Intelligent Systems (IS Fall 2013)

Assignment 3: Hidden Markov Models

Student: Fang–Lin He

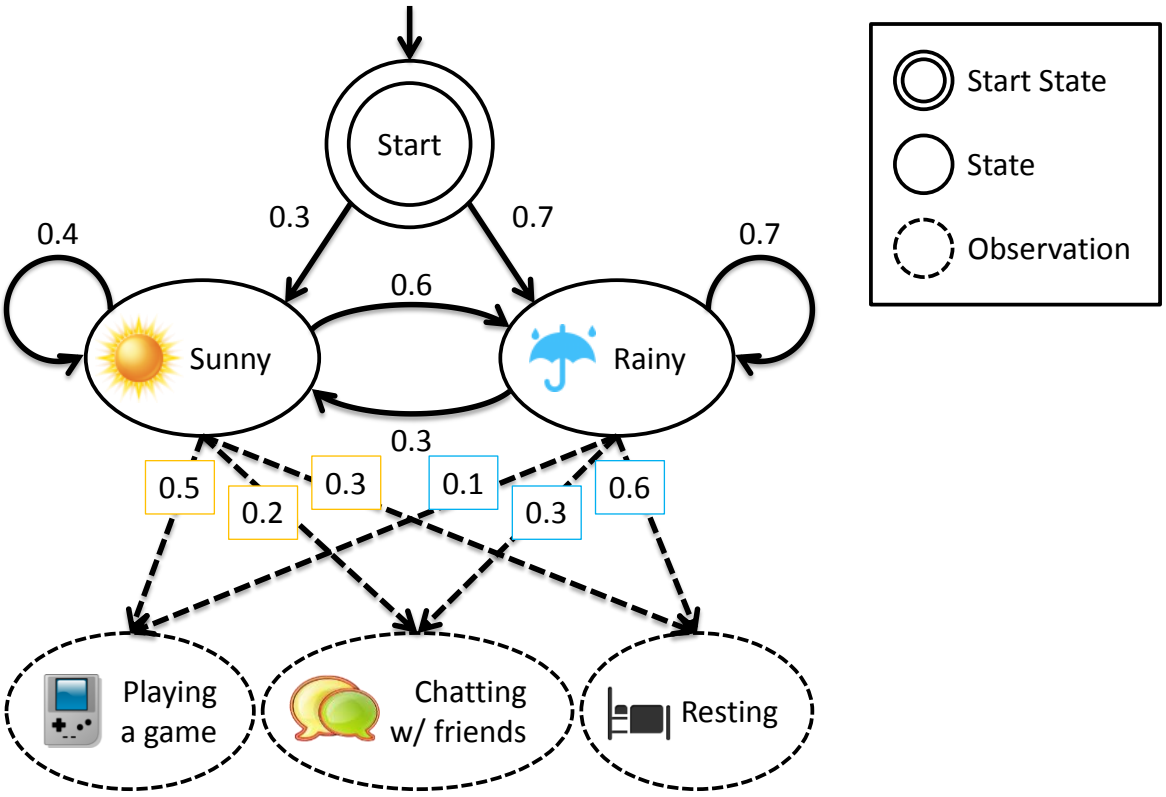
☺ Hi Klaus, to save your time, I marked my answer to the question with yellow background. The rest without yellow background is just supplementary explanation which you may want to kip or just skim.

---

1. (50 points) In recent weeks the weather in Lugano was either sunny or rainy, 70% of the time rainy. The chance it being sunny tomorrow if today is sunny is 40%, and the chance of having a sunny day after a rainy day is 30%. On weekends John divides his time among three activities: playing a game, chatting with friends and resting. His choice depends on the weather. If it is sunny, 50% of the time he plays a game, 20% chats with friends and 30% rests, otherwise, if it is rainy, 10% play a game, 30% chats with friends, 60% rests. Peter, John's friend, knows about his habits, and from talking to him, tries to guess the weather of Lugano based on what John did from Friday to Sunday.

(a) (10 points) Show what Peter's Hidden Markov Model of John's weekend looks like, and provide all probabilities he needs to define the model.

>> The below figure is Peter's HMM of John's weekend looks like.



(b) (10 points) What is (according to the HMM model) the probability of John playing a game on Friday, chatting with friends on Saturday, and chatting with friends on Sunday? What is the "stupid way" of calculating this? What is the clever one?

