EE263s final solutions

Summer quarter 2008-09
Jacob Mattingley
Stanford University

1. **A two-factor model of historic stock prices.** (20 points)

   We want to explain a set of historic tech stock prices using a simple model. We will use a simple *two factor model*, where we simultaneously choose coefficients and basis functions to minimize a particular measure of error.

   You are given a set of N, length-T vector stock price vectors $p^{(1)}, \dots, p^{(N)}$. Component $p_t^{(i)}$, for example, is the price of stock i at time t. We will choose length-T vector factors x and y (with components $x_1, \dots, x_T$ and $y_1, \dots, y_T$), along with scalar coefficients $a_1, b_1, \dots, a_N, b_N$, that minimize the total fitting error

   $$J = \sum_{i=1}^{N} \sum_{t=1}^{T} (p_t^{(i)} - a_i x_t - b_i y_t)^2.$$

   (a) Clearly explain how to simultaneously choose the factors, ie, x_i, y_i, $i = 1, \dots, T$, and all the coefficients a_i, b_i, $i = 1, \dots, N$, that minimize J.

   (b) The file `stocks.m` contains weekly data for ten tech stocks over the course of a year ($N = 10$ and $T = 52$). The data are given as a $T \times N$ matrix P. Fit a two-factor model, and report the minimum J.

   (c) Calculate the *relative fitting errors* for each stock: report

   $$E_i = \frac{\|p^{(i)} - a_i x - b_i y\|}{\|p^{(i)}\|}, \quad i = 1, \dots, n.$$

   Using this measure, which stock price is best explained by your model? Which is most poorly explained? (You'll see which stocks they are at the top of `stocks.m`.)

   Solution.

   (a) We start by placing the prices, factors and coefficients in matrices, so that we have

   $$P = \left[\begin{array}{ccc} p^{(1)} & \cdots & p^{(N)} \end{array} \right], \quad X = \left[\begin{array}{cc} x & y \end{array} \right], \quad A = \left[\begin{array}{ccc} a_1 & \cdots & a_N \\ b_1 & \cdots & b_N \end{array} \right]$$

   Now, the problem is to choose X and A so that we minimize $\|P - XA\|_F^2$. As we discussed in class, this is the same as choosing X and A to minimize $\|P - XA\|$, with the usual matrix norm.

But this problem is easily solved: we find an optimal rank 2 approximant of P. We'll find the SVD of P, then use the first two components, setting

$$X = \begin{bmatrix} u_1 & u_2 \end{bmatrix}, \quad A = \begin{bmatrix} \sigma_1 v_1^T \\ \sigma_2 v_2^T \end{bmatrix}.$$

(The factors σ_1 and σ_2 could also appear in X.)

(b) The Matlab code used in this question appears below.

```
% Load data.
stocks;

[U, E, V] = svd(P);
X = U(:,1:2);
A = E(1:2,1:2)*V(:,1:2)';
D = P - X*A;
J = norm(D, 'fro')^2

% Look at the relative errors.
for i = 1:N
    norm(D(:,i))/norm(P(:,i))
end
```

The minimum J is 2439. The

(c) The relative errors, as given by Matlab, are

1	aapl	0.0101
2	csco	0.0646
3	goog	0.0964
4	hpq	0.0671
5	ibm	0.0040
6	java	0.3453
7	msft	0.0357
8	orcl	0.0652
9	palm	0.0898
10	yhoo	0.1530

Hence, IBM has the best approximation, and Sun Microsystems (java) the worst. You might claim, then, that IBM's performance was (very roughly) 'typical of stocks of its type' and that Sun's performance was 'atypical'.

2. **An energy efficient journey.** (30 points)

A client offers you a job. He has a state-space model for a small, unusual vehicle. He also has some waypoints, and the times the vehicle must be at those waypoints. His instructions say that you should choose a set of inputs that gets you 'to the waypoints at the required times, while simultaneously keeping the inputs small'.

You decide to take the job, and your initial consulting fee goes into making the problem mathematically rigorous. First, our state-space model:

$$x(t+1) = Ax(t) + Bu(t), \quad y(t) = Cx(t).$$

This model incorporates a coordinate change, so our system starts with $x(0) = 0$.

