EE363 homework 4 solutions

1. *Estimating an unknown constant from repeated measurements.* We wish to estimate $x \sim \mathcal{N}(0,1)$ from measurements $y_i = x + v_i$, $i = 1, \ldots, N$, where v_i are IID $\mathcal{N}(0, \sigma^2)$, uncorrelated with x. Find an explicit expression for the MMSE estimator $\hat{x}$, and the MMSE error.

   Solution:
   We write the relationship between the measurements y_i for $i = 1, \ldots, N$, and x in matrix form as
$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} x + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix}$$
   or more compactly as $y = Ax + v$ where $y = (y_1, \ldots, y_N)$, $A = \mathbf{1}$ and $v = (v_1, \ldots, v_N)$. Now we can use the formulas for the MMSE estimator in the linear measurements case to obtain
$$\begin{aligned} \hat{x} &= \bar{x} + \Sigma_x A^T (A \Sigma_x A^T + \Sigma_v)^{-1} (y - \bar{y}) \\ &= \mathbf{1}^T (\mathbf{1}\mathbf{1}^T + I\sigma_v^2)^{-1} y \\ &= (\mathbf{1}^T \mathbf{1} + \sigma_v^2)^{-1} \mathbf{1}^T y \\ &= \frac{\mathbf{1}^T y}{N + \sigma_v^2}. \end{aligned}$$
   For the error covariance we have
$$\begin{aligned} \Sigma_{\text{est}} &= \Sigma_x - \Sigma_x A^T (A \Sigma_x A^T + \Sigma_v)^{-1} A \Sigma_x \\ &= 1 - \mathbf{1}^T (\mathbf{1}\mathbf{1}^T + I\sigma_v^2)^{-1} \mathbf{1} \\ &= 1 - (\mathbf{1}^T \mathbf{1} + \sigma_v^2)^{-1} \mathbf{1}^T \mathbf{1} \\ &= \frac{\sigma_v^2}{N + \sigma_v^2}. \end{aligned}$$
   Note that as $N \to \infty$ (we perform many measurements), $\Sigma_{\text{est}} \to 0$, and as $\sigma \to \infty$ (very large noise), $\Sigma_{\text{est}} \to 1 = \Sigma_x$ (*i.e.*, our prior covariance of x). Both of these limiting cases make intuitive sense. In the first case by making many measurements we are able to estimate x exactly, and in the second case with very large noise, the measurements do not help in estimating x and we cannot improve the a priori covariance of x.

2. *Estimator error variance and correlation coefficient.* Suppose $(x, y) \in \mathbf{R}^2$ is Gaussian, and let $\hat{x}$ denote the MMSE estimate of x given y, and $\bar{x}$ denote the expected value of x. We define the relative mean square estimator error as $\eta = \mathbf{E}(\hat{x} - x)^2 / \mathbf{E}(\bar{x} - x)^2$.

Show that η can be expressed as a function of ρ, the correlation coefficient of x and y. Does your answer make sense?

Solution:
Since $x, y \in \mathbf{R}$ we have

$$\mathbf{E}(\hat{x} - x)^2 = \mathbf{Tr}\,\Sigma_{\text{est}} = \Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{xy}^T,$$

where

$$\Sigma_x = \sigma_x^2, \quad \Sigma_{xy} = \sigma_{xy}, \quad \Sigma_y = \sigma_y^2.$$

So we have

$$\mathbf{E}(\hat{x} - x)^2 = \sigma_x^2 - \frac{\sigma_{xy}^2}{\sigma_y^2}.$$

Of course we have $\mathbf{E}(\bar{x} - x)^2 = \sigma_x^2$, so

$$\eta = \frac{\mathbf{E}(\hat{x} - x)^2}{\mathbf{E}(\bar{x} - x)^2} = (\sigma_x^2 - \frac{\sigma_{xy}^2}{\sigma_y^2})/\sigma_x^2 = 1 - \left(\frac{\sigma_{xy}}{\sigma_x\sigma_y}\right)^2 = 1 - \rho^2.$$

This answer makes perfect sense. When x and y have strong (positive or negative) correlation, $|\rho|$ is close to one, and therefore η is close to zero, which mean that the relative minimum mean square estimation error is small. On the other hand, if x and y are almost uncorrelated, *i.e.*, $|\rho|$ is small, we find that $\eta \approx 1$, which mean that the minimum mean square error is close to the prior variance of x. In other words, when x and y are highly correlated, we can estimate x from y accurately, while when x and y are uncorrelated, the measurement y does not help at all in estimating x.

3. *MMSE predictor and interpolator.* A scalar time series $y(0), y(1), \ldots$ is modeled as

$$y(t) = a_0 w(t) + a_1 w(t-1) + \cdots + a_N w(t-N),$$

where $w(-N), w(-N+1), \ldots$ are IID $\mathcal{N}(0,1)$. The coefficients $a_0, \ldots, a_N$ are known.

   (a) *Predicting next value from current value.* Find the MMSE predictor of $y(t+1)$ based on $y(t)$. (Note: we really mean based on just $y(t)$, and not based on $y(t), y(t-1), \ldots$) Your answer should be as explicit as possible.

   (b) *MMSE interpolator.* Find the MMSE predictor of $y(t)$ (for $t > 1$) based (only) on $y(t-1)$ and $y(t+1)$ (for $t \geq 1$). Your answer should be as explicit as possible.

