COGS 118A: ASSIGNMENT 6

Problem 1. Hidden Markov model (40)

(a) (5) There is a formal equivalence between the order of a Markov model and the dimensionality of state representation. Suppose we have a second-order Markov model, where by $x_t = ax_{t-2} + bx_{t-1}$. Re-represent this as a first-order Markov model by letting the state variable be two dimensional, $x_t = [x_t, x_{t+1}]^T$.

(Solution)

\[
x_t = ax_{t-2} + bx_{t-1}
\]

since $y_t = [x_t, x_{t+1}]^T \Rightarrow y_t = Ay_{t-1}$, where $A = \begin{bmatrix} 0 & 1 \\ a & b \end{bmatrix}$

thus $p(y_t|y_{t-1},...y_1) = p(y_t|y_{t-1})$

(b) (5) Recall in a Markov model, the joint distribution of all the nodes can be written as a produce of conditionals:

\[
p(x_1,...,x_\tau) = p(x_1) \prod_{i=2}^{\tau} p(x_i|x_{i-1}) .
\]

Use this fact and Bayes’ rule to show that the distribution of a single node conditioned on all the other nodes only depends on its immediate neighbors:

\[
p(x_t|x_1,...,x_{t-1},x_{t+1},x_\tau) = p(x_t|x_{t-1},x_{t+1}) .
\]

In Bayesian graphical model terminology, $x_{t-1}$ and $x_{t+1}$ form the Markov blanket for $x_t$. By definition, a random variable $x_t$ is independent of everything else when conditioned on the nodes that form its Markov blanket, i.e. We say $X = \{x_1,\ldots,x_n\}$ is a Markov blanket for a random variable $z$, if $p(z|X, Y) = p(z|X)$ for any collection of random variables $Y = \{y_1,\ldots,y_m\}$.

(Solution)
Show that the Markov blanket of \( s \) be factorized as follows:

\[
p(x_t|x_1 \ldots x_{t-1}, x_{t+1}, \ldots, x_T) = \frac{p(x_1, \ldots, x_{t-1})p(x_t, \ldots, x_T|x_{t-1})}{p(x_1, \ldots, x_{t-1})p(x_{t+1}, \ldots, x_T|x_1, \ldots, x_{t-1})}
\]

since \( p(x_{t+1}, \ldots, x_T|x_1 \ldots x_{t-1}) = \sum_{x_t} p(x_{t+1}, \ldots, x_T|x_t)p(x_t|x_{t-1}) \)

\[
= p(x_{t+1}, \ldots, x_T)p(x_t|x_{t-1})
\]

\[
\Rightarrow p(x_t|x_1 \ldots t-1, x_{t+1}; T) = \frac{p(x_t, \ldots, x_T|x_{t-1})}{p(x_{t+1}, \ldots, x_T|x_{t-1})}
\]

where \( A = x_t, B = \ldots x_T, C = x_{t-1} \)

\[
P(A, B|C) = \frac{p(A, B|C)}{P(B|C)} = P(A|B, C)
\]

\[
= P(x_t|x_{t+1}, \ldots, x_T, x_{t-1})
\]

where \( A = x_t, E = x_{t+1}, D = \ldots x_T, C = x_{t-1} \)

\[
p(A|E, D, C) = \frac{p(D|A, E, C)p(A|E, C)}{p(D|E, C)}
\]

\[
= \frac{p(x_{t+2}, \ldots, x_T|x_t, x_{t+1}, x_{t-1})p(x_t|x_{t-1} x_{t+1})}{p(x_{t+2}, \ldots, x_T|x_{t+1}, x_{t-1})}
\]

\[
= p(x_t|x_{t-1}, x_{t+1})
\]

(c) (5) In a hidden Markov model, the hidden nodes are (typically first-order) Markov, while the observations only depend on their immediate parents. That is, the joint distribution can be factorized as follows:

\[
p(x_1, s_1, \ldots, x_n, s_n) = p(s_1)p(x_1|s_1)\prod_{t=2}^{T} p(s_t|s_{t-1})p(x_t|s_t)
\]

Show that the Markov blanket of \( s_t \) is \( \{s_{t-1}, s_{t+1}, x_t\} \), and the Markov blanket of \( x_t \) is \( \{s_t\} \).