There are four waypoint positions $y_1^{\mathrm{wp}}, \ldots, y_4^{\mathrm{wp}}$ that we wish to reach at times $T_1, \ldots, T_4$. We will measure the total size of our inputs as

$$J_1 = \sum_{t=0}^{T_4-1} \|u(t)\|^2.$$

(a) The first step is to write the positions at times $T_1, \ldots, T_4$ in terms of the inputs $u(0), \ldots, u(T_4-1)$. Explain how to find the matrix F, where

$$\begin{bmatrix} y(T_1) \\ \vdots \\ y(T_4) \end{bmatrix} = F \begin{bmatrix} u(0) \\ \vdots \\ u(T_4-1) \end{bmatrix}$$

(b) Explain how to find the smallest set of inputs ($u(t)$ that minimizes J_1) for which the vehicle reaches the waypoints at the desired times.

(c) Carry out your method on the data in `vehicle.m`. What is your value of J_1? Use `vehicle_plot.m` to show your vehicle's trajectory. Check that the red circles (your vehicle's positions at times $T_1, \ldots, T_4$) are on top of the blue squares (the waypoints).

(d) The client now comes back and says that they really need $J_1 \leq 1$. You explain that it isn't possible to reach these particular waypoints at the desired time, while achieving $J_1 \leq 1$. Clearly explain why you cannot do this.

(e) After much discussion (and some hefty consulting fees), the client decides that getting *near* the waypoints is sufficient. Now, he wants to minimize

$$J_2 = \sum_{i=1}^{4} \|y(T_i) - y_i^{\mathrm{wp}}\|^2,$$

while keeping $J_1 \leq 1$. Thus, you are aiming to get as close as possible to the waypoints with the given constraint. Explain how to find the $u(t)$ that minimizes J_2, while keeping $J_1 \leq 1$.

(f) Again carry out your method on the data in `vehicle.m`. Plot the new trajectory. What is your value of J_2?

Solution.

(a) After a little development, we see that

$$y(t) = CA^{t-1}Bu(0) + CA^{t-2}Bu(1) + \cdots + CABu(t-2) + CBu(t-1).$$

Thus, the tth block row of F is

$$f_t^T = \begin{bmatrix} CA^{t-1}B & CA^{t-2}B & \cdots & CAB & CB & 0 & \cdots & 0 \end{bmatrix},$$

where there are t nonzero blocks followed by $T_4 - t$ zero blocks. If there are m outputs and p inputs, each block is $m \times p$, so f_t^T is $m \times T_4 p$.

We form F by concatenating four of these rows, to get a $4m \times T_4 p$ matrix.

(b) We have an underdetermined system, so we simply find the least-norm solution to the given equation, with our choice of F. This will exactly reach the waypoints. In particular, if we use the notation

$$Y = \begin{bmatrix} y(T_1) \\ \vdots \\ y(T_4) \end{bmatrix} = F \begin{bmatrix} u(0) \\ \vdots \\ u(T_4 - 1) \end{bmatrix} = FU,$$

then

$$U^{\mathrm{ln}} = F^T (FF^T)^{-1} Y.$$

(c) To do this in Matlab, we use the following code. This code also includes the solution for part (f).

```
% Load data.
vehicle;

F = zeros(4*m, T4*p);
Ts = [T1 T2 T3 T4];
for i = 1:length(Ts)
    t = Ts(i);
    for j = 1:t
        % Assign the (i,j) m x p block.
        F(m*(i-1)+1:m*i, p*(j-1)+1:p*j) = C*A^(t-j)*B;
    end
end

Y = [y1; y2; y3; y4];
Uln = pinv(F)*Y;
J1 = sum(Uln.^2)

% Do the regularized business. Use bisection.
objv_l = sum(U.^2);
mu_l = 0;
mu_u = 100;
for i = 1:20
    mu_m = 0.5*(mu_l + mu_u);
    Ureg = (F'*F + mu_m*eye(T4*p)) \ (F'*Y);

    if sum(Ureg.^2) > 1
        mu_l = mu_m;
    else
        mu_u = mu_m;
    end
    disp([mu_m sum(Ureg.^2)]);
end
J2 = sum((Y - F*Ureg).^2)

% Plot both solutions.
U = reshape(Uln, 2, T4);
vehicle_plot;
```

```
print -depsc2 vehicle_ln.eps

U = reshape(Ureg, 2, T4);
vehicle_plot;
print -depsc2 vehicle_reg.eps
```

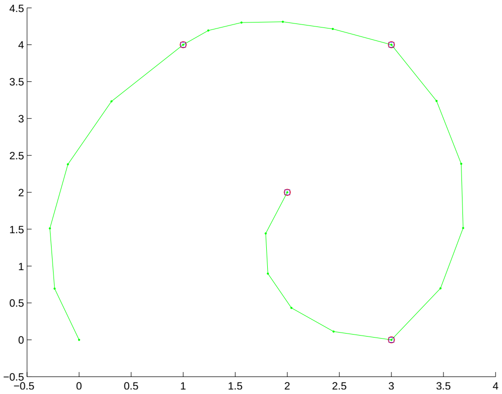
(d) Among all sets of inputs that reach the waypoints at the required times, we chose the ones that gave the smallest J_1. Yet this $J_1 = 1.24$. Thus, there is no set of inputs that reaches the waypoints and still satisfies $J_1 \leq 1$.