Solution:

   (a) *Predicting next value from current value.*
   We'll use the general expression for the MMSE estimator:

$$\begin{aligned}\hat{y}(t+1) &= \bar{y}(t+1) + \Sigma_{y(t+1)y(t)}\Sigma_{y(t)}^{-1}(y(t) - \bar{y}(t)) \\ &= \Sigma_{y(t+1)y(t)}\Sigma_{y(t)}^{-1}y(t)\end{aligned}$$

(since the w's are all zero mean, which implies the y's are all zero mean). Now we will find $\Sigma_{y(t+1)y(t)}$ and $\Sigma_{y(t)}$:

$$\begin{aligned}\Sigma_{y(t)} &= \mathbf{E}(a_0 w(t) + a_1 w(t-1) + \cdots + a_N w(t-N))^2 \\ &= \sum_{i=0}^{N} a_i^2\end{aligned}$$

using the fact that $\mathbf{E}\, w(t)w(s) = \delta_{t-s}$. Similarly we have

$$\begin{aligned}\Sigma_{y(t+1)y(t)} &= \mathbf{E}(a_0 w(t+1) + a_1 w(t) + \cdots + a_N w(t+1-N))(a_0 w(t) + \cdots + a_N w(t-N)) \\ &= \sum_{i=0}^{N-1} a_{i+1} a_i.\end{aligned}$$

Hence the MMSE estimator is:

$$\hat{y}(t+1) = \frac{\sum_{i=0}^{N-1} a_{i+1} a_i}{\sum_{i=0}^{N} a_i^2} y(t).$$

This expression makes sense: you just multiply what you just observed ($y(t)$) by a constant to predict $y(t+1)$. (The constant, by the way, is between zero and one.)

(b) *MMSE interpolator.* Define $z(t) = [y(t+1)\ y(t-1)]^T$. We want to find $\hat{y}(t) = \mathbf{E}(y(t)|z(t))$. We first find the required covariance matrices:

$$\Sigma_{y(t)z(t)} = \mathbf{E}\, y(t)[y(t+1)^T y(t-1)^T] = \left[\sum_{i=0}^{N-1} a_{i+1} a_i \quad \sum_{i=0}^{N-1} a_{i+1} a_i\right]$$

and

$$\Sigma_{z(t)} = \mathbf{E}[y(t+1)\ y(t-1)]^T [y(t+1)\ y(t-1)] = \begin{bmatrix} \sum_{i=0}^{N} a_i^2 & \sum_{i=0}^{N-2} a_{i+2} a_i \\ \sum_{i=0}^{N-2} a_{i+2} a_i & \sum_{i=0}^{N} a_i^2 \end{bmatrix}$$

Therefore the MMSE interpolator is

$$\hat{y}(t) = \left[\sum_{i=0}^{N-1} a_{i+1} a_i \quad \sum_{i=0}^{N-1} a_{i+1} a_i\right] \begin{bmatrix} \sum_{i=0}^{N} a_i^2 & \sum_{i=0}^{N-2} a_{i+2} a_i \\ \sum_{i=0}^{N-2} a_{i+2} a_i & \sum_{i=0}^{N} a_i^2 \end{bmatrix}^{-1} \begin{bmatrix} y(t+1) \\ y(t-1) \end{bmatrix}$$

We find the inverse of the 2×2 matrix $\Sigma_{z(t)}$ and multiply out to obtain the final result:

$$\hat{y}(t) = \frac{\sum_{i=0}^{N-1} a_{i+1} a_i}{\sum_{i=0}^{N} a_i^2 + \sum_{i=0}^{N-2} a_{i+2} a_i} \left[y(t+1) + y(t-1)\right].$$

So the MMSE interpolator takes the average of the two observations $y(t-1)$ and $y(t+1)$, and multiplies it by a constant.

4. *Estimating initial subpopulations from total growth observations.* A sample that contains three types of bacteria (called A, B, and C) is cultured, and the total bacteria population is measured every hour. The bacteria populations grow, independently of each other, exponentially with different growth rates: A grows 2% per hour, B grows 5% per hour, and C grows 10% per hour. The goal is to estimate the initial bacteria populations based on the measurements of total population.

   Let $x_A(t)$ denote the population of bacteria A after t hours (say, measured in grams), for $t = 0, 1, \ldots$, and similarly for $x_B(t)$ and $x_C(t)$, so that

$$x_A(t+1) = 1.02 x_A(t), \quad x_B(t+1) = 1.05 x_B(t), \quad x_C(t+1) = 1.10 x_C(t).$$

   The total population measurements are $y(t) = x_A(t) + x_B(t) + x_C(t) + v(t)$, where $v(t)$ are IID, $\mathcal{N}(0, 0.25)$. (Thus the total population is measured with a standard deviation of 0.5).

   The prior information is that $x_A(0)$, $x_B(0)$, $x_C(0)$ (which are what we want to estimate) are IID $\mathcal{N}(5, 2)$. (Obviously the Gaussian model is not completely accurate since it allows the initial populations to be negative with some small probability, but we'll ignore that.)

   How long will it be (in hours) before we can estimate $x_A(0)$ with a mean square error less than 0.01? How long for $x_B(0)$? How long for $x_C(0)$?

   Solution:
   After t hours we have made $t+1$ measurements:

$$\begin{bmatrix} y(0) \\ \vdots \\ y(t) \end{bmatrix} = F(t) \begin{bmatrix} x_A(0) \\ x_B(0) \\ x_C(0) \end{bmatrix} + \begin{bmatrix} v(0) \\ \vdots \\ v(t) \end{bmatrix}, \quad F(t) = \begin{bmatrix} 1 & 1 & 1 \\ 1.02 & 1.05 & 1.10 \\ 1.02^2 & 1.05^2 & 1.10^2 \\ & \vdots & \\ 1.02^t & 1.05^t & 1.10^t \end{bmatrix}.$$

