Homework 7 additional problems

1. *Identifying a sparse linear dynamical system.* A linear dynamical system has the form

$$x(t+1) = Ax(t) + Bu(t) + w(t), \quad t = 1, \ldots, T-1,$$

where $x(t) \in \mathbf{R}^n$ is the state, $u(t) \in \mathbf{R}^m$ is the input signal, and $w(t) \in \mathbf{R}^n$ is the process noise, at time t. We assume the process noises are IID $\mathcal{N}(0, W)$, where $W \succ 0$ is the covariance matrix. The matrix $A \in \mathbf{R}^{n \times n}$ is called the dynamics matrix or the state transition matrix, and the matrix $B \in \mathbf{R}^{n \times m}$ is called the input matrix.

You are given accurate measurements of the state and input signal, *i.e.*, $x(1), \ldots, x(T)$, $u(1), \ldots, u(T-1)$, and W is known. Your job is to find a state transition matrix $\hat{A}$ and input matrix $\hat{B}$ from these data, that are plausible, and in addition are sparse, *i.e.*, have many zero entries. (The sparser the better.)

By doing this, you are effectively estimating the structure of the dynamical system, *i.e.*, you are determining which components of $x(t)$ and $u(t)$ affect which components of $x(t+1)$. In some applications, this structure might be more interesting than the actual values of the (nonzero) coefficients in $\hat{A}$ and $\hat{B}$.

By plausible, we mean that

$$\sum_{t=1}^{T-1} \left\| W^{-1/2} \left(x(t+1) - \hat{A}x(t) - \hat{B}u(t) \right) \right\|_2^2 \in n(T-1) \pm 2\sqrt{2n(T-1)},$$

where $a \pm b$ means the interval $[a-b, a+b]$. (You can just take this as our definition of plausible. But to explain this choice, we note that when $\hat{A} = A$ and $\hat{B} = B$, the left-hand side is χ^2, with $n(T-1)$ degrees of freedom, and so has mean $n(T-1)$ and standard deviation $\sqrt{2n(T-1)}$.)

(a) Describe a method for finding $\hat{A}$ and $\hat{B}$, based on convex optimization.

We are looking for a *very simple* method, that involves solving *one* convex optimization problem. (There are many extensions of this basic method, that would improve the simple method, *i.e.*, yield sparser $\hat{A}$ and $\hat{B}$ that are still plausible. We're not asking you to describe or implement any of these.)

(b) Carry out your method on the data found in `sparse_lds_data.m`. Give the values of $\hat{A}$ and $\hat{B}$ that you find, and verify that they are plausible.

In the data file, we give you the true values of A and B, so you can evaluate the performance of your method. (Needless to say, you are not allowed to use these values when forming $\hat{A}$ and $\hat{B}$.) Using these true values, give the number of false positives and false negatives in both $\hat{A}$ and $\hat{B}$. A false positive in $\hat{A}$, for example, is an entry that is nonzero, while the corresponding entry in A is zero. A false negative is an entry of $\hat{A}$ that is zero, while the corresponding entry of A is nonzero. To judge whether an entry of $\hat{A}$ (or $\hat{B}$) is nonzero, you can use the test $|\hat{A}_{ij}| \geq 0.01$ (or $|\hat{B}_{ij}| \geq 0.01$).

2. *Maximum likelihood prediction of team ability.* A set of n teams compete in a tournament. We model each team's ability by a number $a_j \in [0,1]$, $j = 1, \ldots, n$. When teams j and k play each other, the probability that team j wins is equal to $\mathbf{prob}(a_j - a_k + v > 0)$, where $v \sim \mathcal{N}(0, \sigma^2)$.

   You are given the outcome of m past games. These are organized as

$$(j^{(i)}, k^{(i)}, y^{(i)}), \quad i = 1, \ldots, m,$$

   meaning that game i was played between teams $j^{(i)}$ and $k^{(i)}$; $y^{(i)} = 1$ means that team $j^{(i)}$ won, while $y^{(i)} = -1$ means that team $k^{(i)}$ won. (We assume there are no ties.)

