PROBLEM SET 4

1. *Modifying measurements to satisfy known conservation laws.* A vector $y \in \mathbf{R}^n$ contains n measurements of some physical quantities $x \in \mathbf{R}^n$. The measurements are good, but not perfect, so we have $y \approx x$.

   From physical principles it is known that the quantities x must satisfy some linear equations, *i.e.*,
   $$a_i^T x = b_i, \qquad i = 1, \ldots, m,$$
   where $m < n$. As a simple example, if x_1 is the current in a circuit flowing into a node, and x_2 and x_3 are the currents flowing out of the node, then we must have $x_1 = x_2 + x_3$. More generally, the linear equations might come from various conservation laws, or balance equations (mass, heat, energy, charge ...). The vectors a_i and the constants b_i are known, and we assume that $a_1, \ldots, a_m$ are independent.

   Due to measurement errors, the measurement y won't satisfy the conservation laws (*i.e.*, linear equations above) exactly, although we would expect $a_i^T y \approx b_i$.

   An engineer proposes to adjust the measurements y by adding a correction term $c \in \mathbf{R}^n$, to get an adjusted estimate of x, given by
   $$y_{\text{adj}} = y + c.$$

   She proposes to find the smallest possible correction term (measured by $\|c\|$) such that the adjusted measurements y_{adj} satisfy the known conservation laws.

   Give an explicit formula for the correction term, in terms of y, a_i, b_i. If any matrix inverses appear in your formula, explain why the matrix to be inverted is nonsingular. Verify that the resulting adjusted measurement satisfies the conservation laws, *i.e.*, $a_i^T y_{\text{adj}} = b_i$.

2. *Estimation with sensor offset and drift.* We consider the usual estimation setup:
   $$y_i = a_i^T x + v_i, \qquad i = 1, \ldots, m,$$
   where

   - y_i is the ith (scalar) measurement

- $x \in \mathbf{R}^n$ is the vector of parameters we wish to estimate from the measurements

- v_i is the sensor or measurement error of the ith measurement

In this problem we assume the measurements y_i are taken at times evenly spaced, T seconds apart, starting at time $t = T$. Thus, y_i, the ith measurement, is taken at time $t = iT$. (This isn't really material; it just makes the interpretation simpler.)

You can assume that $m \geq n$ and the measurement matrix

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_m^T \end{bmatrix}$$

is full rank (*i.e.*, has rank n).

Usually we assume (often implicitly) that the measurement errors v_i are random, unpredictable, small, and centered around zero. (You don't need to worry about how to make this idea precise.) In such cases, least-squares estimation of x works well.

In some cases, however, the measurement error includes some *predictable* terms. For example, each sensor measurement might include a (common) *offset* or *bias*, as well as a term that grows linearly with time (called a *drift*). We model this situation as

$$v_i = \alpha + \beta iT + w_i$$

where α is the sensor bias (which is unknown but the *same* for all sensor measurements), β is the drift term (again the same for all measurements), and w_i is part of the sensor error that is unpredictable, small, and centered around 0.

If we knew the offset α and the drift term β we could just subtract the predictable part of the sensor signal, *i.e.*, $\alpha + \beta iT$ from the sensor signal. But we're interested in the case where we don't know the offset α or the drift coefficient β.

Show how to use least-squares to *simultaneously* estimate the parameter vector $x \in \mathbf{R}^n$, the offset $\alpha \in \mathbf{R}$, and the drift coefficient $\beta \in \mathbf{R}$. Clearly explain your method. If your method always works, say so. Otherwise describe the conditions (on the matrix A) that must hold for your method to work, and give a simple example where the conditions don't hold.

Remark: This is widely used in practical applications. In GPS, for example, the offset in range measurements is due to a fixed skew between the clock in the transmitting satellite and the receiver; the drift is due to the drift between the clocks.

3. *Minimum distance and maximum correlation decoding.* We consider a simple communication system, in which a sender transmits one of N possible signals to a receiver, which receives a version of the signal sent that is corrupted by noise. Based on the corrupted received signal, the receiver has to estimate or guess which of the N signals was sent.

We will represent the signals by vectors in $\mathbf{R}^n$. We will denote the possible signals as $a_1, \ldots, a_N \in \mathbf{R}^n$. These signals, which collectively are called the *signal constellation*, are known to both the transmitter and receiver.