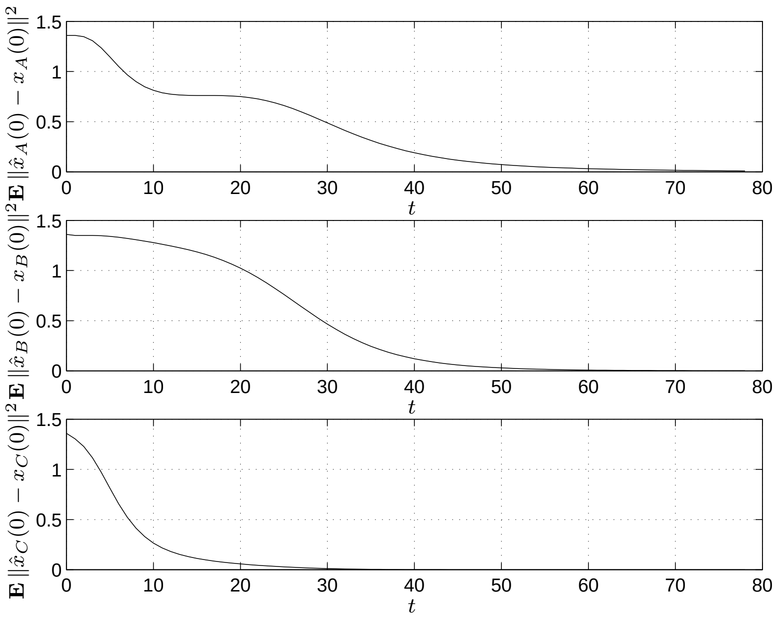
   The covariance of the noise vector $[v(0) \ \cdots \ v(N)]^t$ is $0.25I$. The prior covariance of $[x_A(0) \ x_B(0) \ x_C(0)]^t$ is $2I$.

   Hence the covariance of the estimation error is

$$\Sigma_{\text{est}}(t) = (F(t)^T (0.25I)^{-1} F(t) + (2I)^{-1})^{-1} = (4F(t)^T F(t) + 0.5I)^{-1}.$$

   The mean-square error in estimating $x_A(0)$ is given by the $(1,1)$ entry of Σ_{est}; for B and C it is given by the other diagonal elements. So what we need to do is to find how large t has to be before the diagonal elements of $\Sigma_{\text{est}}(t)$ become less than 0.01. This can be done by plotting, or any other method. The plot below shows the mean-square error in estimating $x_A(0)$, $x_B(0)$, and $x_C(0)$, as a function of t. The shape of the plot makes good intuitive sense, if you think about it long enough. x_C grows most rapidly so it is not surprising that we can estimate $x_A(0)$ accurately more quickly than the other two. (If anyone can think of an intuitive explanation of the flat part between $t = 10$ and $t = 20$ in the estimation error for $x_A(0)$, I'd like to hear it!)

The solution is: 79 hours for $x_A(0)$, 60 hours for $x_B(0)$, and 32 hours for $x_C(0)$.



```
A=diag([1.02 1.05 1.1]); C=[1 1 1];
sigma=0.5; S_x=2*eye(3);;

AA=eye(3); O=[]; a=[]; b=[]; c=[];
t=0;
while 1
   O=[O;C*AA];
   AA=A*AA;
   S_est=inv(0.5*eye(3)+O'*O/sigma^2);
   a=[a;S_est(1,1)];
   b=[b;S_est(2,2)];
   c=[c;S_est(3,3)];
   if max([a(t+1),b(t+1),c(t+1)]) <= 0.01
      break
   end
   t=t+1;
end

% plot MMSE estimation errors
clg
subplot(3,1,1)
```

```
plot(linspace(0,t,t+1),a); grid on; xlabel('t');ylabel('A')
subplot(3,1,2)
plot(linspace(0,t,t+1),b); grid on; xlabel('t');ylabel('B')
subplot(3,1,3)
plot(linspace(0,t,t+1),c); grid on; xlabel('t');ylabel('C')

print -deps bacteria
```

5. *Sensor selection.* Suppose 5 scalar measurements $y_1, \ldots, y_5$ are related to 3 scalar variables $x_1,\ x_2,\ x_3$ by

$$y = Ax + v, \quad A = \begin{bmatrix} 1 & 2 & 3 \\ -1 & 1 & 3 \\ 1 & -2 & 0 \\ -1 & 0 & -3 \\ 1 & 2 & -3 \end{bmatrix},$$

where v_i are IID $\mathcal{N}(0, 0.01)$. Thus the measurement errors have standard deviation 0.1. The variable $x \in \mathbf{R}^3$ is deterministic but unknown.

You are to find an estimator $\hat{x} = By$ that satisfies the following specifications:

- $BA = I$. This means that $\hat{x} = x$ when $v = 0$, and also that the estimator is unbiased, *i.e.*, $\mathbf{E}\,\hat{x} = x$.
- $\mathbf{E}\,\|\hat{x} - x\|^2 \leq 0.005$.

Among all matrices B that satisfy these specifications, you are to find one that minimizes the number of nonzero columns.

To understand the practical meaning of this, note that if the jth column of B is zero, then the estimate $\hat{x}$ does not use the jth measurement, so the jth sensor is not needed. Thus we are seeking the smallest sensor configuration (in terms of number of sensors required) that can be used to estimate x with mean square error not exceeding 0.005.

Make sure to describe exactly how you are going to solve the problem, as well as giving your explicit solution (*i.e.*, B). If no B satisfies the specifications, say so, and show why.

