CS 181 Assignment 4: Bayesian Networks and HMMs

Mark VanMiddlesworth & Shuang Wu

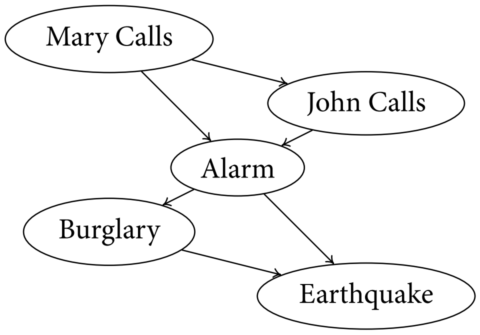
4/8/11

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 $I(B, M \mid A)$, 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.

(a) (i) We want to compute $P(\mathsf{OT} \mid \mathsf{S} = T, \mathsf{SC} = F, \mathsf{ST} = T)$ and $P(\mathsf{OT} \mid \mathsf{S} = F, \mathsf{SC} = F, \mathsf{ST} = T)$. It is sufficient to calculate the joint probability and treat the $P(\mathsf{S}, \mathsf{SC}, \mathsf{ST})$ as a normalizing factor (note that we don't calculate $P(\mathsf{S})$, since it has probability 1 once the driver makes the decision). We have

$$\begin{aligned}
P(\mathsf{OT}, \mathsf{S}, \mathsf{SC} = F, \mathsf{ST} = T) &= \sum_{x \in \mathsf{C}} \sum_{y \in \mathsf{T}} P(\mathsf{C} = x) P(\mathsf{SC} = F \mid \mathsf{C} = x) P(\mathsf{ST} = T \mid \mathsf{C} = x) \\
&\qquad P(\mathsf{T} = y \mid \mathsf{C} = x, \mathsf{S}) P(\mathsf{OT} \mid \mathsf{T} = y, \mathsf{S}, \mathsf{ST} = T) \\
&= \sum_{x \in \mathsf{C}} P(\mathsf{C} = x) P(\mathsf{SC} = F \mid \mathsf{C} = x) P(\mathsf{ST} = T \mid \mathsf{C} = x) \\
&\qquad \sum_{y \in \mathsf{T}} P(\mathsf{T} = y \mid \mathsf{C} = x, \mathsf{S}) P(\mathsf{OT} \mid \mathsf{T} = y, \mathsf{S}, \mathsf{ST} = T) \\
&= \sum_{x \in \mathsf{C}} P(\mathsf{C} = x) P(\mathsf{SC} = F \mid \mathsf{C} = x) P(\mathsf{ST} = T \mid \mathsf{C} = x) g_y(x, \mathsf{OT}),
\end{aligned}$$

where

$$g_y(x, \mathsf{OT}) = \sum_{y \in \mathsf{T}} P(\mathsf{T} = y \mid \mathsf{C} = x, \mathsf{S}) P(\mathsf{OT} \mid \mathsf{T} = y, \mathsf{S}, \mathsf{ST} = T).$$

Let us denote $g_y^{(T)}$ as the above for $\mathsf{S} = T$, and $g_y^{(F)}$ for $\mathsf{S} = F$. Let $f_1^{(T)}(\mathsf{T}, \mathsf{C}) = P(\mathsf{T} \mid \mathsf{C}, \mathsf{S} = T)$, $f_1^{(F)}(\mathsf{T}, \mathsf{C}) = P(\mathsf{T} \mid \mathsf{C}, \mathsf{S} = F)$, $f_2^{(T)}(\mathsf{OT}, \mathsf{T}) = P(\mathsf{OT} \mid \mathsf{T}, \mathsf{S} = T, \mathsf{ST} = T)$, and $f_2^{(F)}(\mathsf{OT}, \mathsf{T}) = P(\mathsf{OT} \mid \mathsf{T}, \mathsf{S} = F, \mathsf{ST} = T)$. We have