When the signal a_k is sent, the received signal is $a_{\text{recd}} = a_k + v$, where $v \in \mathbf{R}^n$ is (channel or transmission) noise. In a communications course, the noise v is described by a statistical model, but here we'll just assume that it is 'small' (and in any case, it does not matter for the problem).

The receiver must make a guess or estimate as to which of the signals was sent, based on the received signal a_{recd}. There are many ways to do this, but in this problem we explore two methods.

- *Minimum distance decoding.* Choose as the estimate of the decoded signal the one in the constellation that is closest to what is received, *i.e.*, choose a_k that minimizes $\|a_{\text{recd}} - a_i\|$. For example, if we have $N = 3$ and

$$\|a_{\text{recd}} - a_1\| = 2.2, \qquad \|a_{\text{recd}} - a_2\| = 0.3, \qquad \|a_{\text{recd}} - a_3\| = 1.1,$$

  then the minimum distance decoder would guess that the signal a_2 was sent.

- *Maximum correlation decoding.* Choose as the estimate of the decoded signal the one in the constellation that has the largest inner product with the received signal, *i.e.*, choose a_k that maximizes $a_{\text{recd}}^T a_i$. For example, if we have $N = 3$ and

$$a_{\text{recd}}^T a_1 = -1.1, \qquad a_{\text{recd}}^T a_2 = 0.2, \qquad a_{\text{recd}}^T a_3 = 1.0,$$

  then the maximum correlation decoder would guess that the signal a_3 was sent.

For both methods, let's not worry about breaking ties. You can just assume that ties never occur; one of the signals is always closest to, or has maximum inner product with, the received signal.

Give some general conditions on the constellation (*i.e.*, the set of vectors $a_1, \ldots, a_N$) under which these two decoding methods are the same. By 'same' we mean this: for any received signal a_{recd}, the decoded signal for the two methods is the same.

Give the simplest condition you can; we'll take off credit for answers that are technically correct but longwinded. You can refer to any of the concepts from the course, *e.g.*, range, nullspace, independence, norms, QR factorization, etc.

You must show how the decoding schemes always give the same answer, when your conditions hold. Also, give a specific counterexample, for which your conditions don't hold, and the methods differ. (We are *not* asking you to show that when your conditions don't hold, the two decoding schemes differ for some received signal.)

You might want to check simple cases like $n = 1$ (scalar signals), $N = 2$ (only two messages in the constellation), or draw some pictures. But then again, you might not.

4. *Estimating emissions from spot measurements.* There are n sources of a pollutant, at known locations $s_1, \ldots, s_n \in \mathbf{R}^2$. Each source emits the pollutant at some emission

rate; we let x_j denote the emission rate for source j. (These are positive, but to simplify things we won't concern ourselves with that.) The emission rates are to be determined, or estimated.

We measure the total pollutant level at m spots, located at $t_1, \ldots, t_m \in \mathbf{R}^2$, which are known. The total pollutant measured at spot i is the sum of the contributions from the n sources. The contribution from source j to measurement i is given by $\alpha x_j / \|s_j - t_i\|^2$, where α is a known (positive) constant. In other words, the pollutant concentration from a source follows an inverse square law, and is proportional to the emission rate. We assume that measurement spots do not coincide with the source locations, *i.e.*, we do not have $s_j = t_i$ for any i or j. We also assume that none of the spot locations is repeated (*i.e.*, we have $t_i \neq t_j$ for $i \neq j$) and that none of the source locations is repeated (*i.e.*, we have $s_i \neq s_j$ for $i \neq j$).

(a) Give a *specific* example of source and spot measurement locations, with 4 sensors and 3 sources, for which it is *impossible* to find the emission rates given the spot measurements. In this part, we ignore the issue of noise or sensor errors; we assume the spot measurements are exactly as described above. To show that your configuration is a valid example, give two *specific* different sets of emission rates that yield identical spot measurements. You are free to (briefly) explain your example using concepts such as range, nullspace, rank, and so on; but remember, we want a *specific numerical example*, such as as $s_1 = [0 \ 1]^T, \ldots, s_3 = [1 \ 2]^T$, $t_1 = [1 \ 1]^T, \ldots, t_4 = [3 \ 2]^T$. (And similarly for the two emission rates that give the same spot measurements.)

(b) Get the data from the file `emissions_data.m` that is available on the class web site. This file defines three source locations (given as a 2×3 matrix; the columns give the locations), and ten spot measurement locations (given as a 2×10 matrix). It also gives two sets of spot measurements: one for part (b), and one for part (c). Be careful to use the right set of measurements for each problem!