Solution: To say that $BA = I$, where B has some columns equal to zero, means that $\tilde{B}\tilde{A} = I$, where $\tilde{B}$ is B with the zero columns removed, and $\tilde{A}$ is A with the corresponding rows removed. This tells us immediately that we have to have at least three nonzero columns (which is kind of obvious) in order to have $BA = I$. The possible configurations are therefore the three sensor ones:

$$123,\quad 124,\quad 125,\quad 134,\quad 135,\quad 145,\quad 234,\quad 235,\quad 245,\quad 345,$$

(where 123 means the 4th and 5th columns of B are zero), the four sensor configurations:

$$1234, \quad 1235, \quad 1245, \quad 1345, \quad 2345,$$

and one five sensor configuration: 12345. All together that makes 16 possible sensor configurations, *i.e.*, patterns of zero columns.

Since $BA = I$, we have

$$\hat{x} - x = By - x = B(Ax + v) - x = Bv,$$

so we can express the mean square error as

$$\mathbf{E}\,\|\hat{x} - x\|^2 = \mathbf{Tr}(B\Sigma_v B^T) = 0.01\,\mathbf{Tr}(BB^T) = 0.01\,\mathbf{Tr}(\tilde{B}\tilde{B}^T).$$

Once we have chosen a pattern, which fixes $\tilde{A}$, the best $\tilde{B}$ is evidently the pseudo-inverse, *i.e.*, $\tilde{B} = \tilde{A}^\dagger = (\tilde{A}^T\tilde{A})^{-1}\tilde{A}^T$. This $\tilde{B}$ is best in the sense that any other with the same pattern of zero columns will have mean square estimation error larger (or equal). The MS estimation error can be expressed as

$$0.01\,\mathbf{Tr}\,\tilde{B}\tilde{B}^T = 0.01\,\mathbf{Tr}\left((\tilde{A}^T\tilde{A})^{-1}\tilde{A}^T A(\tilde{A}^T\tilde{A})^{-1}\right) = 0.01\,\mathbf{Tr}(\tilde{A}^T\tilde{A})^{-1}$$

(where $\tilde{A}$, recall, is A with the appropriate rows removed).

Hence we need to evaluate the mean square estimation error achieved by all fifteen patterns, and find those that are less than the specified limit 0.005. With Matlab this is not a problem; we can define 16 matrices with nice mnemonic names such as `A124 = A([1 2 4],:)`. Then we can find the minimum mean square estimation error as `MMSE124 = 0.01*trace(inv(A124'*A124))`. We get the results shown below:

columns	MMSE
123	0.0072
124	0.0097
125	0.0100
134	0.0417
135	0.0052
145	0.0206
234	0.0174
235	0.0243
245	0.0065
345	0.0057
2345	0.0045
1345	0.0043
1245	0.0052
1235	0.0041
1234	0.0065
12345	0.0031

From this table we see that none of the three-sensor configurations achieve the spec (although 135 comes pretty close); and three of the six different four-sensor configurations do (2345, 1345, and 1235). Hence the minimum number of sensors is 4, and the only possible choices are 2345, 1345 and 1235. As a specific answer, for example, we can take the pseudo-inverse that results when the fourth row of A is deleted:

$$B = \begin{bmatrix} 0.3284 & -0.2090 & 0.3433 & 0 & 0.1194 \\ 0.0970 & 0.0746 & -0.1940 & 0 & 0.1716 \\ 0.1368 & 0.0796 & 0.0597 & 0 & -0.1169 \end{bmatrix}.$$

Of course, you don't have to use the pseudo-inverse of $\tilde{B}$; you only have to ensure that $BA = I$ and $0.01\,\mathbf{Tr}(BB^T) \le 0.005$. (On the other hand there is no reason not to use the pseudo-inverse.)

This problem gives an example of *experiment design.* You can think of the the five sensors as describing five (potential) experiments that can be carried out, each with some cost. Then the problem can be interpreted as asking you to decide which of the five experiments you would carry out, in order to meet the MMSE condition and also minimize cost (*i.e.*, number of experiments actually carried out).

```
A=[1  2  3
-1  1  3
1 -2  0
-1  0 -3
1  2 -3];

s=0.1;  % standard deviation of sensor noise
z=zeros(1,3);

% using 3 sensors
AA=[A(1,:);A(2,:);A(3,:);z;z]; B=inv(AA'*AA)*AA';
mmse123=trace(B*B'*s^2) AA=[A(1,:);A(2,:);z;A(4,:);z];
B=inv(AA'*AA)*AA'; mmse124=trace(B*B'*s^2)
AA=[A(1,:);A(2,:);z;z;A(5,:)]; B=inv(AA'*AA)*AA';
mmse125=trace(B*B'*s^2) AA=[A(1,:);z;A(3,:);A(4,:);z];
B=inv(AA'*AA)*AA'; mmse134=trace(B*B'*s^2)
AA=[A(1,:);z;A(3,:);z;A(5,:)]; B=inv(AA'*AA)*AA';
mmse135=trace(B*B'*s^2) AA=[A(1,:);z;z;A(4,:);A(5,:)];
B=inv(AA'*AA)*AA'; mmse145=trace(B*B'*s^2)
AA=[z;A(2,:);A(3,:);A(4,:);z]; B=inv(AA'*AA)*AA';
mmse234=trace(B*B'*s^2) AA=[z;A(2,:);A(3,:);z;A(5,:)];
B=inv(AA'*AA)*AA'; mmse235=trace(B*B'*s^2)
AA=[z;A(2,:);z;A(4,:);A(5,:)]; B=inv(AA'*AA)*AA';
```

```
mmse245=trace(B*B'*s^2) AA=[z;z;A(3,:);A(4,:);A(5,:)];
B=inv(AA'*AA)*AA'; mmse345=trace(B*B'*s^2)

% using 4 sensors
AA=[A(1,:);A(2,:);A(3,:);A(4,:);z]; B=inv(AA'*AA)*AA';
mmse1234=trace(B*B'*s^2) AA=[A(1,:);A(2,:);A(3,:);z;A(5,:)];
B=inv(AA'*AA)*AA'; mmse1235=trace(B*B'*s^2)
AA=[A(1,:);z;A(3,:);A(4,:);A(5,:)]; B=inv(AA'*AA)*AA';
mmse1345=trace(B*B'*s^2) AA=[A(1,:);A(2,:);z;A(4,:);A(5,:)];
B=inv(AA'*AA)*AA'; mmse1245=trace(B*B'*s^2)
AA=[z;A(2,:);A(3,:);A(4,:);A(5,:)]; B=inv(AA'*AA)*AA';
mmse2345=trace(B*B'*s^2)

% using 5 sensors
B=inv(AA'*AA)*AA'; mmse12345=trace(B*B'*s^2)
```