>> First, for convenience, *sunny weather* is denoted by s_1, *rainy weather* is denoted by s_2; *John playing a game* is denoted by *G*, *chatting with friends* is denoted by *C*, and *resting* is denoted by *R*. The observations are denoted by O_t, where $t \in \{1, 2, 3\}$ represents day one (Friday), day two (Saturday), and day three (Sunday), respectively; $O_t \in \{G, C, R\}$ represents three activities respectively.

Thus, the observed activities are $O_1 = G$, $O_2 = C$, and $O_3 = C$. The probability of John's activities can be written as $P(\mathbf{O}) = P(O_1 = G \wedge O_2 = C \wedge O_3 = C)$. The state (weather) on day *i* is denoted by $q_i \in \{s_1, s_2\}$; the path of states (i.e. the weather on Friday, Saturday, and Sunday) can be thus denoted by $\mathbf{Q} = q_1\, q_2\, q_3$,

According to the slides page 38, the stupid way is to first find out all possible paths (all possible weather on Friday, Saturday, and Sunday) **Q** (e.g. $\mathbf{Q} = s_1\, s_2\, s_1$), and for each path, calculate the probability of the observations $\mathbf{O} = G\ C\ C$, and finally sum them up. To write it as a formula:

$$P(\mathbf{O}) = \sum_{\mathbf{Q} \in \text{Paths of length 3}} P(\mathbf{O} \wedge \mathbf{Q})$$

$$= \sum_{\mathbf{Q} \in \text{Paths of length 3}} P(\mathbf{O}|\mathbf{Q}) P(\mathbf{Q})$$

So, all possible **Q**'s are: { $s_1\, s_1\, s_1$, $s_1\, s_1\, s_2$, $s_1\, s_2\, s_1$, $s_1\, s_2\, s_2$, $s_2\, s_1\, s_1$, $s_2\, s_1\, s_2$, $s_2\, s_2\, s_1$, $s_2\, s_2\, s_2$ }. $P(\mathbf{Q}) = P(q_1)\, P(q_2 \mid q_1)\, P(q_3 \mid q_2)$ and $P(\mathbf{O} \mid \mathbf{Q}) = P(O_1 \mid q_1)\, P(O_2 \mid q_2)\, P(O_3 \mid q_3)$. According to the graph: $P(s_1) = 0.3$, $P(s_2) = 0.7$, $P(s_1 \mid s_1) = 0.4$, $P(s_2 \mid s_1) = 0.6$, $P(s_1 \mid s_2) = 0.3$, $P(s_2 \mid s_2) = 0.7$; $P(G \mid s_1) = 0.5$, $P(C \mid s_1) = 0.2$, $P(R \mid s_1) = 0.3$, $P(G \mid s_2) = 0.1$, $P(C \mid s_2) = 0.3$, $P(R \mid s_2) = 0.6$. Therefore, I can fill the following table:

$q_1\, q_2\, q_3$	$s_1\, s_1\, s_1$	$s_1\, s_1\, s_2$	$s_1\, s_2\, s_1$	$s_1\, s_2\, s_2$	$s_2\, s_1\, s_1$	$s_2\, s_1\, s_2$	$s_2\, s_2\, s_1$	$s_2\, s_2\, s_2$
$P(\mathbf{Q})$	0.3*0.4*0.4 = 0.048	0.3*0.4*0.6 = 0.072	0.3*0.6*0.3 = 0.054	0.3*0.6*0.7 = 0.126	0.7*0.3*0.4 = 0.084	0.7*0.3*0.6 = 0.126	0.7*0.7*0.3 = 0.147	0.7*0.7*0.7 = 0.343
$P(\mathbf{O} \mid \mathbf{Q})$	0.5*0.2*0.2 = 0.02	0.5*0.2*0.3 = 0.03	0.5*0.3*0.2 = 0.03	0.5*0.3*0.3 = 0.045	0.1*0.2*0.2 = 0.004	0.1*0.2*0.3 = 0.006	0.1*0.3*0.2 = 0.006	0.1*0.3*0.3 = 0.009
$P(\mathbf{O} \wedge \mathbf{Q})$	0.00096	0.00216	0.00162	0.00567	0.000336	0.000756	0.000882	0.003087

Summing $P(\mathbf{O} \wedge \mathbf{Q})$ up, $P(\mathbf{O}) = 0.015471$.

According to the above calculations, it needs 8 $P(\mathbf{Q})$ computations and 8 $P(\mathbf{O} \mid \mathbf{Q})$ computations, where $8 = 2^3 = (\#\text{ states})^{(\#\text{ activities})}$, just the same as mentioned in the slide page 41. The stupid way takes too much computation (complexity = $O(N^M)$).

