}, \ldots, \mathbf{x}_n | s_t = i) \\ &= \sum_i \alpha_t(i) \beta_t(i)\end{aligned}$$

Forward/backward probabilities cont'd

$$
\begin{array}{ccccccccccc}
s_0 & \longrightarrow \ldots \longrightarrow & s_t & \longrightarrow & s_{t+1} & \longrightarrow \ldots \longrightarrow & s_n \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
\mathbf{x}_0 & & \mathbf{x}_t & & \mathbf{x}_{t+1} & & \mathbf{x}_n
\end{array}
$$

- We can evaluate the posterior probability that the HMM was in a particular state i at time t

$$
\begin{aligned}
P(s_t = i | \mathbf{x}_0, \ldots, \mathbf{x}_n) &= \frac{P(\mathbf{x}_0, \ldots, \mathbf{x}_n, s_t = i)}{P(\mathbf{x}_0, \ldots, \mathbf{x}_n)} \\
&= \frac{\alpha_t(i)\beta_t(i)}{\sum_j \alpha_t(j)\beta_t(j)} \stackrel{def}{=} \gamma_t(i)
\end{aligned}
$$

Forward/backward probabilities cont'd

$$
\begin{array}{ccccccccccc}
s_0 & \rightarrow \ldots \rightarrow & s_t & \rightarrow & s_{t+1} & \rightarrow \ldots \rightarrow & s_n \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
\mathbf{x}_0 & & \mathbf{x}_t & & \mathbf{x}_{t+1} & & \mathbf{x}_n
\end{array}
$$

- We can also compute the posterior probability that the system was in state i at time t AND transitioned to state j at time $t+1$:

$$
\begin{aligned}
&P(s_t = i, s_{t+1} = j|\mathbf{x}_0, \ldots, \mathbf{x}_n) \\
&\quad = \frac{\alpha_t(i) \overbrace{P_1(s_{t+1} = j|s_t = i) P_o(\mathbf{x}_{t+1}|s_{t+1} = j)}^{\text{fixed } i \rightarrow j \text{ transition, one observation}} \beta_{t+1}(j)}{\sum_j \alpha_t(j)\beta_t(j)} \\
&\quad \stackrel{def}{=} \xi_t(i,j),
\end{aligned}
$$

where $t = 0, \ldots, n-1$.

The EM algorithm for HMMs

Assume we have L observation sequences $\mathbf{x}_0^{(l)}, \ldots, \mathbf{x}_{n_l}^{(l)}$

E-step: compute the posterior probabilities

$$
\begin{aligned}
\gamma_t^{(l)}(i) & \quad \text{for all } l,\ i, \text{ and } t\ (t = 0, \ldots, n_l) \\
\xi_t^{(l)}(i,j) & \quad \text{for all } l,\ i, \text{ and } t\ (t = 0, \ldots, n_l - 1)
\end{aligned}
$$

M-step: First, the initial state distribution can be updated according to the expected fraction of times the sequences started from a specific state i

$$
\hat{P}_0(i) \;\leftarrow\; \frac{1}{L} \sum_{l=1}^{L} \gamma_0^{(l)}(i)
$$

M-step cont'd

Second, to update the transtion probabilities, we first define the expected number of transitions from i to j

$$\hat{n}(i,j) \;=\; \sum_{l=1}^{L} \sum_{t=0}^{n-1} \xi_t^{(l)}(i,j)$$

- The maximum likelihood estimate of the one step transition probabilities can be obtained by normalization

$$\hat{P}_1(j|i) \;\leftarrow\; \frac{\hat{n}(i,j)}{\sum_{j'} \hat{n}(i,j')}$$

M-step cont'd

- Third, if the outputs are discrete, we define the expected number of times a particular observations say $\mathbf{x} = k$ was generated from a specific state i

$$\hat{n}_o(i,k) = \sum_{l=1}^{L} \sum_{t=0}^{n_l} \gamma_t^{(l)}(i)\, \delta(\mathbf{x}_t^{(l)}, k)$$

The ML estimate is again obtained by normalization

$$\hat{P}_o(k|i) \;\; \leftarrow \;\; \frac{\hat{n}_o(i,k)}{\sum_{k'} \hat{n}_o(i,k')}$$