T	C	$f_1^{(T)}(\mathsf{T},\mathsf{C})$
T	T	0.5
T	F	0
F	T	0.5
F	F	1

T	C	$f_1^{(F)}(\mathsf{T},\mathsf{C})$
T	T	0
T	F	0
F	T	1
F	F	1

OT	T	$f_2^{(T)}(\mathsf{OT},\mathsf{T})$
T	T	0
T	F	0.9
F	T	1
F	F	0.1

OT	T	$f_2^{(F)}(\mathsf{OT},\mathsf{T})$
T	T	0
T	F	0.1
F	T	1
F	F	0.9

and the intermediate tables $h(\mathsf{T} = y, \mathsf{C} = x, \mathsf{OT} = z)$

T	C	OT	$h^{(T)}(\mathsf{T},\mathsf{C},\mathsf{OT})$
T	T	T	0
T	T	F	0.5
T	F	T	0
T	F	F	0
F	T	T	0.45
F	T	F	0.05
F	F	T	0.9
F	F	F	0.1

T	C	OT	$h^{(F)}(\mathsf{T},\mathsf{C},\mathsf{OT})$
T	T	T	0
T	T	F	0
T	F	T	0
T	F	F	0
F	T	T	0.1
F	T	F	0.9
F	F	T	0.1
F	F	F	0.9

which gives us the tables for $g_y(x, \mathsf{OT})$

C	OT	$g_y^{(T)}(\mathsf{C}, \mathsf{OT})$
T	T	0.45
T	F	0.55
F	T	0.9
F	F	0.1

C	OT	$g_y^{(F)}(\mathsf{C}, \mathsf{OT})$
T	T	0.1
T	F	0.9
F	T	0.1
F	F	0.9

We eliminate the next variable

$$\begin{aligned} P(\mathsf{OT}, \mathsf{S}, \mathsf{SC} = F, \mathsf{ST} = T) &= \sum_{x \in \mathsf{C}} P(\mathsf{C} = x) P(\mathsf{SC} = F \mid \mathsf{C} = x) P(\mathsf{ST} = T \mid \mathsf{C} = x) g_y(x, \mathsf{OT}) \\ &= g_x(\mathsf{OT}). \end{aligned}$$

We compute $f_1(\mathsf{C}) = P(\mathsf{C})$, $f_2(\mathsf{C}) = P(\mathsf{SC} = F \mid \mathsf{C})$, and $f_3(\mathsf{C}) = P(\mathsf{ST} = T \mid \mathsf{C})$

C	$f_1(\mathsf{C})$
T	0.1
F	0.9

C	$f_2(\mathsf{C})$
T	0.4
F	1

C	$f_3(\mathsf{C})$
T	0.8
F	0.3

Combined with our tables for g_y, we have the intermediate tables

C	OT	$h^{(T)}(\mathsf{C}, \mathsf{OT})$
T	T	0.0144
T	F	0.0176
F	T	0.243
F	F	0.027

C	OT	$h^{(F)}(\mathsf{C}, \mathsf{OT})$
T	T	0.0032
T	F	0.0288
F	T	0.027
F	F	0.243

which gives us the tables for $g_x(\mathsf{OT})$

OT	$g_x^{(T)}(\mathsf{OT})$
T	0.2574
F	0.0446

OT	$g_x^{(F)}(\mathsf{OT})$
T	0.0302
F	0.2718

Normalizing the probabilities for each case, we see that the probability of being late is 0.148 when the driver decides to speed, and 0.9 when the driver decides not to speed.