The smarter way is to use dynamic programming. By defining $\alpha_t(i) = P(O_1\ O_2\ \ldots\ O_t \wedge q_t = s_i \mid \lambda)$ where $1 <= t <= T$, $T = 3$ in this case, we can calculate such probabilities from the beginning ($t = 1$), save the values, and go to next state (t += 1) until the end. After that, to obtain $P(\mathbf{Q})$, we simply sum up $\alpha_t(i)$ with all states *i*. To do

the calculation, we first compute the initial probabilities: $\alpha_1(i) = P(q_1 = s_i)\, P(O_1 \mid q_1 = s_i)$, $i = 1, 2$, so $\alpha_1(1) = 0.3 * 0.5 = 0.15$, and $\alpha_1(2) = 0.7 * 0.1 = 0.07$. With initial probabilities, we can therefore calculate $\alpha_{t+1}(j) = \sum_{i=1}^{N} a_{ij} b_j(O_{t+1}) \alpha_t(i)$, where $N = 2$ is the number of observations, $a_{ij} = P(q_{t+1} = s_j \mid q_t = s_i)$ is the state transition probabilities ($a_{11} = 0.4$, $a_{12} = 0.6$, $a_{21} = 0.3$, $a_{22} = 0.7$), and $b_j(O_{t+1}) = P(O_{t+1} \mid q_t = s_i)$ is the observation probabilities ($b_1(G) = 0.5$, $b_1(C) = 0.2$, $b_1(R) = 0.3$, $b_2(G) = 0.1$, $b_2(C) = 0.3$, $b_2(R) = 0.6$). Thus:

$$\alpha_2(1) = \sum_{i=1}^{2} a_{i1} b_1(O_2) \alpha_1(i)$$
$$= a_{11} b_1(C) \alpha_1(1) + a_{21} b_1(C) \alpha_1(2)$$
$$= 0.4 * 0.2 * 0.15 + 0.3 * 0.2 * 0.07$$
$$= 0.0162$$

$$\alpha_2(2) = \sum_{i=1}^{2} a_{i2} b_2(O_2) \alpha_1(i)$$
$$= a_{12} b_2(C) \alpha_1(1) + a_{22} b_2(C) \alpha_1(2)$$
$$= 0.6 * 0.3 * 0.15 + 0.7 * 0.3 * 0.07$$
$$= 0.0417$$

$$\alpha_3(1) = \sum_{i=1}^{2} a_{i1} b_1(O_3) \alpha_2(i)$$
$$= a_{11} b_1(C) \alpha_2(1) + a_{21} b_1(C) \alpha_2(2)$$
$$= 0.4 * 0.2 * 0.0162 + 0.3 * 0.2 * 0.0417$$
$$= 0.003798$$

$$\alpha_3(2) = \sum_{i=1}^{2} a_{i2} b_2(O_3) \alpha_2(i)$$
$$= a_{12} b_2(C) \alpha_2(1) + a_{22} b_2(C) \alpha_2(2)$$
$$= 0.6 * 0.3 * 0.0162 + 0.7 * 0.3 * 0.0417$$
$$= 0.011673$$

So, in a smarter way to compute, $P(\mathbf{O}) = \sum_{i=1}^{2} \alpha_3(i) = 0.003798 + 0.011673 = 0.015471$, which only takes 6 computations (6 = 3 * 2 = (# observations) * (# states)), and the computational complexity is $O(N \cdot M)$. The much lower complexity is the reason to call it "*clever way*".

(c) (10 points) What would be Peter's guess about the weather in Lugano on Monday, if he knows that John played a game on Friday, chatted with friends on Saturday, and chatted with friends on Sunday? Calculate the probabilities using the forward algorithm. Is there a relation between this algorithm and the forward pass of a neural

network?

>> *Although I think my answer is wrong… just try to write down my opinion.* The weather in Lugano on Monday only depends on the weather on Sunday, so we can write this question as: argmax(j) $P(O_1\ O_2\ O_3\ \wedge\ q_4 = s_j)$, and by the follow calculations:

$$P(O_1O_2O_3\text{^}q_4 = s_j) = \sum_{i=1}^{2} P(O_1O_2O_3\text{^}q_3 = s_i\text{^}q_4 = s_j)$$

$$= \sum_{i=1}^{2} P(q_4 = s_j \mid O_1O_2O_3\text{^}q_3 = s_i)P(O_1O_2O_3\text{^}q_3 = s_i)$$

$$= \sum_{i=1}^{2} P(q_4 = s_j \mid q_3 = s_i)P(O_1O_2O_3\text{^}q_3 = s_i)$$

$$= \sum_{i=1}^{2} a_{ij}\alpha_3(i)$$

j = 1: $P(O_1\ O_2\ O_3\ \wedge\ q_4 = s_j)$ = 0.4 * 0.003798 + 0.3 * 0.011673 = 0.0050211

j = 2: $P(O_1\ O_2\ O_3\ \wedge\ q_4 = s_j)$ = 0.6 * 0.003798 + 0.7 * 0.011673 = 0.0104499

