STA 4103/5107: Midterm Project

(Wednesday, 02/15)
Due: noon, Thursday, 03/01

Choose one topic from the following two topics. Write one report on it.

Your report should include descriptions of:
(a) The Problem Statement,
(b) Methodology or Approach,
(c) Matlab programs,
(d) Main Results (using Tables or Figures),
(e) Summary.

Points are allocated towards presentation of results and clarity of your report.

Topic I: Bayesian Analysis of Noisy Images

1. **Goal:** Given observed noisy images, our goal is to perform a Bayesian analysis of this data. We will assume a prior probability model and an observation model to obtain a posterior density, and will generate samples from the posterior.

2. **Models:**
   (a) **Prior Model:** Let I be an $m \times n$ matrix of random variables such that they form a Markov Random Field (MRF). The conditional density of an element is dependent only on the values of its vertical and horizontal neighbors (except for the boundaries where the neighbors are limited). Let the conditional density of a pixel be Gaussian with mean μ and variance σ_1^2, where μ is the mean of its neighbors. That is:
   $$f(I_{i,j} \mid \text{all other pixels}) = f(I_{i,j} \mid I_{i,j-1}, I_{i,j+1}, I_{i-1,j}, I_{i+1,j}) = N(\mu, \sigma_1^2).$$
   This model specifies the prior probability density $f(I)$ on the image space.
   (b) **Observation Model:** Let D be a noisy observation of I given by the model:
   $$D = I + W,$$
   where each element of W is an independent normal random variable with mean zero and variance σ_2^2. This equation specifies the likelihood function $f(D|I)$ for a given observation D.
   (c) **Posterior Density:** The posterior density on the image space can be written as:
   $$f(I|D) \propto f(D|I)f(I).$$
   Given the structure of $f(I)$ and the independence of the elements of W, the full conditional of this posterior density can be written as:
   $$f(I_{i,j} \mid D, \text{all other pixels}) = f(I_{i,j} \mid D_{i,j}, I_{i,j-1}, I_{i,j+1}, I_{i-1,j}, I_{i+1,j})$$
   $$= N(\mu, \sigma_1^2) N(D_{i,j}, \sigma_2^2)$$

3. **Sampling from the Posterior:** We will use an MCMC technique for sampling from the posterior. To sample from the prior density we have already used a Gibbs sampler that sequentially samples from each full conditional (of the prior). Using the same idea for $f(I|D)$ we can sequentially sample from the full conditional (of the posterior). One suggestion is that you use a Metropolis Hasting for sampling at each pixel. Write a matlab program to sample from the posterior density on the image space.

4. **Experiment:** Download five data images (midterm_1_data.mat) from the class website and use your program to generate posterior samples. For each image document the evolution of the Gibbs sampler by showing intermediate results. Use $\sigma_2 = 30$, and try different values of $\sigma_1 = 10$, 20, and 100.

Topic II: Neural Decoding by Kalman filter and Sequential Monte Carlo Methods

1. **Goal:** Using observed neural activity from brain cortex in research animals, we will perform a Bayesian analysis on this data to understand the brain mechanism and make statistical inferences about the external behaviors.

2. **Kalman Filter Model:** Let x_t in $\mathbf{R}^4$ denote [x-position, y-position, x-velocity, y-velocity] of a 2-d hand movement at time t, and y_t in $\mathbf{R}^c$ denote the spiking rate of c neurons in the primary motor cortex at the same time. A classical Kalman filter is used to model the hand kinematic state and neural activity as follows:

$$x_{t+1} = Ax_t + w_t$$
$$y_t = Hx_t + q_t$$

where $w_t \sim N(0,W), q_t \sim N(0,Q)$.

3. **Model Identification:** In the training set (midterm_2_train.mat), both hand state and neural activity are known. Use the close-form formula to estimate the model parameters A, H, W, Q.

4. **Neural Decoding:** Once the parameters are identified, we can perform neural decoding on the testing data (midterm_2_test.mat). That is, we will use neural activity to infer the movement behaviors of the hand. Two inference methods need to be used here:

    a. **Kalman Filter Algorithm:** Use a Kalman filter to estimate the hand movement. In particular, plot the true and estimated hand positions (the first two components in the state). Compute the estimation accuracy of the positions using R^2 Error.

    b. **Sequential Monte Carlo Method:** Based on the same Kalman filter model, estimate the hand positions using a sequential Monte-Carlo method. Let the number of samples n be 20, 50, 100, and 500. For each n, plot the true and estimated states. Compute the estimation accuracy using R^2 Error. Plot the accuracy as a function of *n* and compare it with the accuracy in the Kalman filter algorithm.

5. **Experiment:** Download the datasets (midterm_2_train.mat and midterm_2_test.mat) from the class website. Each set has two variables: *kin* and *rate*. *kin* is the kinematic state of the hand which includes *x-position*, *y-position*, *x-velocity*, and *y-velocity*. *rate* is the spiking rates of 42 neurons, where the rate at each time is the number of spikes within 70*ms*. Use the *kin* and *rate* in the training data to identify the model. In the test data, use the identified model and *rate* to infer *kin*, and compare the estimate with the true *kin*. In this project, the comparison only needs to focus on the hand positions, which are the first two components in *kin*.