18-699 / 42-590
Neural Signal Processing
Prof. Byron Yu Fall 2010

Problem Set 9

This problem set is due **Friday, December 3** at **4pm**. Please turn in your work, including all Matlab code and plots, to the ECE Course Hub (D200 Hamerschlag).

If you have questions, please post them on the Discussion Board on the Blackboard course website, rather than emailing the course staff. This will allow other students with the same question to see the response and any ensuing discussion.

1. Kalman filter

   In Problem Set 3, we classified neural activity recorded during movement *planning* to one of eight possible classes (where each class corresponds to a reaching angle). This is referred to as *discrete* neural decoding and is analogous to typing on a keyboard. Here, we will decode the moment-by-moment details of arm movements using the neural activity recorded around the time of movement *execution*, as illustrated in class. This is referred to as *continuous* neural decoding.

   We will analyze real neural data recorded using a 100-electrode array in premotor cortex of a macaque monkey[1]. The dataset can be found on the Blackboard course website under "Course Documents → Data sets → ps9_data.mat".

   The following describes the data format. The .mat file has two variables: `train_trial` contains the training data and `test_trial` contains the test data. Each variable is a structure of dimensions (91 trials) $\times$ (8 reaching angles). Each structure contains spike trains recorded simultaneously from 97 neurons while the monkey reached 91 times along each of 8 different reaching angles.

   The spike train recorded from the `i`th neuron on the `n`th trial of the `k`th reaching angle is contained in `train_trial(n,k).spikes(i,:)`, where $\texttt{n} = 1, \ldots, 91$, $\texttt{k} = 1, \ldots, 8$, and $\texttt{i} = 1, \ldots, 97$. A spike train is represented as a sequence of zeros and ones, where time is discretized in 1 ms bins. A zero indicates that the neuron did not spike in the 1 ms bin, whereas a one indicates that the neuron spiked once in the 1 ms bin. Each spike train is taken from movement onset to movement end. The length of the spike trains differ from trial to trial because each arm reach takes a different amount of time, even for repeated reaches to the same target.

   The arm trajectory recorded on the `n`th trial of the `k`th reaching angle is contained in `train_trial(n,k).handPos`. The hand position (in mm) in the horizontal (`handPos(1,:)`)

[1]The neural data have been generously provided by the laboratory of Prof. Krishna Shenoy at Stanford University. The data are to be used exclusively for educational purposes in this course.

and vertical (`handPos(2,:)`) directions are taken from movement onset to movement end, at 1 ms time steps. On each trial, the data in `spikes` and `handPos` are aligned in time. The structure `test_trial` has the same format as `train_trial`.

(a) Data preprocessing

The first step is to take binned spike counts and prepare the arm state vectors. We will simply look at your Matlab code when grading this part.

- **(10 points)** Take spike counts in non-overlapping 20 ms bins. For each trial, the result should be a $97 \times T$ matrix, where T is the number of 20 ms time bins on that trial. (Note: Because the length of spike trains is generally not an integer multiple of 20 ms, you can discard any partial time bins at the end of each trial. Also, T may be different on each trial.)
- **(10 points)** We will use a 4-dimensional arm state $\mathbf{z}_t$ in this problem, where

$$\mathbf{z}_t = \begin{bmatrix} \text{horz position} \\ \text{vert position} \\ \text{horz velocity} \\ \text{vert velocity} \end{bmatrix}.$$

  Sample the arm position at 20 ms intervals at the end of each spike count bin. Then, take first differences of the sampled arm position to obtain velocity (expressed in m/s). For each trial, the result should be a $4 \times T$ matrix, where T is the number of 20 ms timesteps (or time bins) on that trial.

(b) **(20 points)** Training phase

Fit the model parameters $\theta = \{A, Q, \boldsymbol{\pi}, V, C, R\}$ to the training data. Show the values in the 4×4 matrices A and Q. (Hint: The relationship between position and velocity defined by A should be consistent with the laws of physics.)

(c) Test phase

Using the model parameters found in (b), apply the Kalman filter to decode an arm trajectory for each test trial. We will create two plots, one for each of the following trials: `test_trial(1,1)` and `test_trial(1,4)`. In each plot, show:

- **(10 points)** the mean position estimate $\boldsymbol{\mu}_t^t$ ($t = 1, \ldots, T$) as red points connected by a red line (note: $\boldsymbol{\mu}_t^t$ is a 4-dimensional vector that includes both position and velocity estimates, but you can simply ignore the velocity estimate here),
- **(10 points)** the one-standard-deviation confidence ellipse corresponding to the uncertainty Σ_t^t of each position estimate in red, and
- **(5 points)** the actual arm trajectory as a black line.

(d) We will plot the same data as in part (c), but now versus time. For *each* of the two trials, create a two-panel plot, where one panel shows horizontal position versus time, and the other panel shows vertical position versus time. In each panel, show:

- **(10 points)** the decoded mean trajectory as a red solid line,
- **(10 points)** the one-standard-deviation confidence intervals using red dotted lines (note: there should be one dotted line above and another below the decoded mean trajectory), and
- **(5 points)** the actual arm trajectory as a black line.

(e) **(10 points)** Performance evaluation

At each timepoint, find the distance (in mm) between the decoded position and actual position. Average these distance errors across the timepoints of all test trials, then report this average error.