GEORGIA INSTITUTE OF TECHNOLOGY

SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

ECE 6250 Fall 2018

Problem Set #11

Assigned: 26-Nov-18

Due Date: 3-Dec-18

As stated in the syllabus, unauthorized use of previous semester course materials is strictly prohibited in this course.

All assignments will be due at the beginning of class on the due date.

PROBLEM 11.1:

Using your class notes, prepare a 1-2 paragraph summary of what we talked about in class in the last week. We do not want just a bulleted list of topics, we want you to use complete sentences and establish context (Why is what we have learned relevant? How does it connect with other things you have learned here or in other classes?). The more insight you give, the better.

PROBLEM 11.2:

Please go to http://gatech.smartevals.com and fill out survey for ECE 6250. We would very much appreciate your feedback; we do read and consider carefully the comments for every course that we teach.

PROBLEM 11.3:

(This question was on the final exam for this class a few years ago.)

Suppose we are trying to solve $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{b}$, with

$$\boldsymbol{H} = \begin{bmatrix} 3 & -1 \\ -1 & 2 \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

We are going to look at how this problem is solved by steepest descent (SD) and conjugate gradients (CG). We start both algorithms at

$$\boldsymbol{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

At the first iteration, both SD and CG move in direction

$$\boldsymbol{d}_0 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

a distance of $\alpha_0 = 2/3$ to arrive at

$$\boldsymbol{x}_1 = \begin{bmatrix} 1/3 \\ 1/3 \end{bmatrix}$$

What direction does steepest descent move in next? That is, find $\boldsymbol{d}_1$ such that $\boldsymbol{x}_2 = \boldsymbol{x}_1 + \alpha_1 \boldsymbol{d}_1$ for some $\alpha_1 \in \mathbb{R}$.

What direction does conjugate gradients move in next? (For both of these, do not worry about α_1; your answer will be only unique up to a constant.)

PROBLEM 11.4:
Recall the simple "pulse tracking" example we looked at in the Kalman filter notes.

1. Write down the Kalman update equations for this special case. Make them as simple as possible ... (Do you really need to compute $\hat{\boldsymbol{x}}_{k+1|k}$ and $\boldsymbol{P}_{k+1|k}$ as intermediate steps?)

2. As k gets large, the solution for $\hat{\boldsymbol{x}}_{k|k}$ becomes a fixed weighted sum of the previous measurements, i.e. it (approximately) obeys the convolution equation

$$\hat{\boldsymbol{x}}_{k|k} \approx \sum_{\ell=0}^{L} w[\ell] y[k-\ell].$$

   This equation is extremely accurate for very moderate values of k and L. What is an appropriate value of L and what are the weights $w[\ell]$? (Hint: For a $k = 2, 3, 4, 5, 6, \ldots$, create $\underline{\boldsymbol{A}}_k^{\mathrm{T}} \underline{\boldsymbol{A}}_k$, invert it, and extract the appropriate row. One "easy" way to create $\underline{\boldsymbol{A}}_k^{\mathrm{T}} \underline{\boldsymbol{A}}_k$ is to use the `toeplitz` command and then modify the far upper left and far lower right entries.)

PROBLEM 11.5:

We are using a radar to track a truck moving in a 2D plane with coordinates (p_x, p_y). At a series of times t_k indexed by k, we are interesting in estimating its position $(p_x(t_k), p_y(t_k)$ in the plane, and its velocity $(v_x(t_k), v_y(t_k))$ along each coordinate. We stack these into a single vector

$$\boldsymbol{x}_k = \begin{bmatrix} p_x(t_k) \\ p_y(t_k) \\ v_x(t_k) \\ v_y(t_k) \end{bmatrix}.$$

Although the velocity of the truck will drift, we expect it to remain close to a constant — that is, our best guess for $(v_x(t_{k+1}), v_y(t_{k+1}))$ is simply $(v_x(t_k), v_y(t_k))$. Our best guess for the position at time t_{k+1} is determined by the previous position $(p_x(t_k), p_y(t_k))$, previous velocity $(v_x(t_k), v_y(t_k))$, and the time $t_{k+1} - t_k$ that has elapsed between the samples. The evolution of the parameters can be modeled by

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k \boldsymbol{x}_k + \boldsymbol{\epsilon}_k.$$

At time $t = 0$, we make a direct observation that

$$(p_x(0), p_y(0)) = (0, 0), \quad \text{and} \quad (v_x(0), v_y(0)) = (1, \pi/2).$$

You might write this as $\boldsymbol{y}_0 = \boldsymbol{A}_0 \boldsymbol{x}_0$, where $\boldsymbol{A}_0 = \mathbf{I}$. At subsequent times $t_k > 0$, we make a single measurement of the position and velocity of the truck. This measurement is of the form

$$y[k] = \cos(1150\pi t_k)\, p_x(t_k) + \sin(1150\pi t_k)\, p_y(t_k) + \cos(1250\pi t_k)\, v_x(t_k) + \sin(1250\pi t_k)\, v_y(t_k).$$

1. Write down the "state evolution equations". That is, write down the 4×4 matrix $\boldsymbol{F}_k$ explicitly (it should depend on $t_{k+1} - t_k$).

2. The file `kalman_data.mat` contains 499 measurements (in the vector `y`) for different times (in the vector `t`). Implement a Kalman filter to track the truck, and plot each of your 500 estimates of the position in the (p_x, p_y) plane from $\hat{\boldsymbol{x}}_{k|k}$ on a set of axes. To plot discrete points rather than a connected line, use something like

   ```
   plot(pxhat,pyhat,'o')
   ```

3. Construct and solve the (large) system to estimate all 500 positions with knowledge of all of the measurements. That is, find $\hat{\boldsymbol{x}}_{0|499}, \hat{\boldsymbol{x}}_{1|499}, \ldots, \hat{\boldsymbol{x}}_{499|499}$. Plot the corresponding position estimates on another set of axes (maybe using `'x'` for the plot command), and overlay your answer from the previous part (maybe using `'o'` for the plot command again). Comment on what you see, in particular on how it related to your answer in the previous part.