(ii) We want to compute $P(\mathsf{T} \mid \mathsf{S}, \mathsf{SC} = F, \mathsf{ST} = T)$. We proceed as in the previous problem

$$\begin{aligned} P(\mathsf{T}, \mathsf{S}, \mathsf{SC} = F, \mathsf{ST} = T) &= \sum_{x \in \mathsf{C}} P(\mathsf{C} = x) P(\mathsf{SC} = F \mid \mathsf{C} = x) P(\mathsf{ST} = T \mid \mathsf{C} = x) P(\mathsf{T} \mid \mathsf{C} = x, \mathsf{S}) \\ &\qquad \sum_{y \in \mathsf{OT}} P(\mathsf{OT} = y \mid \mathsf{T}, \mathsf{S}, \mathsf{ST} = T) \\ &= \sum_{x \in \mathsf{C}} P(\mathsf{C} = x) P(\mathsf{SC} = F \mid \mathsf{C} = x) P(\mathsf{ST} = T \mid \mathsf{C} = x) P(\mathsf{T} \mid \mathsf{C} = x, \mathsf{S}), \end{aligned}$$

since we are summing a single probability of OT over all its possible values. We then eliminate the second variable by writing

$$\begin{aligned} P(\mathsf{T}, \mathsf{S}, \mathsf{SC} = F, \mathsf{ST} = T) &= \sum_{x \in \mathsf{C}} P(\mathsf{C} = x) P(\mathsf{SC} = F \mid \mathsf{C} = x) P(\mathsf{ST} = T \mid \mathsf{C} = x) P(\mathsf{T} \mid \mathsf{C} = x, \mathsf{S}) \\ &= g_x(\mathsf{T}). \end{aligned}$$

We compute $f_1(\mathsf{C}) = P(\mathsf{C})$, $f_2(\mathsf{C}) = P(\mathsf{SC} = F \mid \mathsf{C})$, $f_3(\mathsf{C}) = P(\mathsf{ST} = T \mid \mathsf{C})$, and $f_4(\mathsf{C}, \mathsf{T}) = P(\mathsf{T} \mid \mathsf{C}, \mathsf{S})$

C	$f_1(\mathsf{C})$
T	0.1
F	0.9

C	$f_2(\mathsf{C})$
T	0.4
F	1

C	$f_3(\mathsf{C})$
T	0.8
F	0.3

C	T	$f_4^{(T)}(\mathsf{C}, \mathsf{T})$
T	T	0.5
T	F	0.5
F	T	0
F	F	1

C	T	$f_4^{(F)}(\mathsf{C}, \mathsf{T})$
T	T	0
T	F	1
F	T	0
F	F	1

Combining, we have the intermediate tables

C	T	$h^{(T)}(\mathsf{C}, \mathsf{T})$
T	T	0.016
T	F	0.016
F	T	0
F	F	0.27

C	T	$h^{(F)}(\mathsf{C}, \mathsf{T})$
T	T	0
T	F	0.032
F	T	0
F	F	0.27

which gives us the tables for $g_x(\mathsf{T})$

T	$g_x^{(T)}(\mathsf{T})$
T	0.016
F	0.286

T	$g_x^{(F)}(\mathsf{T})$
T	0
F	0.302

Hence we see that the probability of getting a ticket when the driver decides to speed is 0.053, and the driver never gets a ticket when he doesn't speed.

(b) The expected utility of speeding is

$$-150 \cdot 0.053 - 10 \cdot 0.148 = -9.43,$$

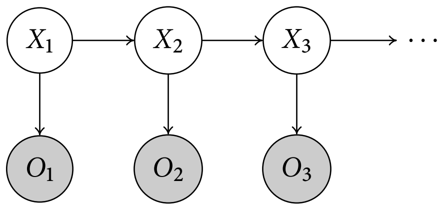
while the expected utility of not speeding is

$$-150 \cdot 0 - 10 \cdot 0.9 = -9.$$

Since the expected cost of speeding ($9.43) is greater than the expected cost of not speeding ($9.00), the optimal decision is to not speed.

PROBLEM 3.