6. *MMSE estimation example.* We wish to estimate $x \in \mathbf{R}^n$, given a set of measurements $y \in \mathbf{R}^m$, where

$$y = Ax + v, \qquad x \sim \mathcal{N}(\bar{x}, \Sigma_x), \qquad v \sim \mathcal{N}(0, \Sigma_v),$$

with x and v independent. We'll consider the specific case with $n = 4$, $m = 6$, and measurement matrix

$$A = \begin{bmatrix} 2 & 3 & -1 & 4 \\ 1 & 0 & 0 & -2 \\ 2 & 1 & 1 & 0 \\ -3 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 \end{bmatrix}.$$

The prior distribution of x is given by $\bar{x} = (2, -1, 0.5, -3)$,

$$\sigma_1^2 = 2, \quad \sigma_2^2 = 5, \quad \sigma_3^2 = 1, \quad \sigma_4^2 = 2,$$

and

$$\rho_{12} = -0.1, \quad \rho_{13} = 0.025, \quad \rho_{23} = -0.01.$$

Here σ_i is the standard deviation of x_i, and ρ_{ij} is the correlation coefficient between x_i and x_j; correlation coefficients not given are zero. The noise statistics are characterized by standard deviations

$$\tilde{\sigma}_1^2 = 2, \quad \tilde{\sigma}_2^2 = 1, \quad \tilde{\sigma}_3^2 = 2, \qquad \tilde{\sigma}_4^2 = 3, \qquad \tilde{\sigma}_5^2 = 2, \qquad \tilde{\sigma}_6^2 = 1,$$

and correlation coefficients

$$\tilde{\rho}_{13} = -0.25, \quad \tilde{\rho}_{24} = 0.5, \quad \tilde{\rho}_{35} = 0.3, \quad \tilde{\rho}_{46} = -0.04,$$

with other correlation coefficients zero.

In this problem you will compare the performance of two estimators. The first is the simple pseudo-inverse estimator, $\hat{x}_{\text{pinv}} = A^{\dagger}y$, (as studied in EE263). The second is the MMSE estimator, denoted $\hat{x}_{\text{mmse}}$. You will first make some predictions about the performance of the two estimators, based on some linear algebra calculations; then, you will carry out simulations of the two to verify your calculations.

(a) Find $\mathbf{E}\,\|\hat{x}_{\text{pinv}}-x\|^2$ and $\mathbf{E}\,\|\hat{x}_{\text{mmse}}-x\|^2$, explaining your method. Briefly comment on the results.

(b) Let $\mathcal{E} = \{\hat{x}-x \mid (\hat{x}-x)^T\Sigma^{-1}(\hat{x}-x) \le \alpha\}$ be the 90% confidence ellipsoid for the MMSE estimation error $\hat{x}_{\text{mmse}}-x$. Find Σ and α.

(c) Generate 1000 samples of (x,v) pairs from the distributions described above. For each sample, find $\hat{x}_{\text{mmse}}-x$ and $\hat{x}_{\text{pinv}}-x$. Plot the empirical distributions (histograms) of $\|\hat{x}_{\text{mmse}}-x\|^2$ and $\|\hat{x}_{\text{pinv}}-x\|^2$. (Plot the histograms on the same horizontal axis, so they are easier to compare.) Verify that the empirical averages of these are close to values predicted in part (a).

(d) For how many of your 1000 samples from part (c) does $\hat{x}_{\text{mmse}}-x$ fall inside the 90% confidence ellipsoid $\mathcal{E}$ found in part (b)? (Obviously this number should be near 900. If the number turns out to exactly 900, change your seed and run the simulations again.)

Some Matlab hints.

- `sqrtm(A)` finds the (matrix) square root of matrix `A`.
- `randn(n,1)` generates a column vector of dimension `n` drawn from a normal distribution $\mathcal{N}(0,I)$. (Do not confuse `randn` with `rand`, which draws the entries of the vector from a uniform distribution on $[0,1]$.)
- `hist(y,M)` bins the elements of `y` into `M` equally spaced bins and plots the results. (For 1000 samples, `M` around 50 gives a reasonable plot.)
- `chi2inv(p,n)` returns the inverse of the chi-square cumulative distribution with `n` degrees of freedom, at the value `p`.

Solution.

(a) Let $B_{\text{pinv}} = A^{\dagger}$ and $B_{\text{mmse}} = \Sigma_x A^T\left(A\Sigma_x A^T + \Sigma_v\right)^{-1}$ where Σ_x and Σ_v are the covariance of x and v respectively.

