CS 181 Assignment 4: Bayesian Networks and HMMs

Mark VanMiddlesworth & Shuang Wu

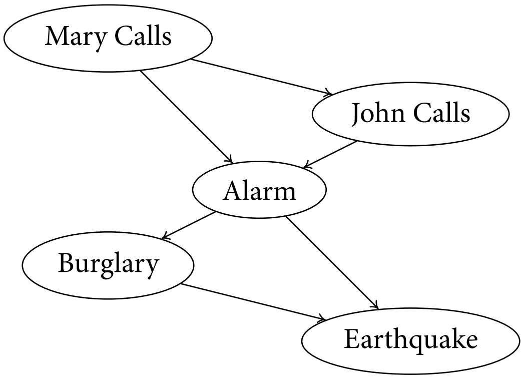
4/8/11

PROBLEM 1.

(a) We have the following table

Node Pair (U, V)	$S' \in S \setminus \{U, V\}, I(U, V \mid S')$
B, A	Impossible
B, E	$\{\}$
B, J	$\{A\}, \{E, A\}, \{A, M\}, \{E, A, M\}$
B, M	$\{A\}, \{E, A\}, \{A, J\}, \{E, A, J\}$
A, E	Impossible
A, J	Impossible
A, M	Impossible
E, J	$\{A\}, \{B, A\}, \{A, M\}, \{B, A, M\}$
E, M	$\{A\}, \{B, A\}, \{A, J\}, \{B, A, J\}$
J , M	$\{A\}, \{B, A\}, \{E, A\}, \{B, A, E\}$

(b) We have the following Bayesian Network constructed with variable order M, J, A, B, E



This is constructed as follows (referring to the dependence table above):

- We add node M.
- We add node J; since M and J are dependent given no other observations, we add M $\in$ Pa [J].
- We add node A; since A and M are dependent, we add M $\in$ Pa [A], and similarly J $\in$ Pa [A].
- We add node B; since B and A are dependent, we add A $\in$ Pa [B]. Since I (B, M | A), M $\notin$ Pa [B], and similarly for J.
- We add node E; since E and A are dependent, we add A $\in$ Pa [E]. Since B and E are not conditionally independent given any of the other variables, and the path between B and E doesn't have converging arrows at A, we must have a path directly between B and E, so we add B $\in$ Pa [E].

(c) The Bayesian Network in (a) has 10 parameters, while the Bayesian Network in (b) has 13. Bayesian

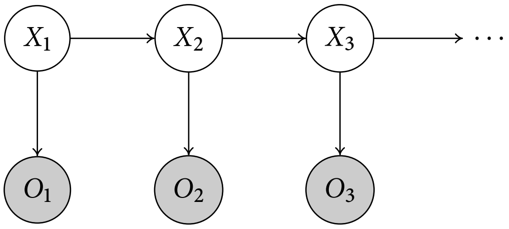
Network (a) is preferable because it encapsulates the information with fewer parameters, represents the natural causal ordering, and also yields more conditional independence properties (so that it is more fruitful to perform qualitative reasoning).

Problem 2.

Problem 3.

Problem 4.

(a) The HMM to model this weather problem looks like



where the X_i's form the unknown sequence of hidden states (which could represent a combination of things like temperature, pressure, etc.), and the O_i's form the sequence of observations (rain, clouds, sun, etc.). The model parameters necessary are the probabilities

$$\begin{aligned} P\left(X_t = s' \mid X_{t-1} = s\right) &= \theta_{s,s'}, \\ P\left(O_t = k \mid X_{t-1} = s\right) &= \theta_{k,s}, \\ P\left(X_1 = s\right) &= \theta_s^{(1)}, \end{aligned}$$

describing transitions between hidden states, observations given hidden states, and initial hidden states respectively. Altogether, with 4 possible observations (sun, clouds, fog, rain) and n hidden states, we have $(n-1)n + (4-1)n + n - 1$ parameters.

