EE263 Homework 5
Fall 2023

6.790. Identifying a system from input/output data. We consider the standard setup:

$$y = Ax + v,$$

where $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$ is the input vector, $y \in \mathbb{R}^m$ is the output vector, and $v \in \mathbb{R}^m$ is the noise or disturbance. We consider here the problem of estimating the matrix A, given some input/output data. Specifically, we are given the following:

$$x^{(1)}, \ldots, x^{(N)} \in \mathbb{R}^n, \qquad y^{(1)}, \ldots, y^{(N)} \in \mathbb{R}^m.$$

These represent N samples or observations of the input and output, respectively, possibly corrupted by noise. In other words, we have

$$y^{(k)} = Ax^{(k)} + v^{(k)}, \quad k = 1, \ldots, N,$$

where $v^{(k)}$ are assumed to be small. The problem is to estimate the (coefficients of the) matrix A, based on the given input/output data. You will use a least-squares criterion to form an estimate $\hat{A}$ of A. Specifically, you will choose as your estimate $\hat{A}$ the matrix that minimizes the quantity

$$J = \sum_{k=1}^{N} \|Ax^{(k)} - y^{(k)}\|^2$$

over A.

a) Explain how to do this. If you need to make an assumption about the input/output data to make your method work, state it clearly. You may want to use the matrices $X \in \mathbb{R}^{n\times N}$ and $Y \in \mathbb{R}^{m\times N}$ given by

$$X = \begin{bmatrix} x^{(1)} & \cdots & x^{(N)} \end{bmatrix}, \qquad Y = \begin{bmatrix} y^{(1)} & \cdots & y^{(N)} \end{bmatrix}$$

in your solution.

b) On the course web site you will find some input/output data for an instance of this problem in the file `sysid_data.json`. Executing this Julia file will assign values to m, n, and N, and create two matrices that contain the input and output data, respectively. The $n \times N$ matrix variable `X` contains the input data $x^{(1)}, \ldots, x^{(N)}$ (*i.e.*, the first column of `X` contains $x^{(1)}$, etc.). Similarly, the $m\times N$ matrix `Y` contains the output data $y^{(1)}, \ldots, y^{(N)}$. You must give your final estimate $\hat{A}$, your source code, and also give an explanation of what you did.

6.810. Estimating a signal with interference. This problem concerns three proposed methods for estimating a signal, based on a measurement that is corrupted by a small noise and also by an interference, that need not be small. We have

$$y = Ax + Bv + w,$$

where $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{m\times p}$ are known. Here $y \in \mathbb{R}^m$ is the measurement (which is known), $x \in \mathbb{R}^n$ is the signal that we want to estimate, $v \in \mathbb{R}^p$ is the interference, and w is a noise. The noise is unknown, and can be assumed to be small. The interference is unknown, but cannot be assumed to be small. You can assume that the matrices A and B are skinny and full rank (*i.e.*, $m > n$, $m > p$), and that the ranges of A and B intersect only at 0. (If this last condition does not hold, then there is no hope of finding x, even when $w = 0$, since a nonzero interference can masquerade as a signal.) Each of the EE263 TAs proposes a method for estimating x. These methods, along with some informal justification from their proposers, are given below. Nikola proposes the **ignore and estimate method.** He describes it as follows:

> We don't know the interference, so we might as well treat it as noise, and just ignore it during the estimation process. We can use the usual least-squares method, for the model $y = Ax + z$ (with z a noise) to estimate x. (Here we have $z = Bv + w$, but that doesn't matter.)

Almir proposes the **estimate and ignore method.** He describes it as follows:

> We should simultaneously estimate both the signal x *and* the interference v, based on y, using a standard least-squares method to estimate $[x^\mathsf{T}\ v^\mathsf{T}]^\mathsf{T}$ given y. Once we've estimated x and v, we simply ignore our estimate of v, and use our estimate of x.

Miki proposes the **estimate and cancel method.** He describes it as follows:

> Almir's method makes sense to me, but I can improve it. We should simultaneously estimate both the signal x and the interference v, based on y, using a standard least-squares method, exactly as in Almir's method. In Almir's method, we then throw away $\hat{v}$, our estimate of the interference, but I think we should use it. We can form the "pseudo-measurement" $\tilde{y} = y - B\hat{v}$, which is our measurement, with the effect of the estimated interference subtracted off. Then, we use standard least-squares to estimate x from $\tilde{y}$, from the simple model $\tilde{y} = Ax + z$. (This is exactly as in Nikola's method, but here we have subtracted off or cancelled the effect of the estimated interference.)