M-step cont'd

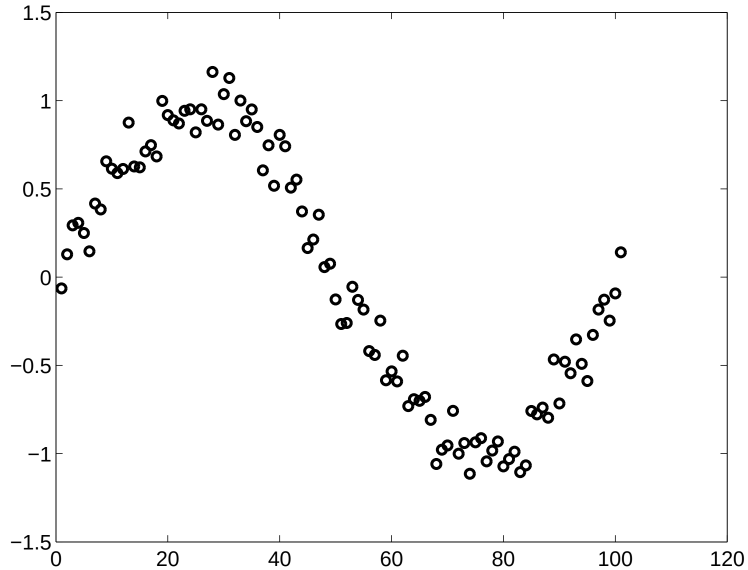
- If the outputs are continuous (e.g., multi-variate Gaussian), we have to solve a weighted maximum likelihood estimation problem as in the mixture of Gaussians models

  Separately for each state i we maximize:

$$J(\theta_i) = \sum_{l=1}^{L} \sum_{t=0}^{n_l} \gamma_t^{(l)}(i) \log P(\mathbf{x}_t^{(l)}|\theta_i)$$

  with respect to the parameters θ_i (e.g, the mean and the covariance).

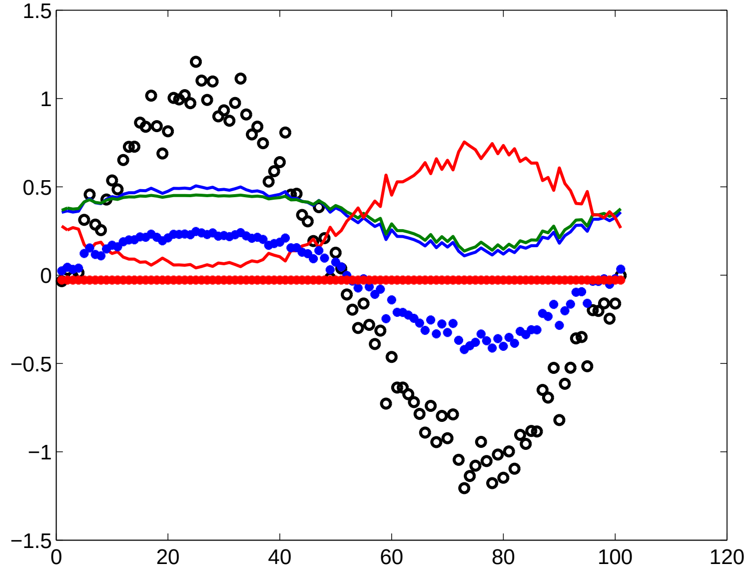
HMM example



Observed output as a function of time

- We will try to model this with a 3-state HMM with Gaussian outputs $p(x|s = i) = p(x|\mu_i, \sigma_i^2)$, $i = 1, 2, 3$.