(b) (iii) The output from `test_bw_beta_all_equal` and from `test_bw_gamma_first_col_equal` is

```
beta:                                       gamma:
[  0.333333  0.333333  0.333333  ]          [  0.333333  0.500000  0.166667  ]
[  0.333333  0.333333  0.333333  ]          [  0.333333  0.500000  0.166667  ]
[  0.333333  0.333333  0.333333  ]          [  0.333333  0.500000  0.166667  ]
[  0.333333  0.333333  0.333333  ]          [  0.333333  0.500000  0.166667  ]
[  0.333333  0.333333  0.333333  ]          [  0.333333  0.166667  0.500000  ]
[  0.333333  0.333333  0.333333  ]          [  0.333333  0.500000  0.166667  ]
[  0.333333  0.333333  0.333333  ]          [  0.333333  0.166667  0.500000  ]
[  0.333333  0.333333  0.333333  ]          [  0.333333  0.166667  0.500000  ]
[  0.333333  0.333333  0.333333  ]          [  0.333333  0.166667  0.500000  ]
[  0.333333  0.333333  0.333333  ]          [  0.333333  0.166667  0.500000  ]
```

(1) The update rule for β is

$$\beta(x_t) = \begin{cases} \sum_{x_{t+1}} P\left(x_{t+1} \mid x_t\right) P\left(o_{t+1} \mid x_{t+1}\right) \beta\left(x_{t+1}\right), & \text{if } 1 \leq t < T, \\ 1, & \text{if } t = T. \end{cases}$$

Since transition probabilities are initialized to uniform and there are 3 hidden states, $P\left(x_{t+1} \mid x_t\right) = \frac{1}{3}, \forall t$. We also have $\sum_{x_{t+1}} P\left(o_{t+1} \mid x_{t+1}\right) = 1$. Now, we know that by definition $\beta(x_T) = 1$; using backward induction, supposing that $\beta(x_{t+1}) = 1$, we see that the update rule for β gives us $\beta(x_t) = 1 = \beta(x_{t+1})$, and hence all β's are equal for each time step.

(2) We know that γ is given by

$$\gamma_t^i(s) = P\left(x_t = s \mid o_1^i, \ldots, o_T^i\right).$$

Since we initialize $P(O_t = 1 \mid X_t = 1) = P(O_t = 2 \mid X_t = 2) = \frac{1}{2}$, O_t gives no information in the conditional probability $P\left(X_t = 1 \mid o_1^i, \ldots, o_T^i\right)$, so all gammas are equal for $X_t = 1$.

(iv) The update rule for normalized α is

$$\alpha(x_t) = \begin{cases} P(o_t \mid x_t) \sum_{x_{t-1}} P(x_t \mid x_{t=1}) \alpha(x_{t-1})/N_t, & \text{if } 1 < t \leq T, \\ P(o_1 \mid x_1) P(x_1)/N_t, & \text{if } t = 1, \end{cases}$$

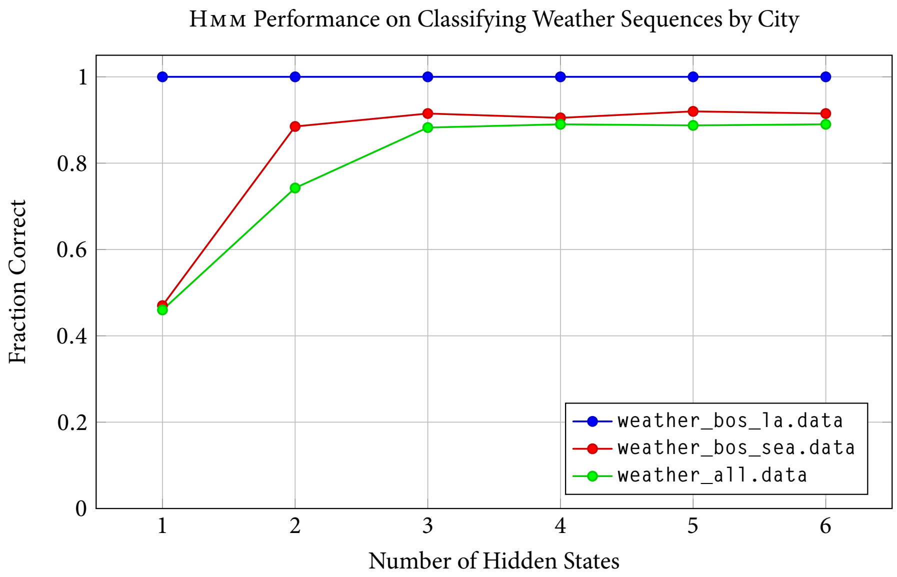
for $N_t = \sum_{x_t} \alpha(x_t)$. For a sequence of three observations, we have