(Solution)

claim 1: \( p(x_t|s_1, \ldots, s_T, x_1, \ldots x_{t-1}, x_{t+1}, \ldots x_T) = p(x_t|s_t) \)

proof: LHS = \[
\sum_{x_t} p(s_1, \ldots, s_T, x_1, \ldots x_T)
\]

\[
= \sum_{x_t} p(s_1)p(x_1|s_1)\prod_{i=2}^{T} p(s_i|s_{i-1})p(x_i|s_i)
\]

\[
= \sum_{x_t} p(x_t|s_t)
\]

\[
\to 1 = p(x_t|s_t) = \text{RHS}
\]

COG 118A Cogs 118A: Assignment 6 2 Winter 2010
(d) (5) In a hidden Markov model, the hidden states can be first-order Markov while the observations are not Markov of any order. Give an example of a generative model for which this is true.

(e) (5) Generate a sequence of 100 data points from the following hidden Markov model:

\[
p(s_1) = \text{Beta}(s_1; a = 1, b = 1)
\]
\[
p(s_t | s_1, \ldots, s_{t-1}) = \alpha \delta(s_t - s_{t-1}) + (1 - \alpha) \text{Beta}(s_t; a = 1, b = 1)
\]
\[
\mathbb{P}(x_t | s_t) = \text{Bern}(x_t; s_t)
\]

where \( \alpha = .9 \) and \( \delta(x) \) is the Dirac delta function, i.e.

\[
\delta(x) = \begin{cases} 
+\infty, & x = 0 \\
0, & x \neq 0 
\end{cases}
\]

and

\[
\int_{-\infty}^{\infty} \delta(x) dx = 1 \quad \int_{-\infty}^{\infty} f(x) \delta(x) dx = f(0) .
\]

In other words, the hidden variable is initially drawn from a (uniform) Beta distribution, then has probability \( \alpha = .9 \) of remaining the same on each time step, and probability \( 1 - \alpha = .1 \) of being redrawn randomly from the (uniform) Beta distribution. And the observed variable is drawn from a Bernoulli distribution that depends on the hidden variable as its rate parameter.

(Solution) refer to f for graph
(f) (15) For the sequence of data points generated in (e), compute the likelihood of the sequence for different values of $\alpha$. Assume it is known that $s$ is drawn from the uniform Beta distribution initially, and also every time when it is redrawn. In Matlab, let

$$\alpha = [0:.05:1];$$

and we also grid up the $s_t$ space, because there is no analytical form for the likelihood function (why?):

$$s = [0:.01:1];$$

Evaluate the likelihood function for each value of $\alpha$ numerically. Create a plot of this, and label the axes. What is the maximum likelihood estimate for $\alpha$?

(Solution)
Problem 2. Stochastic linear dynamical systems (state-space model, Kalman filter) (25)

(a) (5) We will consider a simple special case of a stochastic linear dynamical system when both the hidden and observed variables are scalar valued:

\[ y_1 \sim \mathcal{N}(0, \sigma_0^2) \]
\[ y_t = A y_{t-1} + w_t \]
\[ x_t = C y_t + v_t \]

where \( w_t \sim \mathcal{N}(0, q^2) \) and \( v_t \sim \mathcal{N}(0, r^2) \). Rewrite the Kalman filter equations (P. 10 of lecture notes from 02/09) for this special case.

(Solution)

Prediction: \( \hat{y}_{t-1} = A (\hat{y}_{t-1})^{t-1} \)
\[ \hat{y}_t = \hat{y}_{t-1} + k_t (x_t - C \hat{y}_{t-1}) \]
where \( k_t = C \hat{v}_{t-1} (C^2 \hat{v}_{t-1} + v_{t-1}) \)
\[ \hat{v}_1 = \sigma_o^2 \]
\[ \hat{v}_2 = A^2 \sigma_o^2 + q^2 \]
\[ k_2 = \hat{v}_2 \frac{C (C^2 \hat{v}_{t-1}^2 + v_{t-1})}{C^2 \hat{v}_{t-1}^2 + v_{t-1}} - 1 \]
\[ \hat{v}_{t-1} = A^2 \hat{v}_{t-1}^t + q^2 \]
\[ \hat{v}_t^t = \hat{v}_{t-1}^t - k_t \hat{v}_t^t \]

(b) (5) Assume \( \sigma_0 = 5, A = .5, C = 1, q = .05, \) and \( r = .2 \). Generate 4 sequences of 20 data points each from this state-space model. Use four separate subplots to show the trajectories of \( y_t \) in each case. Use hold to super-impose over \( y_t \) the corresponding trajectories of \( x_t \). Label the axis. Also describe in words the dynamics of the hidden variables \( y_t \).

(Solution)
(c) (15) Use the equations from (a) to compute the corrected posterior mean and variance of $y$ at each time point for each of the data sequences. On four subplots, plot the trajectories of the mean. Now add and subtract the standard deviation from the mean at each time step, and plot these two extra lines (in a different color) on the same figures. Label the axes. What do you notice about the evolution of mean and standard deviation over time? How does the estimated mean depend on observations, and how well does it match up to the true $y_t$ of (b)? In a separate figure, plot the Kalman gain as a function of time. Does it differ for the four data sequences?