HMM example cont'd



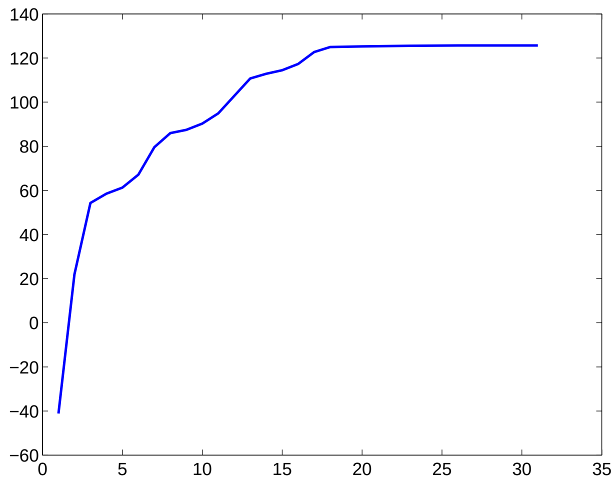
prior/posterior means and $\gamma_t(\cdot)$

$$\text{prior mean}(t) = \sum_i P_t(i)\hat{\mu}_i \quad ('*')$$

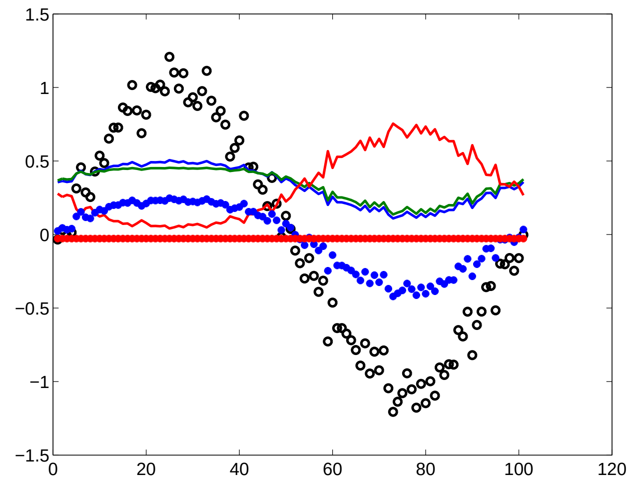
$$\text{posterior mean}(t) = \sum_i \gamma_t(i)\hat{\mu}_i \quad ('*')$$

where $P_t(i)$ is the probability of being in state i after t steps without observations; $\hat{\mu}_i$ is the mean output from the i^{th} state