So argmax(j) $P(O_1\ O_2\ O_3\ \wedge\ q_4 = s_j)$ = 2, which means the weather in Lugano on Monday has higher probability (around 67.5 %) to be rainy.

This algorithm and the forward pass of a neural network both calculate something in order, from the beginning to the end, and store them. For this algorithm, it calculates the probable state for each time step (t=1, t=2, …); for a neural network, it calculates the activation of each node for each layer (first hidden layer, second hidden layer, …, output layer).

(d) (20 points) What is the most probable weather-sequence from Friday to Sunday given John's weekend activities from (c)? What algorithm do you use to calculate it? How is it different from the forward-algorithm and the forward-backward algorithm?

>> I use the *Viterbi algorithm* to calculate the most probable weather-sequence from Friday to Sunday. According to the formulas:

$$\left.\begin{aligned}\delta_{t+1}(j) &= \delta_t(i^*)\, a_{ij}\, b_j(O_{t+1}) \\ mpp_{t+1}(j) &= mpp_{t+1}(i^*)S_{i^*}\end{aligned}\right\} \quad i^* = \underset{i}{\operatorname{argmax}}\ \delta_t(i)\, a_{ij}\, b_j(O_{t+1})$$

$$\begin{aligned}\delta_1(i) &= \text{one choice } P(q_1 = S_i \wedge O_1) \\ &= P(q_1 = S_i)P(O_1 \mid q_1 = S_i) \\ &= \pi_i b_i(O_1)\end{aligned}$$

I build the table of $\delta_t(j)$ and then back-tracking to obtain mpp (red arrows):

$\delta_1(1) = 0.3 * 0.5 = 0.15$	$\delta_2(1) = \max(0.15 * 0.4, 0.07 * 0.3) * 0.2$ $= \max(0.06, 0.021) * 0.2$ $= 0.012$ $i^* = 1$	$\delta_3(1) = \max(0.012 * 0.4, 0.027 * 0.3) * 0.2$ $= \max(0.0048, 0.0081) * 0.2$ $= 0.00162$ $i^* = 2$
$\delta_1(2) = 0.7 * 0.1 = 0.07$	$\delta_2(2) = \max(0.15 * 0.6, 0.07 * 0.7) * 0.3$ $= \max(0.09, 0.049) * 0.3$ $= 0.027$ $i^* = 1$	$\delta_3(2) = \max(0.012 * 0.6, 0.027 * 0.7) * 0.3$ $= \max(0.0072, 0.0189) * 0.3$ $= 0.00567$ $i^* = 2$

Thus, the most probable weather-sequence is: Sunny → Rainy → Rainy.

The forward algorithm is to calculate the probability of a state at a certain time, given the history of observations: $P(q_k \mid O_1\ O_2\ \dots\ O_k)$. The backward algorithm is to calculate a set of backward probabilities which provide the probability of observing the remaining observations given any starting point (from Wiki): $P(O_{k+1}\ O_{k+2}\ \dots\ O_t \mid q_k)$. So the forward-backward algorithm is to combine forward and backward distributions to obtain the distribution over states at any specific point in time given the entire observation sequence: $P(q_k \mid O_1\ O_2\ \dots\ O_t) = P(q_k \mid O_1\ O_2\ \dots\ O_k, O_{k+1}\ O_{k+2}\ \dots\ O_t) \propto P(O_{k+1}\ O_{k+2}\ \dots\ O_t \mid q_k)\ P(q_k \mid O_1\ O_2\ \dots\ O_k)$. Forward-backward algorithm is also called smoothing which considers both forward and backward observations to calculate the distribution. The Viterbi algorithm is used to calculate the most probable sequence of hidden states given a sequence of observations. Thus, the most difference of Viterbi algorithm from forward and forward-backward algorithm is: the purpose of Viterbi algorithm is to find the "most probable" state sequence, and the rest is used to calculate the probability distributions.