   (a) Formulate the problem of finding the maximum likelihood estimate of team abilities, $\hat{a} \in \mathbf{R}^n$, given the outcomes, as a convex optimization problem. You will find the *game incidence matrix* $A \in \mathbf{R}^{m \times n}$, defined as

$$A_{il} = \begin{cases} y^{(i)} & l = j^{(i)} \\ -y^{(i)} & l = k^{(i)} \\ 0 & \text{otherwise,} \end{cases}$$

   useful.

   The prior constraints $\hat{a}_i \in [0,1]$ should be included in the problem formulation. Also, we note that if a constant is added to all team abilities, there is no change in the probabilities of game outcomes. This means that $\hat{a}$ is determined only up to a constant, like a potential. But this doesn't affect the ML estimation problem, or any subsequent predictions made using the estimated parameters.

   (b) Find $\hat{a}$ for the team data given in `team_data.m`, in the matrix `train`. (This matrix gives the outcomes for a tournament in which each team plays each other team once.) You may find the `cvx` function `log_normcdf` helpful for this problem.

   You can form A using the commands

```
A = sparse(1:m,train(:,1),train(:,3),m,n) + ...
    sparse(1:m,train(:,2),-train(:,3),m,n);
```

   (c) Use the maximum likelihood estimate $\hat{a}$ found in part (b) to predict the outcomes of next year's tournament games, given in the matrix `test`, using $\hat{y}^{(i)} = \mathbf{sign}(\hat{a}_{j^{(i)}} - \hat{a}_{k^{(i)}})$. Compare these predictions with the actual outcomes, given in the third column of `test`. Given the fraction of correctly predicted outcomes.

   The games played in `train` and `test` are the same, so another, simpler method for predicting the outcomes in `test` it to just assume the team that won last year's match will also win this year's match. Give the percentage of correctly predicted outcomes using this simple method.

3. *Three-way linear classification.* We are given data

$$x^{(1)}, \ldots, x^{(N)}, \quad y^{(1)}, \ldots, y^{(M)}, \quad z^{(1)}, \ldots, z^{(P)},$$

three nonempty sets of vectors in $\mathbf{R}^n$. We wish to find three affine functions on $\mathbf{R}^n$,

$$f_i(z) = a_i^T z - b_i, \quad i = 1, 2, 3,$$

that satisfy the following properties:

$$\begin{array}{ll} f_1(x^{(j)}) > \max\{f_2(x^{(j)}), f_3(x^{(j)})\}, & j = 1, \ldots, N, \\ f_2(y^{(j)}) > \max\{f_1(y^{(j)}), f_3(y^{(j)})\}, & j = 1, \ldots, M, \\ f_3(z^{(j)}) > \max\{f_1(z^{(j)}), f_2(z^{(j)})\}, & j = 1, \ldots, P. \end{array}$$

In words: f_1 is the largest of the three functions on the x data points, f_2 is the largest of the three functions on the y data points, f_3 is the largest of the three functions on the z data points. We can give a simple geometric interpretation: The functions f_1, f_2, and f_3 partition $\mathbf{R}^n$ into three regions,

$$\begin{array}{rcl} R_1 & = & \{z \mid f_1(z) > \max\{f_2(z), f_3(z)\}\}, \\ R_2 & = & \{z \mid f_2(z) > \max\{f_1(z), f_3(z)\}\}, \\ R_3 & = & \{z \mid f_3(z) > \max\{f_1(z), f_2(z)\}\}, \end{array}$$

defined by where each function is the largest of the three. Our goal is to find functions with $x^{(j)} \in R_1$, $y^{(j)} \in R_2$, and $z^{(j)} \in R_3$.

Pose this as a convex optimization problem. You may not use strict inequalities in your formulation.

Solve the specific instance of the 3-way separation problem given in `sep3way_data.m`, with the columns of the matrices `X`, `Y` and `Z` giving the $x^{(j)}$, $j = 1, \ldots, N$, $y^{(j)}$, $j = 1, \ldots, M$ and $z^{(j)}$, $j = 1, \ldots, P$. To save you the trouble of plotting data points and separation boundaries, we have included the plotting code in `sep3way_data.m`. (Note that `a1`, `a2`, `a3`, `b1` and `b2` contain arbitrary numbers; you should compute the correct values using CVX.)