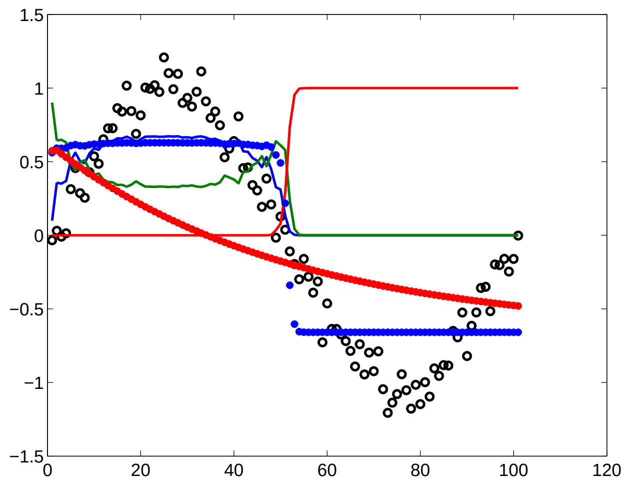
HMM example cont'd



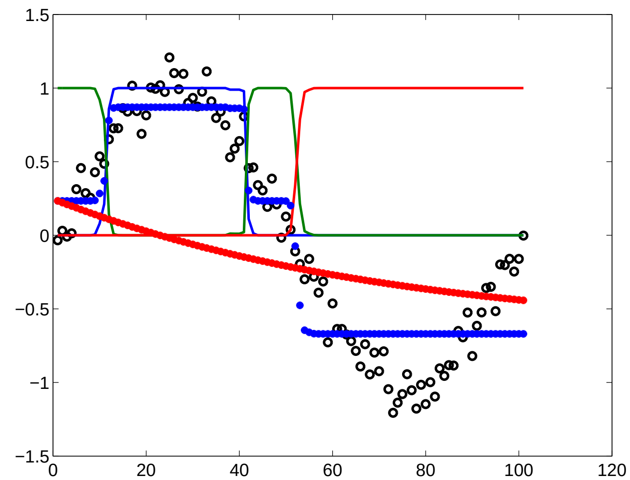
Log-prob. of data



after 0 iterations

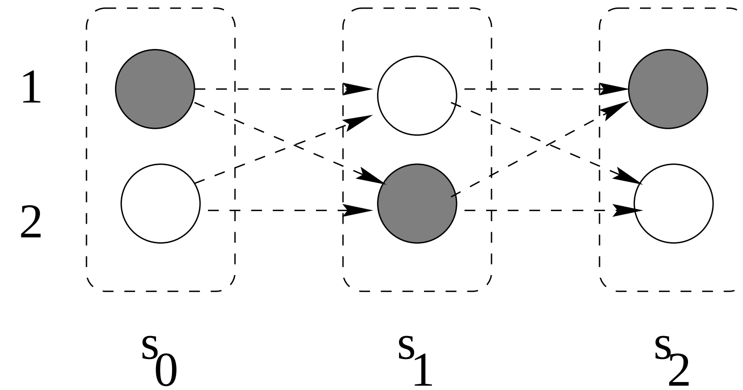


after 7 iterations



final

Dynamic programming (Viterbi)

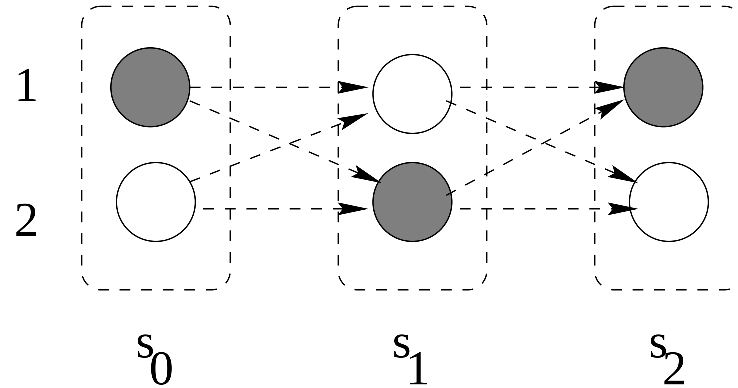


$$\mathbf{x}_0 = heads,\ \mathbf{x}_1 = tails,\ \mathbf{x}_2 = heads$$

- The probability of generating a particular hidden state sequence $s_0 = 1,\ s_1 = 2,\ s_2 = 1$ and the observations is

$$\begin{array}{ccccc}
P_0(1)P_o(heads|1) & \times & P_1(2|1)P_o(tails|2) & \times & P_1(1|2)P_o(heads|1 \\
\longrightarrow\ s_0 & & \longrightarrow\ s_1 & & \longrightarrow\ s_2 \\
\downarrow & & \downarrow & & \downarrow \\
\mathbf{x}_0 & & \mathbf{x}_1 & & \mathbf{x}_2
\end{array}$$

Dynamic programming (Viterbi)



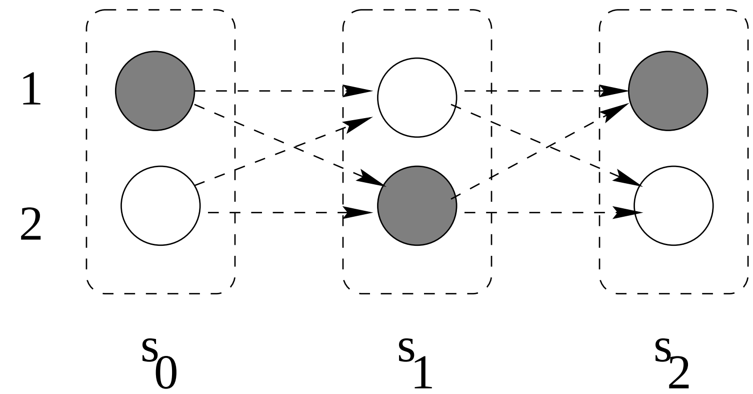
$$\mathbf{x}_0 = heads,\ \mathbf{x}_1 = tails,\ \mathbf{x}_2 = heads$$

- The probability of the most likely (partial) state sequence and the corresponding observations:

$$\delta_t(i) = \max_{s_0,\ldots,s_{t-1}} \left\{ P(\mathbf{x}_0,\ldots,\mathbf{x}_{t-1},s_0,\ldots,s_{t-1}) \right\} P_o(\mathbf{x}_t|s_t=i)$$

$$= \max_{s_0,\ldots,s_{t-1}} \left\{ P(s_0)P_o(\mathbf{x}_0|s_0)\cdots P_1(s_t=i|s_{t-1}) \right\} P_o(\mathbf{x}_t|s_t=i)$$