(Solution)
Problem 3. Linear basis models of regression (35)

(a) (5) Given a linear basis function model for regression, \( \bar{y}(x) = w_0 + \sum_{j=1}^{m} w_j \phi_j(x) \), and additive Gaussian noise, \( y \sim N(\bar{y}, \sigma^2) \), show that the maximum likelihood estimate (MLE) for \( w_0 \) is:

\[
\hat{w}_0^* = \frac{1}{n} \sum_{i=1}^{n} y_i - \sum_{j=1}^{m} w_j \tilde{\phi}_j
\]

where \( \tilde{\phi}_j = \frac{1}{n} \sum_{i=1}^{n} \phi_j(x_i) \). That is, \( w_0 \) compensates for the difference between the sample mean of \( y \) and the weighted sample mean of the basis functions (each evaluated at all the data points).

(Solution)

\[
E_D(w) = \frac{1}{2} \left( \sum_{i=1}^{n} (y_i - w_0 - \sum_{j=1}^{m} w_j \phi_j(x_i))^2 \right)
\]

\[
\frac{\partial E_D(w)}{\partial w_0} = \sum_{i=1}^{n} (y_i - w_0 - \sum_{j=1}^{m} w_j \phi_j(x_i)) \cdot (-1) = 0
\]

\[\Rightarrow w_0^* = \frac{1}{n} \sum_{i=1}^{n} y_i - \sum_{j=1}^{m} w_j \phi_j(x_i)\]

(b) (5) Minimizing least squares error in regression leads to a nice analytical solution (Bishop Eq. 3.15). However, inverting a matrix can be computationally intense. Use \texttt{tic} and \texttt{toc} in Matlab to measure the amount of time it takes to invert a random matrix of size \( k \times k \) generated with \texttt{rand(k)}, for different values of \( k \). Start with some small value like 500 or 1000, and then increase it up to a point when Matlab hangs so long that you lose patience – note the value of \( k \) when this happens. Try at least 10 random matrices of each size, and average the computation time returned by \texttt{tic} and \texttt{toc}. Plot the mean duration against \( k \).

(Solution)
(c) (25) As we saw in class (and also Bishop 3.45-3.54), a Gaussian prior over $w$ leads to simple update rules for the Gaussian posterior. This process lends itself readily to sequential updating. We make the same assumptions here as in Problem 3, except that the prior is zero-mean and isotropic, $m_0 = 0, S_0 = \tau^{-2}I$, where $\tau^2 = .5$, and that we re-compute the posterior after each data pair $(x, y)$, to obtain $(m_1, S_1), (m_2, S_2), \ldots$. We also assume that the observation noise has variance $\sigma = .2$.

Now you will begin your Matlab coding, all of which will be with respect to the provided m file runBayesLinReg. Set your initial prior parameters as above. Plot this initial prior over $w$ using the provided function plotwdist. Sample six $w$ values (i.e. pairs $(w_0, w_1)$) from this prior\(^1\) and plot the lines $y(x, w) = w_0 + w_1 x$ that would correspond to those values using the provided function plotwsandfunc. With these initial steps complete, you will

- Generate a “true” data sample from the function $\tilde{y}(x; a) = a_0 + a_1 x$ with parameter values $a_0 = -0.3$ and $a_1 = 0.5$ by first choosing a value of $x$ from the uniform distribution $U(x| -1, 1)^2$, then evaluating $\tilde{y}(x; a)$, and finally adding Gaussian noise with standard deviation 0.2 to obtain the target value $y$.
- Plot the likelihood function of $w$ of this new data sample\(^3\).
- Update the distribution over $w$ by incorporating the new data sample.
- Plot the resulting posterior distribution over $w$ using the provided plotwdist function.
- Sample six $w$ values (i.e. pairs $(w_0, w_1)$) from the updated (posterior) distribution over $w$.
- Plot all the $(x, y)$ pairs observed so far, together with the lines $y(x, w) = w_0 + w_1 x$ that would correspond to the sampled $w$ values using the provided function plotwsandfunc.
- Begin again at the top of this list, repeating until 20 rounds of data samples are complete.

\(^1\)Consider the mvnrnd command.
\(^2\)Consider the unifrnd command.
\(^3\)Consider using normpdf.
The general outline needed to accomplish the above steps has already been written, as have several of the helper functions. Your specific task is to finish the job by providing the code for `samplewdist`, `samplefunc`, `plotwlik`, `make_phi_and_t`, and `updatewdist`. For more guidance, consult `runBayesLinReg.m` (Solution)