These descriptions are a little vague; part of the problem is to translate their descriptions into more precise algorithms.

a) Give an explicit formula for each of the three estimates. (That is, for each method give a formula for the estimate $\hat{x}$ in terms of A, B, y, and the dimensions n, m, p.)

b) Are the methods really different? Identify any pairs of the methods that coincide (*i.e.*, always give exactly the same results). If they are all three the same, or all three different, say so. Justify your answer. To show two methods are the same, show that the formulas given in part (a) are equal (even if they don't appear to be at first). To show two methods are different, give a specific numerical example in which the estimates differ.

c) Which method or methods do you think work best? Give a very brief explanation. (If your answer to part (b) is "The methods are all the same" then you can simply repeat here, "The methods are all the same".)

7.1040. Fitting a Gaussian function to data. A Gaussian function has the form

$$f(t) = ae^{-(t-\mu)^2/\sigma^2}.$$

Here $t \in \mathbb{R}$ is the independent variable, and $a \in \mathbb{R}$, $\mu \in \mathbb{R}$, and $\sigma \in \mathbb{R}$ are parameters that affect its shape. The parameter a is called the *amplitude* of the Gaussian, μ is called its *center*, and σ is called the *spread* or *width*. We can always take $\sigma > 0$. For convenience we define $p \in \mathbb{R}^3$ as the vector of the parameters, *i.e.*, $p = [a\ \mu\ \sigma]^\mathsf{T}$. We are given a set of data,

$$t_1, \ldots, t_N, \qquad y_1, \ldots, y_N,$$

and our goal is to fit a Gaussian function to the data. We will measure the quality of the fit by the root-mean-square (RMS) fitting error, given by

$$E = \left(\frac{1}{N} \sum_{i=1}^{N} (f(t_i) - y_i)^2 \right)^{1/2}.$$

Note that E is a function of the parameters a, μ, σ, *i.e.*, p. Your job is to choose these parameters to minimize E. You'll use the Gauss-Newton method.

a) Work out the details of the Gauss-Newton method for this fitting problem. Explicitly describe the Gauss-Newton steps, including the matrices and vectors that come up. You can use the notation $\Delta p^{(k)} = [\Delta a^{(k)}\ \Delta\mu^{(k)}\ \Delta\sigma^{(k)}]^\mathsf{T}$ to denote the update to the parameters, *i.e.*,

$$p^{(k+1)} = p^{(k)} + \Delta p^{(k)}.$$

(Here k denotes the kth iteration.)

b) Get the data t, y (and N) from the file `gauss_fit_data.json`, available on the class website. Implement the Gauss-Newton (as outlined in part (a) above). You'll need an initial guess for the parameters. You can visually estimate them (giving a short justification), or estimate them by any other method (but you must explain your method). Plot the RMS error E as a function of the iteration number. (You should plot enough iterations to convince yourself that the algorithm has nearly converged.) Plot the final Gaussian function obtained along with the data on the same plot. Repeat for another reasonable, but different initial guess for the parameters. Repeat for another set of parameters that is *not* reasonable, *i.e.*, not a good guess for the parameters. (It's possible, of course, that the Gauss-Newton algorithm doesn't converge, or fails at some step; if this occurs, say so.) Briefly comment on the results you obtain in the three cases.

8.1110. Simultaneous left inverse of two matrices. Consider a system where

$$y = Gx, \quad \tilde{y} = \tilde{G}x$$

where $G \in \mathbb{R}^{m\times n}$, $\tilde{G} \in \mathbb{R}^{m\times n}$. Here x is some variable we wish to estimate or find, y gives the measurements with some set of (linear) sensors, and $\tilde{y}$ gives the measurements with some *alternate* set of (linear) sensors. We want to find a *reconstruction matrix* $H \in \mathbb{R}^{n\times m}$ such that $HG = H\tilde{G} = I$. Such a reconstruction matrix has the nice property that it recovers x

perfectly from *either* set of measurements (y or $\tilde{y}$), *i.e.*, $x = Hy = H\tilde{y}$. Consider the specific case

$$G = \begin{bmatrix} 2 & 3 \\ 1 & 0 \\ 0 & 4 \\ 1 & 1 \\ -1 & 2 \end{bmatrix}, \quad \tilde{G} = \begin{bmatrix} -3 & -1 \\ -1 & 0 \\ 2 & -3 \\ -1 & -3 \\ 1 & 2 \end{bmatrix}.$$

Either find an explicit reconstruction matrix H, or explain why there is no such H.