The pseudo-inverse estimator is

$$\hat{x}_{\text{pinv}} = B_{\text{pinv}}y = B_{\text{pinv}}(Ax+v) = x + A^{\dagger}v,$$

and so $\hat{x}_{\text{pinv}}-x = A^{\dagger}v$. The mean square error for this estimator is

$$\mathbf{E}\,\|\hat{x}_{\text{pinv}}-x\|^2 = \mathbf{E}\,\|A^{\dagger}v\|^2 = \mathbf{Tr}\,A^{\dagger}\Sigma_v A^{\dagger T}.$$

The MMSE estimator is

$$\hat{x}_{\text{mmse}} = \bar{x} + B_{\text{mmse}}(y - A\bar{x}),$$

with associated estimation error vector

$$\hat{x}_{\text{mmse}} - x \sim \mathcal{N}\left(0, \Sigma_x - \Sigma_x A^T \left(A\Sigma_x A^T + \Sigma_v\right)^{-1} A\Sigma_x\right).$$

Its mean square error is

$$\begin{aligned}\mathbf{E}\,\|\hat{x}_{\text{mmse}} - x\|^2 &= \mathbf{Tr}\left(\Sigma_x - \Sigma_x A^T \left(A\Sigma_x A^T + \Sigma_v\right)^{-1} A\Sigma_x\right) \\ &= \mathbf{Tr}\left((A^T\Sigma_v^{-1}A + \Sigma_x)^{-1}\right).\end{aligned}$$

Of course, the MMSE mean square error has to be smaller than the mean square error for the pseudo-inverse estimator, because among all estimators, the MMSE estimator is the one which minimizes the mean square error. We guess that it's possible to show this using a linear algebra argument, using the expressions above, but it's not an obvious calculation (at least for us).

(b) Since $\mathcal{E}$ is the 90% confidence ellipsoid of $\hat{x}_{\text{mmse}} - x$, then it is straightforward to see that Σ should be the covariance matrix of $\hat{x}_{\text{mmse}} - x$, i.e.

$$\Sigma = \Sigma_x - \Sigma_x A^T \left(A\Sigma_x A^T + \Sigma_v\right)^{-1} A\Sigma_x.$$

We want that $\mathbf{Prob}\,(\hat{x}_{\text{mmse}} - x \in \mathcal{E}) = 0.9$. Since $(\hat{x} - x)^T\Sigma(\hat{x} - x)$ is distributed according to a χ_n^2 distribution with cumulative distribution function $F_{\chi_n^2}$, then we need to have $F_{\chi_n^2}(\alpha) = 0.9$. Then $\alpha = F_{\chi_n^2}^{-1}(0.9) = 7.7794$ (you can use the MATLAB function `chi2inv` to evaluate $F_{\chi_n^2}^{-1}$).

(c) The simulations required for parts (c) and (d) are performed in the following MATLAB script

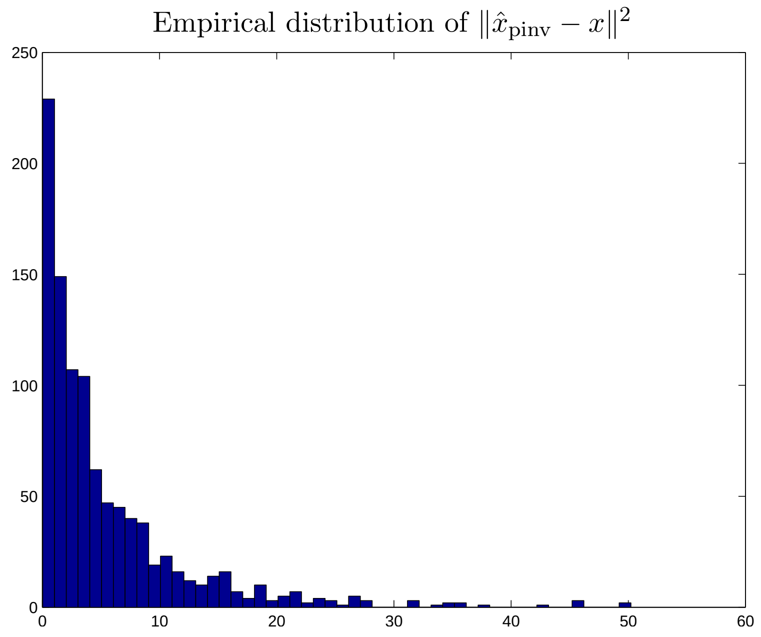
```
clear all; close all;

% dimensions
m = 6;
n = 4;

mean_x = [2 ; -1; 0.5; -3];                     % mean of x
mean_v = zeros(m,1);                            % mean of v
var_x = [2 5 1 2];                              % variances of x_i's
var_v = [2 1 2 3 2 1];                          % variances of v_i's

% constructing the covariance matrix of x
```