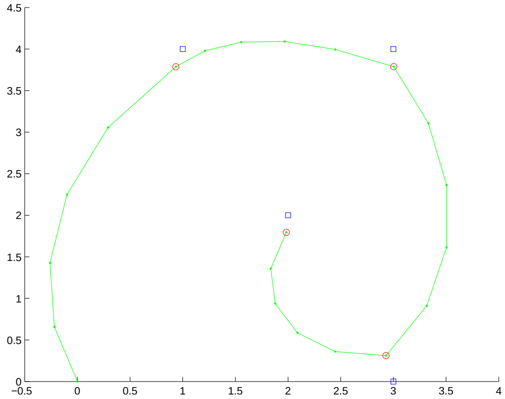
(e) This is now a regularized least-squares problem. We will adjust μ while minimizing

$$\|FU - Y\|^2 + \mu\|U\|^2,$$

until we find a μ so that $\|U\|^2 = 1$. That will give the required set of inputs. Matlab code that does this appears in part (c).

(f) We get $J_2 = 0.2387$ (with $\mu = 2.1714$). Plots of the two vehicle trajectories appear below.





3. **State estimation.** (20 points)

You wish to carry out state estimation for a chemical process. Each state component is the quantity of a certain chemical. After placing them in a mixing tank at time $t = 0$, they satisfy a discrete-time autonomous linear dynamical system, with

$$x(t+1) = Ax(t), \quad y(t) = Cx(t) + v(t),$$

where $v(t)$ is a small amount of measurement noise.

Unfortunately, the chemicals are added in unknown quantities. Your job is to estimate how much of each chemical was added to the tank; that is, estimate $x(0)$. You will observe $T + 1$ measurements $y(0), \ldots, y(T)$.

(a) Start by assuming that $v(t) = 0$ for all t. Under what conditions can you uniquely determine $x(0)$? How soon can it be determined (ie, what is the minimum value of T)? Your answer may refer to the matrices A and C, and can include statements involving properties of those matrices.

(b) Now we will assume $v(t) \neq 0$. Suppose $T = 3$, and assume that this would be sufficient to determine $x(0)$ if $v(t) = 0$. How do you find an estimate $\hat{x}(0)$ of $x(0)$ which minimizes

$$J = \sum_{t=0}^{T} \|y(t) - C\hat{x}(t)\|^2,$$

where $\hat{x}(t+1) = A\hat{x}(t)$ for $t = 0, \ldots, T-1$? You can think of this estimate as 'minimizing the implied noise'.

(c) Now download `chemicals.m`, which defines A, C, and $y(0), \ldots, y(3)$. If we assumed that $v(0) = \cdots = v(3) = 0$, would there be sufficient information to determine $x(0)$? Why?

(d) Use this data, and your method from part (b), to find the estimate $\hat{x}(0)$ that minimizes J. What is your resulting J?

Solution.

(a) With $v(t) = 0$, we can write

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(T) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^T \end{bmatrix} x(0) = \mathcal{O}_T x(0).$$

We can uniquely determine $x(0)$ as long as $\mathcal{O}_T$ is full rank and skinny. The smallest T, then, is the smallest T for which $\mathcal{O}_T$ is full rank and skinny.

(b) This is a least squares problem: set

$$\hat{x}(0) = \mathcal{O}_T^\dagger \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(T) \end{bmatrix}.$$

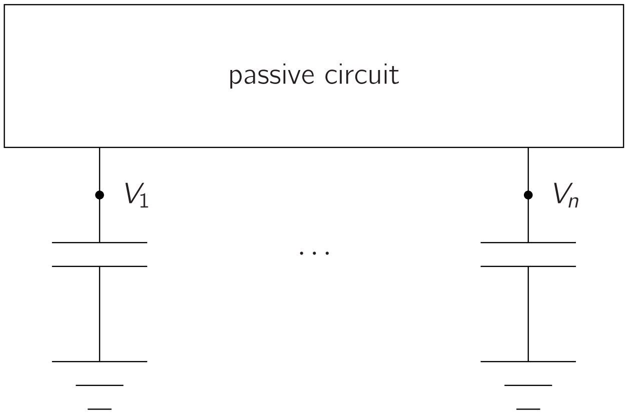
(c) We form $\mathcal{O}_3$, and confirm that it is full rank (see Matlab code below). Thus, there would be sufficient information to find $x(0)$.