(c) (i) The HMM to model the no-momentum case looks like



where the X_i's form the sequence of states (the sequence of locations of the robot in its world) and the O_i's form the sequence of observations (the sequence of perceived colors by the robot). 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 each location in the world, observations given a particular location, and initial positions respectively. Note that only the 12 valid (non-black) locations (`1:2`, `1:3`, `2:1`, `2:3`, `2:4`, `3:1`, `3:2`, `3:3`, `3:4`, `4:1`, `4:2`, `4:4`) are used since we build our set of states from the robot data. Since we have 4 possible observations (`r`, `g`, `b`, `y`), the total number of parameters for the model is $(12-1)12 + (4-1)12 + 12 - 1 = 179$.

(ii) `robot_no_momentum.data`: Learning our HMM on the `robot_no_momentum.data` dataset produced a performance of 81.2%. Using an HMM for this problem makes sense intuitively, since the hidden states correspond directly to locations on the board, and the relationship between hidden state and observation is established as sensor error. In using an HMM, we assume that the transition probabilities between any given pair of locations is constant and independent of previous transitions, and the transition probability of observing a particular color on a given square is also constant, which matches this problem exactly.

The model has trouble with ambiguous cases where the current location is adjacent to multiple squares of the color of the next observation. As an example, if we have an observation sequence of `y` $\rightarrow$ `r` $\rightarrow$ `b`, there are two paths (`3:4` $\rightarrow$ `2:4` $\rightarrow$ `2:3` and `3:4` $\rightarrow$ `3:3` $\rightarrow$ `2:3`) which are equally likely given only sensor readings, and it is impossible for any model to perform better than 50% in this case. Since this situation appears in multiple places in the robot world, this ambiguity likely contributes to a substantial portion of the total error.

`robot_with_momentum.data`: Learning our HMM on the `robot_no_momentum.data` dataset produced a performance of 86.7%, which is an improvement over the data with no momentum. This may be somewhat surprising at first because the HMM assumption that the transition probabilities between a pair of locations is constant is no longer true of the domain (it is still the case however that the probability of observing a color is constant given the square), since previous states contain information about the future states which cannot be fully captured in just the current position of the robot.

All is not lost however, since the board we are using contains black (inaccessible) squares, which inherently encode some information about previous states into each location. That is, since the world has boundaries and inaccessible squares, any given current square on this board has at least one neighboring direction which could not have been the direction of approach (and thus would carry no momentum in that direction, given the current square). As an example, consider the square `3:2`. In the case without momentum, the transition probabilities to each of the square's 3 possible neighbors and for staying in the same square is uniform and should be $\frac{1}{4}$; however, with momentum, the robot would never have momentum going to `4:2` (since `2:2` is inaccessible), some likelihood of having momentum going to `3:3` (there is one square from which the robot could have gained momentum moving in that direction), some likelihood of staying put, and a greater likelihood of having momentum going to `3:1` (since there are two squares from which the robot could have gained momentum in that direction). This property is generally consistent with the transition matrices generated in our experiments, and accounts for the improvement in performance since this additional information helps resolve the color ambiguities described in the no momentum data.

We can modify our HMM model in this case to better fit the data by allowing a larger state space, consisting of not just each location, but of each possible pair (location, momentum) for each of the four momentum directions and a fifth for no momentum. With this modification, the HMM assumptions would once again hold and we would expect performance to be better; however, it is also more computationally expensive to learn this model and we might require additional training data.

Problem 4.

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

$$\begin{tikzcd}
X_1 \arrow[r] \arrow[d] & X_2 \arrow[r] \arrow[d] & X_3 \arrow[r] \arrow[d] & \cdots \\
O_1 & O_2 & O_3 &
\end{tikzcd}$$

where the X_i's form the unknown sequence of hidden states (chosen from an unknown set which could represent a combination of things like temperature, pressure, etc.), and the O_i's form the sequence of observations (the observed weather types, chosen from the set of {sun, clouds, fog, rain}). 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, weather 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 $P\left(X_t = 1 \,\middle|\, o_1^i, \ldots, o_T^i\right)$, so all gammas are equal for $X_t = 1$.