$$\begin{aligned} P(o_1, o_2, o_3) &= \sum_{x_3} P(o_3 \mid x_3) \sum_{x_2} P(x_3 \mid x_2) P(o_2 \mid x_2) \sum_{x_1} P(o_1 \mid x_1) P(x_1) P(x_2 \mid x_1) \\ &= \sum_{x_3} P(o_3 \mid x_3) \sum_{x_2} P(x_3 \mid x_2) P(o_2 \mid x_2) \sum_{x_1} P(x_2 \mid x_1) \alpha(x_1) N_1 \\ &= \sum_{x_3} P(o_3 \mid x_3) \sum_{x_2} P(x_3 \mid x_2) \alpha(x_2) N_2 N_1 \\ &= \sum_{x_3} \alpha(x_3) N_3 N_2 N_1 \\ &= N_3 N_2 N_1 \end{aligned}$$

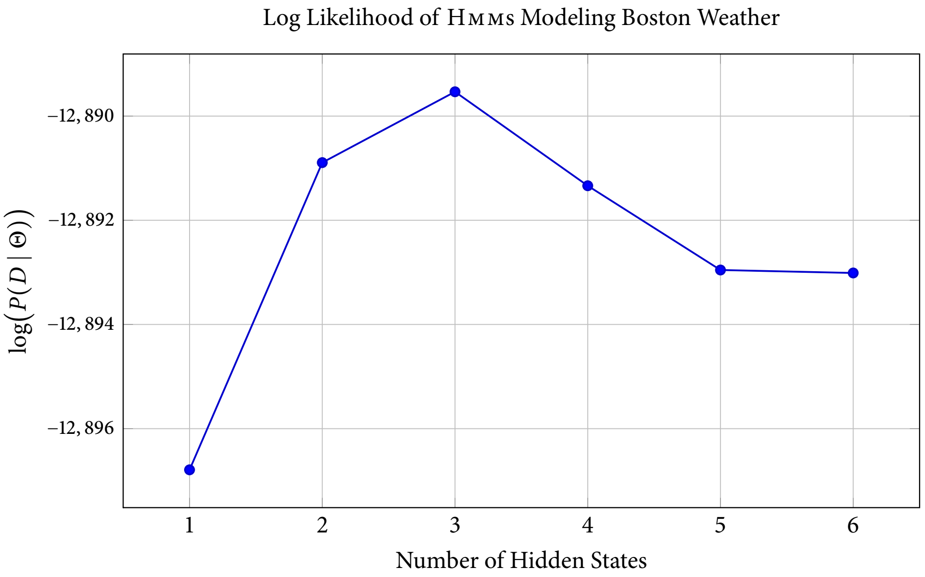
since the α's are all normalized. Generalizing, we see that the probability of the observation sequence is the product of all the normalization factors of the α's, so the sum of the logs of the normalization factors is the same as the log likelihood of the returned observation sequence.

(c) Note that in the following questions, we have enabled random restarts, and given the initial a random distribution on each run. The graphs show the best performing run out of six for each number of hidden states (this is cheating a little bit since we aren't reporting the results of separate validation set, but using random restarts in this way produces much smoother graphs than the deterministic method).

(i) We have the following plot



(ii) We have the following plot



The log likelihood of the data increases as we increase the number of hidden states from 1 to 3, after which it drops off.

(iii) We see from the learned models that for the Boston/LA classification, more hidden states are not necessary since a single hidden state is enough to perfectly classify all instances. Inspecting the data, we see that this is since Boston always has more rainy days while LA has more sunny days, so a correct prediction only needs to look at the proportion of rainy to sunny days.

We also see that increasing hidden states increases performance in the Boston/Seattle classification, and even more so in the classification of data from all four cities. This makes sense since by inspecting the data, we see that Boston and Seattle have weather that is more similar, so the classification is harder than Boston/LA and thus the first extra hidden state gives a drastic improvement over the single hidden state, while there are not any significant improvements after 2 hidden states. The dataset containing all of the cities is the hardest to classify, and this is reflected by its poorer performance and greater performance gains when using up to three hidden states, compared to the Boston/Seattle classification. Overall, we note that the performance for all three sets seems to level off at 3 hidden states.

More hidden states help when observation sequences are similar, but once we are able to identify differences between observation sequences, additional hidden states are unnecessary.

(iv) Since the log likelihood of the data peaks at 3 hidden states and the performance of the classifiers leveled off after 3 hidden states, we believe that the generating model had 3 hidden states.