(d) In Matlab, we use the following code. This gives $J = 0.0026$.

```
% Load data.
chemicals;

O3 = [C; C*A; C*A^2; C*A^3];

% Check that O3 has full column rank.
assert (rank(O3) == size(O3, 2), 'T = 3 is insufficient')

% Find the least squares approximate solution.
Y = [y0; y1; y2; y3];
x0hat = pinv(O3)*Y;

% Calculate J.
J = sum(Y - O3*x0hat).^2
```

4. **Voltage distribution.** (20 points)

Consider this small circuit, with n 1 Farad capacitors (supercaps were cheap that day), and n associated voltages $V_1, \ldots, V_n$.



The total energy stored in the capacitors at time t is given by

$$J(t) = (1/2)\sum_{i=1}^{n} V_n^2(t).$$

The circuit obeys the dynamics

$$\frac{dV(t)}{dt} = AV(t), \quad t \geq 0.$$

You will choose an initial voltage distribution $V(0)$, where the individual components $V_1(0), \ldots, V_n(0)$ represent the voltages across each of the capacitors at time $t = 0$. You're required to have 1 J of energy stored in the capacitors at time 0, ie, $J(0) = 1$. It's ok if some capacitor voltages are negative.

(a) Suppose you want to choose voltages to dissipate the energy stored in the capacitors as quickly as possible. In particular, how do you choose $V(0)$, with $J(0) = 1$, to minimize $J(10)$? How much energy will remain? (Your answer may involve A.)

(b) Suppose you want to choose voltages to dissipate the energy stored in the capacitors as slowly as possible. In particular, how do you choose $V(0)$, with $J(0) = 1$, to maximize $J(10)$? How much energy will remain? (Your answer may involve A.)

(c) Suppose we cared instead about the amount of energy stored at 20 seconds, $J(20)$. Would you choose the same answers as in parts (a) and (b)? Why or why not?

Solution.

(a) The voltages at time t are

$$V(t) = e^{tA}V(0),$$

so

$$J(t) = (1/2)\|e^{tA}V(0)\|^2.$$

Meanwhile, the initial energy is

$$J(0) = (1/2)\|V(0)\|^2.$$

We want to minimize $J(t)$ over all choices of $J(0) = 1$. We find the SVD of e^{10A}, and use v_n, the right singular vector associated with the smallest singular value. Set $V(0) = \sqrt{2}v_n$. The remaining energy will be σ_r^2 J.

(b) This proceeds in the same way as part (a), except we set $V(0) = \sqrt{2}v_1$. The remaining energy at $t = 10$ will be σ_1^2 J.

(c) In general, the right singular vectors of e^{tA}, $t \neq 10$, point in different directions to those of e^{10A}. Thus, you would not choose the same distributions. The reason for the difference is that over time, energy is transferred to and from the passive system.

5. **Monotone energy decrease.** (10 points)

Consider the simple circuit from question 4. The total energy stored in the capacitors is

$$J(t) = (1/2)\sum_{i=1}^{n} V_n^2(t),$$

and the circuit obeys the dynamics

$$\frac{dV(t)}{dt} = AV(t), \quad t \geq 0.$$

Under what conditions (on A) is the amount of energy stored in the capacitors monotone decreasing, for any initial voltage distribution? In other words, when can we guarantee that

$$\frac{dJ(t)}{dt} \leq 0, \quad t \geq 0,$$

for all $V(0)$?

Solution.

We start by writing $J(t) = (1/2)V(t)^T V(t)$. Then,

$$\frac{dJ(t)}{dt} = V(t)^T \frac{dV(t)}{dt} = V(t)^T AV(t) = (1/2)V(t)^T(A + A^T)V(t).$$

But this is the same as asking: when is $x^T(A + A^T)x \leq 0$ for all x. And thus we know the answer: the energy stored in the capacitors is monotone decreasing, for all x, if $A + A^T \leq 0$. We can't make any direct claims for a general $A \neq A^T$; hence the symmetrization.