Applied Stochastic Processes, 18-751, TX Brown, Fall 2017
Homework #9

Due 8pm Monday October 30.

Read ch 9.1 and 9.2 in JW. On your own do Quizzes 12.1

1. The number of customers that arrive in a store during one hour has a Poisson distribution with mean 5. Each customer buys a number of lottery tickets, independently of other customers, and this number is Poisson distributed with mean 2. For this hour find:
   (a) The moment generating function, $\Theta(s)$, of the total number of tickets sold.
   (b) The probability that no tickets are sold (Hint: what happens to the MGF of a discrete non-negative r.v. when $s \to -\infty$?).
   (c) The expected number of tickets sold. The *standard deviation* of the number of tickets sold.
   (d) It takes one hour for the store to stock up on new lottery tickets. The store does not like to run out of tickets. What is the minimum number of tickets the store should have on hand to have a 99% chance of meeting the demand for the next hour.

2. You are building a learning system where you ask students questions and they give you answers. You want the system to identify when students are bored (B) and when they are engaged (E). You do not know directly when they are in states B or E. You can only see the rate that they answer questions correctly. When they are bored, they get answers correct 50% of the time. When they are engaged they get answers correct 90% of the time. Experience shows that once students are bored, they stay bored for awhile and when they are engaged they stay engaged for awhile. This experience shows the following Markov Chain model:

   Initial state probability for states B and E, $\pi_1 = (0.5, 0.5)$

Previous State, x_{m-1}	Next State, x_m	$P(x_m \mid x_{m-1})$
B	B	0.9
B	E	0.1
E	E	0.9
E	B	0.1

   (a) Generate N=10 steps of the Markov Chain states $x_1, x_2, \ldots x_N$. For each step one question is asked. Generate the sequence of observed right (R) and wrong (W) answers for the student $y_1, y_2, \ldots y_N$.
   (b) Create the trellis diagram for this problem. It should consist of states laid out in 10 columns with 2 states in each column similar to Fig. 9.3 in the text. Label one row B and the other E. Above each column, m, label it with the observable y_m. Now draw edges from one column to the next. Label each edge with a distance $d_m(x_{m-1}, x_m) = -\log(P(x_m|x_{m-1})P(y_m|x_m))$. Add one additional state, S, to the left of the trellis with one edge going to each of the states in the first column. Label this with weight $d_1(S, x_1) = -\log(\pi_1(x_1)P(y_1|x_1))$.
   (c) Find the shortest path through the trellis using the Bellman-Ford Algorithm as follows. Label the states in the last column with $V_N(x) = 0$. Starting with column $m = N - 1$ perform the following steps. For each state x in column m, label the state with $V_m(x) = \min_{x'}\{d_{m+1}(x, x') + V_{m+1}(x')\}$ and let the x' that minimizes this be the selected edge out of state x at step m. Continue this recursively to state S. Determine the lowest cost path. Write below your trellis the true states from the Markov Chain and your estimate of the true state from the Bellman-Ford algorithm.
   (d) Automate this procedure using Matlab code. Generate 50 states and in a stacked graph show the observed values, the true state, and the estimated state. How accurate is this method? How do errors typically occur (e.g. in long bursts, mostly at transitions, etc.)?

3. Consider a room temperature monitoring system. The system tracks both the interior and exterior temperatures, $x = (t^I, t^E)$, according to the following model. To be clear, there is a true model which is hidden from the Kalman filter. This is what actually happens in the environment and is what our sensors will measure. In a real system it would not be a set of equations as below but rather it would be the actual sensor measurements that would have a different profile every day. Then there is the model that we create for the Kalman filter. The Kalman model is what we use to track and predict the system state. In a typical system, this model is fixed and does not change from day to day. In fact, it might be a model developed at the time a system is designed in the factory and so may not vary from one deployed system to the next. Because weather changes from day to day and performance can vary from system to system, the two models are not identical and so the question is to see how the Kalman filter performs despite these differences.

   True Model (used to generate actual temperatures and sensor measurements):

$$\begin{aligned}
t_k^I &= t_{k-1}^I + 0.04(t_{k-1}^E - t_{k-1}^I) + w_k^I, \quad w_k^I \sim N(0, 0.03) && (1)\\
t_k^E &= t_{k-1}^E + 0.14\sin(\frac{2\pi k}{300}) + w_k^E, \quad w_k^E \sim N(0, 0.02) && (2)\\
z_k^I &= t^I k + v_k^I, v_k^I \sim N(0, 1.5) && (3)\\
z_k^E &= t^E k + v_k^E, v_k^E \sim N(0, 1.5) && (4)\\
B_k &= (-0.2, 0), u_k \in \{0, 1\} && (5)\\
x_0 &= (23, 20) && (6)
\end{aligned}$$

   Kalman Model (used to estimate inside and outside temperature):

$$\begin{aligned}
t_k^I &= t_{k-1}^I + 0.05(t_{k-1}^E - t_{k-1}^I) + w_k^I, \quad w_k^I \sim N(0, 0.04) && (7)\\
t_k^E &= t_{k-1}^E + 0.1\sin(\frac{2\pi k}{300}) + w_k^E, \quad w_k^E \sim N(0, 0.01) && (8)\\
z_k^I &= t^I k + v_k^I, v_k^I \sim N(0, 4) && (9)\\
z_k^E &= t^E k + v_k^E, v_k^E \sim N(0, 1) && (10)\\
B_k &= (-1, 0), u_k \in \{0, 1\} && (11)\\
x_0 &= (25, 25), P_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} && (12)
\end{aligned}$$

   Consider a sensor network where The timing of new measurements may be highly variable or bursty due to network errors or other faults. We model with the below Markov chain. Entering the state P means that time is advanced and there is a prediction step. Entering the state M means there is a new measurement and there should be a fusion step. Let $\alpha = 0.25, \beta = 0.5$ (if there are multiple measurements in a row, they should be processed in order and only the last estimate plotted.

   Simulate this system for 300 times steps. Time is only advanced on a prediction step. So, you will simulate more than 300 Markov Chain steps to simulate 300 time steps. At the end of each step compute the probability that the internal temperature is more than 28 degrees. If this exceeds 10%, then in the next step let $u_k = 1$ otherwise $u_k = 0$. To show the performance of your system make two graphs:

   (a) Plot the true model internal temperature, the Kalman filter internal temperature estimate, and $\pm$ one standard deviation error bars around the internal temperature estimate.

   (b) Plot on one graph, the actual external temperature, the actual internal temperature with the air conditioner, and the actual internal temperature if there is no air conditioner (e.g. rerun with $B_k = 0$). What is your opinion of the air conditioning?