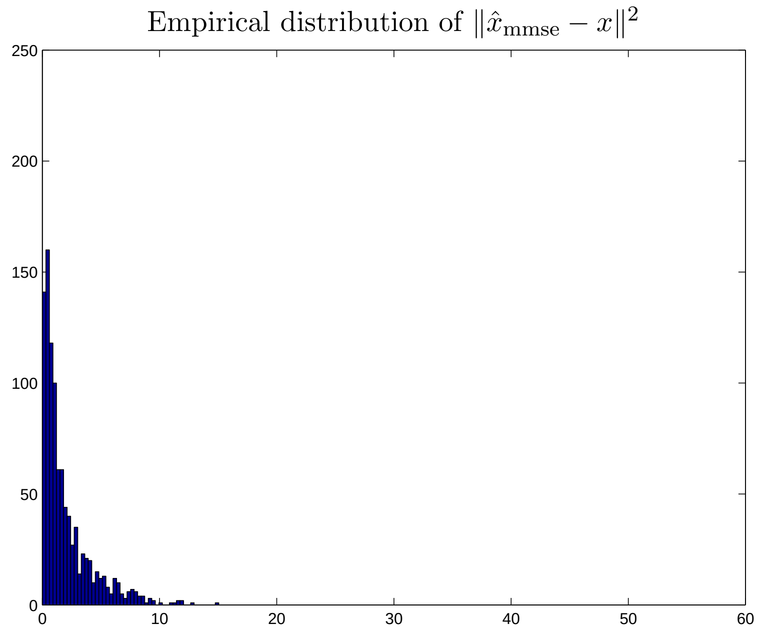
```
corr_x = zeros(n);
corr_x(1,2) = -0.1;     corr_x(1,3) = 0.025;    corr_x(2,3) = -0.01;
corr_x = corr_x + corr_x' + eye(n);
sigma_x = diag(sqrt(var_x))*corr_x*diag(sqrt(var_x));

% constructing the covariance matrix of v
corr_v = zeros(m);
corr_v(1,3) = -0.25;    corr_v(2,4) = 0.5;      corr_v(3,5) = 0.3;
corr_v(4,6) = -0.04;
corr_v = corr_v + corr_v' + eye(m);
sigma_v = diag(sqrt(var_v))*corr_v*diag(sqrt(var_v));

A = [  2  3 -1  4; ...
       1  0  0 -2; ...
       2  1  1  0; ...
      -3 -1  0  0; ...
       1  0  0 -1; ...
       0 -1  1  0];

% Pseudo-inverse estimator
B_pinv = pinv(A);

sigma_pinv_err = B_pinv*sigma_v*B_pinv';
pinv_mse = trace(sigma_pinv_err)

% MMSE estimator
B_mmse = sigma_x*A'*inv(A*sigma_x*A' + sigma_v);

mean_y = A*mean_x + mean_v;

sigma_mmse_err = sigma_x - sigma_x*A'*inv(A*sigma_x*A' + sigma_v)*A*sigma_x;
mmse = trace(sigma_mmse_err)

% generate N instances
N = 1000;
alpha = chi2inv(.9,n);
cnt = 0;                % to count the number of error vectors inside the
                        % the 90% confidence ellipsoid
for i = 1:N
    x = sqrtm(sigma_x)*randn(n,1) + mean_x;
    v = sqrtm(sigma_v)*randn(m,1) + mean_v;
    y = A*x +v;
    x_pinv = B_pinv*y;
    sq_err_pinv(i) = norm(x-x_pinv)^2;
    x_mmse = mean_x + B_mmse*(y - A*mean_x - mean_v);
    sq_err_mmse(i) = norm(x-x_mmse)^2;
    if (x-x_mmse)'*inv(sigma_mmse_err)*(x-x_mmse) <= alpha
        cnt = cnt + 1;
    end
end

mean_sq_error_pinv = mean(sq_err_pinv)
mean_sq_error_mmse = mean(sq_err_mmse)
cnt

figure;
hist(sq_err_pinv,50)
title('title1')
axis([0 60 0 250]);
print -deps mmse_example_pinv_dist
figure;
hist(sq_err_mmse,50)
title('title2')
axis([0 60 0 250]);
print -deps mmse_example_mmse_dist
```

Running the script yields the following results:

- The empirical mean square error obtained when using the pseudo-inverse estimator is 5.3025. It is very close to 5.2382 which is the theoretical mean square error for that case, found in part (a).
- The empirical mean square error obtained when using the MMSE estimator is 1.9051. It is very close to 1.9214 which is the theoretical mean square error for that case, found in part (a).
- The empirical mean square error obtained when using the pseudo-inverse estimator is greater than the one obtained when using the MMSE estimator. This confirms the results obtained in part (a).

The empirical distributions of the mean square error are displayed, for each case, in the following two figures.





(d) In our simulation, 910 out of the 1000 MMSE error vectors fell inside $\mathcal{E}$.

7. *Cholesky decomposition and estimation.* Every symmetric positive definite matrix P can be decomposed as $P = LL^T$, with L a lower triangular matrix with diagonal entries $L_{ii} > 0$. This decomposition is unique and is called the *Cholesky decomposition* of P. The Cholesky decomposition comes up in many contexts, such as solving linear equations with symmetric positive definite coefficient matrix, and least-squares and least-norm problems. In this problem we explore the connection between the Cholesky factorization and estimation. Let $x \sim \mathcal{N}(0, \Sigma)$ and let $\Sigma = LL^T$ be the Cholesky decomposition of Σ.

Notation: We use $k:l$ to denote a range of integers, $k, k+1, \ldots, l$. If $z \in \mathbf{R}^n$ is a vector, we let $z_{k:l}$ denote the subvector with the corresponding entries, *i.e.*, $(z_k, \ldots, z_l)$. (We assume here that $1 \leq k \leq l \leq n$.) For a matrix A, we use $A_{k:l,p:q}$ to denote the submatrix of A consisting of rows from k to l and the columns from p to q. For example, $A_{1:i,1:i}$ denotes the leading (top left) $i \times i$ submatrix of A.

(a) Show that $L_{ii}^2 = \mathbf{E}\left(x_i - \mathbf{E}(x_i|x_{1:i-1})\right)^2$. Thus, L_{ii} is the standard deviation of the error in estimating x_i, given $x_1, \ldots, x_{i-1}$.

(b) Let $y = x_{i:n} - \mathbf{E}(x_{i:n}|x_{1:i-1})$ be the estimation error in predicting the last $n-i+1$ components of x, given the first $i-1$ components. Show that $\mathbf{E}\,\|y\|^2 = \|L_{i:n,i:n}\|_F^2$. (Here $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, defined as the squareroot of the sum of the squares of its components.)

Hint: The Cholesky factorization of $\Sigma_{1:i,1:i}$ is $L_{1:i,1:i}L_{1:i,1:i}^T$.