The spot measurements are not perfect (as we assumed in part (a)); they contain small noise and errors. Estimate the pollutant emission rates. Explain your method, and give your estimate for the emissions rates of the three sources.

(c) Now we suppose that *one* of the spot measurments is faulty, *i.e.*, its associated noise or error is far larger than the errors of the other spot measurements. Explain how you would identify or guess which one is malfunctioning, and then estimate the source emission rates. Carry out your method on the data given in the matlab file. Be sure to tell us which spot measurement you believe to be faulty, and what your guess of the emission rates is. (The emission rates are *not* the same as in part (b), but the source and spot measurement locations are.)

5. *Optimal flow on a data collection network.* We consider a communications network with m nodes, plus a special destination node, and n communication links. Each communication link connects two (distinct) nodes and is bidirectional, *i.e.*, information can flow in either direction. We will assume that the network is connected, *i.e.*, there is a path, or sequence of links, from every node (including the special destination

node) to every other node. With each communication link we associate a directed arc, which defines the direction of information flow that we will call positive. Using these reference directions, the flow or traffic on link j is denoted f_j. (The units are bits per second, but that won't matter to us.) The traffic on the network (*i.e.*, the flow in each communication link) is given by a vector $f \in \mathbf{R}^n$.

A small example is shown in part 2 of this problem. In this example, nodes 1 and 3 are connected by communication link 4, and the associated arc points from node 1 to node 3. Thus $f_4 = 12$ means the flow on that link is 12 (bits per second), from node 1 to node 3. Similarly, $f_4 = -3$ means the flow on link 4 is 3 (bits per second), from node 3 to node 1.

External information enters each of the m regular nodes and flows across links to the special destination node. In other words, the network is used to collect information from the nodes and route it through the links to the special destination node. (That explains why we call it a data collection network.) At node i, an external information flow s_i (which is nonnegative) enters. The vector $s \in \mathbf{R}^m$ of external flows is sometimes called the *source vector*.

Information flow is conserved. This means that at each node (except the special destination node) the sum of all flows entering the node from communication links connected to that node, plus the external flow, equals the sum of the flows leaving that node on communication links. As an example, consider node 3 in the network of part 2. Links 4 and 5 enter this node, and link 6 leaves the node. Therefore, flow conservation at node 3 is given by

$$f_4 + f_5 + s_3 = f_6.$$

The first two terms on the left give the flow entering node 3 on links 4 and 5; the last term on the left gives the external flow entering node 3. The term on the righthand side gives the flow leaving over link 6. Note that this equation correctly expresses flow conservation regardless of the signs of f_4, f_5, and f_6.

Finally, here is the problem.

(a) The vector of external flows, $s \in \mathbf{R}^m$, and the network topology, are given, and you must find the flow f that satisfies the conservation equations, and minimizes the mean-square traffic on the network, *i.e.*,

$$\frac{1}{n} \sum_{j=1}^{n} f_j^2.$$

Your answer should be in terms of the external flow s, and the *node incidence matrix* $A \in \mathbf{R}^{m \times n}$ that describes the network topology. The node incidence matrix is defined as

$$A_{ij} = \begin{cases} 1 & \text{arc } j \text{ enters (or points into) node } i \\ -1 & \text{arc } j \text{ leaves (or points out of) node } i \\ 0 & \text{otherwise.} \end{cases}$$

Note that each row of A is associated with a node on the network (not including the destination node), and each column is associated with an arc or link.

(b) Now consider the specific (and very small) network shown below. The nodes are shown as circles, and the special destination node is shown as a square. The external flows are

$$s = \begin{bmatrix} 1 \\ 4 \\ 10 \\ 10 \end{bmatrix}.$$

One simple feasible flow is obtained by routing all the external flow entering each node along a shortest path to the destination. For example, all the external flow entering node 2 goes to node 1, then to the destination node. For node 3, which has two shortest paths to the destination, we arbitrarily choose the path through node 4. This simple routing scheme results in the feasible flow

$$f_{\text{simple}} = \begin{bmatrix} 5 \\ 4 \\ 0 \\ 0 \\ 0 \\ 10 \\ 20 \end{bmatrix}.$$

Find the mean square optimal flow for this problem (as in part 1). Compare the mean square flow of the optimal flow with the mean square flow of f_{simple}.