Dynamic programming (Viterbi)

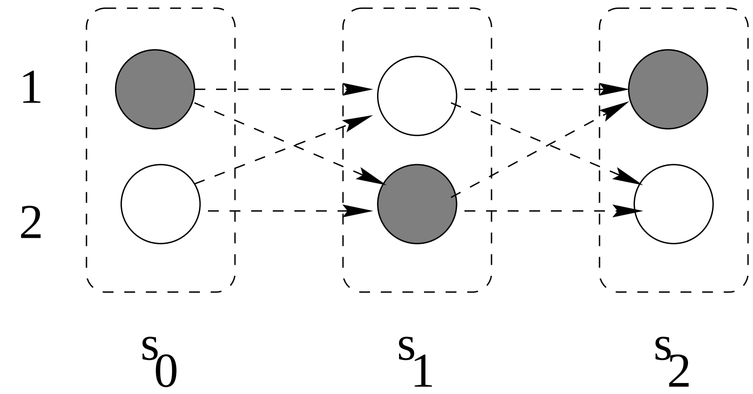


$$\mathbf{x}_0 = heads,\ \mathbf{x}_1 = tails,\ \mathbf{x}_2 = heads$$

- Recursive updates (same as forward probabilities but "sum" replaced with "max")

$$\begin{aligned}
\delta_0(j) &= P_0(j)P_o(heads|j), \quad j = 1,2 \\
\delta_1(1) &= \max\{\, \delta_0(1)P_1(1|1),\ \delta_0(2)P_1(1|2)\,\} \times P_o(tails|1) \\
\delta_1(2) &= \max\{\, \delta_0(1)P_1(2|1),\ \delta_0(2)P_1(2|2)\,\} \times P_o(tails|2) \\
&\ldots
\end{aligned}$$

Dynamic programming: backtracking



- The most likely value for state s_2 is the one that corresponds to the most likely path

$$s_2^* = \arg\max \{ \, \delta_2(1), \; \delta_2(2) \, \}$$

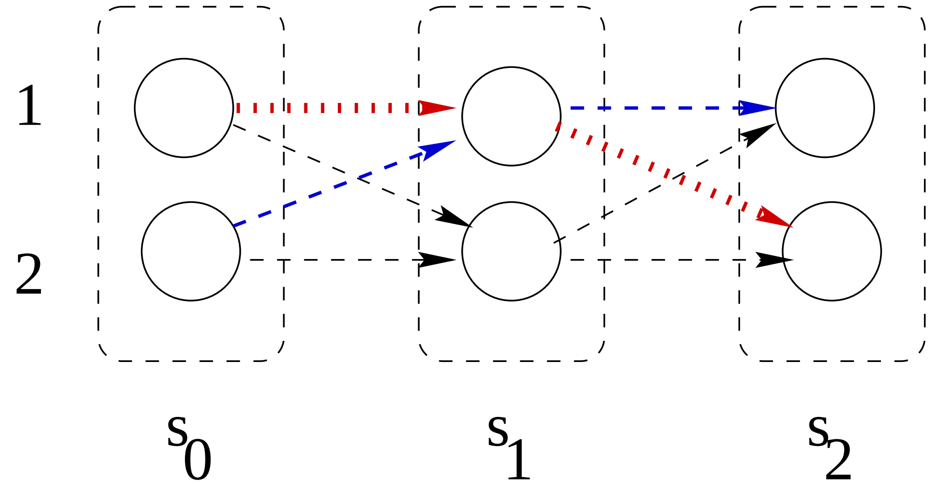
(say $s_2^* = 1$ as in the figure)

- The most likely previous state is

$$s_1^* \;\; = \;\; \arg\max \{ \, \delta_1(1) P_1(1|1), \; \delta_1(2) P_1(1|2) \, \}$$

and so on... (what about observations?)

Dynamic programming: properties



Red path (dotted): most likely path landing on $s_2 = 2$
Blue path (dashed): most likely path landing on $s_2 = 1$

- Possible?