(iv) The update rule for normalized α is

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

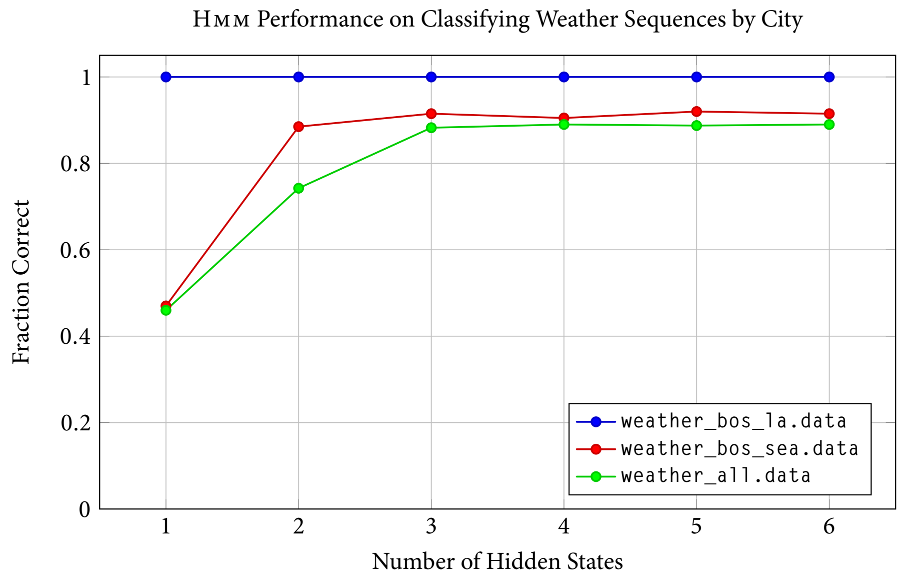
for $N_t = \sum_{x_t} \alpha(x_t)$. For a sequence of three observations, we have

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

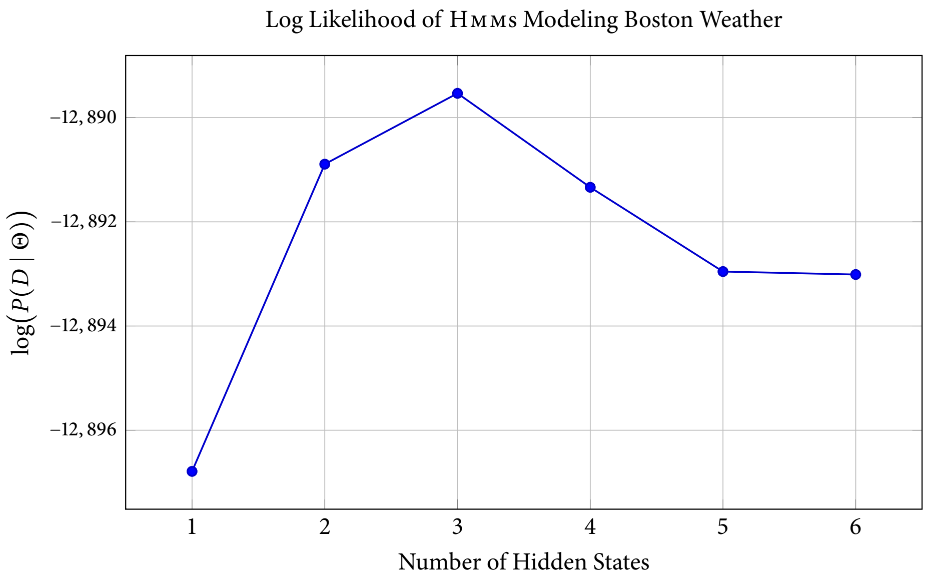
since the α's are all normalized. Generalizing, we see that the probability of the observation sequence is the product of all the normalization factors of the α's, so the sum of the logs of the normalization factors is the same as the log likelihood of the returned observation sequence.

(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.