Solution:

(a) It is easy to see that $\Sigma_{1:i,1:i} = L_{1:i,1:i}L_{1:i,1:i}^T$. We can partition the matrices in this equation as follows and get

$$\begin{bmatrix} \Sigma_{1:i-1,1:i-1} & s \\ s^T & \Sigma_{ii} \end{bmatrix} = \begin{bmatrix} L_{1:i-1,1:i-1} & 0 \\ l^T & L_{ii} \end{bmatrix} \begin{bmatrix} L_{1:i-1,1:i-1}^T & l \\ 0 & L_{ii} \end{bmatrix},$$

where $s = \Sigma_{1:i-1,i}$ and $l = L_{1:i-1,i}$. Note that Σ_{ii} and L_{ii} are scalars, and $L_{1:i-1}$ is lower triangular.

Hence we have

$$s = L_{1:i-1,1:i-1}l.$$

Since $L_{ii} > 0$ then $L_{1:i-1,1:i-1}$ is invertible and

$$l = L_{1:i-1,1:i-1}^{-1}s.$$

Since

$$\Sigma_{ii} = ll^T + L_{ii}^2,$$

then

$$L_{ii}^2 = \Sigma_{ii} - ll^T = \Sigma_{ii} - s^T L_{1:i-1,1:i-1}^{-T} L_{1:i-1,1:i-1}^{-1} s.$$

Since

$$\Sigma_{1:i-1,1:i-1} = L_{1:i-1,1:i-1}L_{1:i-1,1:i-1}^T,$$

then

$$\Sigma_{1:i-1,1:i-1}^{-1} = L_{1:i-1,1:i-1}^{-T} L_{1:i-1,1:i-1}^{-1}$$

and

$$L_{ii}^2 = \Sigma_{ii} - s^T \Sigma_{1:i-1,1:i-1}^{-1} s.$$

$\mathbf{E}(x_i|x_{1:i-1})$ is the MMSE estimate of x_i given $x_1, \ldots, x_{i-1}$. Then $\mathbf{E}\left(x_i - \mathbf{E}(x_i|x_{1:i-1})\right)^2$ is just the MMSE error variance and

$$\mathbf{E}\left(x_i - \mathbf{E}(x_i|x_1, \ldots, x_{i-1})\right)^2 = \Sigma_{ii} - s^T \Sigma_{1:i-1,1:i-1}^{-1} s.$$

Hence we have showed that

$$L_{ii}^2 = \mathbf{E}\left(x_i - \mathbf{E}(x_i|x_{1:i-1})\right)^2.$$

(b) We know that $\Sigma = LL^T$. We can partition the matrices in this relation in the following way

$$\begin{bmatrix} \Sigma_{1:i-1,1:i-1} & S \\ S^T & \Sigma_{i:n,i:n} \end{bmatrix} = \begin{bmatrix} L_{1:i-1,1:i-1} & 0 \\ \tilde{L}^T & L_{i:n,i:n} \end{bmatrix} \begin{bmatrix} L_{1:i-1,1:i-1}^T & \tilde{L} \\ 0 & L_{i:n,i:n} \end{bmatrix},$$

where $S = \Sigma_{1:i-1,i:n}$, $\tilde{L} = L_{1:i-1,i:n}$, and $L_{1:i-1,1:i-1}$ and $L_{i:n,i:n}$ are lower triangular. Hence we have

$$S = L_{1:i-1,1:i-1}\tilde{L}.$$

Since $L_{ii} > 0$ then $L_{1:i-1,1:i-1}$ is invertible and

$$\tilde{L} = L_{1:i-1,1:i-1}^{-1} S.$$

Since

$$\Sigma_{i:n,i:n} = \tilde{L}\tilde{L}^T + L_{i:n,i:n}L_{i:n,i:n}^T,$$

then

$$L_{i:n,i:n}L_{i:n,i:n}^T = \Sigma_{i:n,i:n} - \tilde{L}\tilde{L}^T = \Sigma_{i:n,i:n} - S^T L_{1:i-1,1:i-1}^{-T} L_{1:i-1,1:i-1}^{-1} S.$$

Since

$$\Sigma_{1:i-1,1:i-1} = L_{1:i-1,1:i-1}L_{1:i-1,1:i-1}^T,$$

then

$$\Sigma_{1:i-1,1:i-1}^{-1} = L_{1:i-1,1:i-1}^{-T} L_{1:i-1,1:i-1}^{-1}$$

and

$$L_{i:n,i:n}L_{i:n,i:n}^T = \Sigma_{i:n,i:n} - S^T \Sigma_{1:i-1,1:i-1}^{-1} S.$$

$\mathbf{E}(x_{i:n}|x_{1:i-1})$ is the MMSE estimate of $x_i, \ldots, x_n$ given $x_1, \ldots, x_{i-1}$ and y is the MMSE estimation error vector, with covariance matrix

$$\Sigma_y = \Sigma_{i:n,i:n} - S^T \Sigma_{1:i-1,1:i-1}^{-1} S = L_{i:n,i:n}L_{i:n,i:n}^T.$$

Then the estimation error is

$$\mathbf{E}\,\|y\|^2 = \mathbf{Tr}\,\Sigma_y = \mathbf{Tr}\,L_{i:n,i:n}L_{i:n,i:n}^T = \|L_{i:n,i:n}\|_F^2$$

8. *Hadamard product.* The Hadamard product (also called the elementwise product) of two matrices $F \in \mathbf{R}^{p\times q}$, $G \in \mathbf{R}^{p\times q}$, denoted $H = F \circ G$, is defined by $H_{ij} = F_{ij}G_{ij}$. (The Hadamard product of two vectors is usually called the elementwise product.)

   Suppose x and y are independent random vectors in $\mathbf{R}^n$, with means $\mathbf{E}\, x = \bar{x}$, $\mathbf{E}\, y = \bar{y}$, and second moments $\mathbf{E}\, xx^T = X$, $\mathbf{E}\, yy^T = Y$, respectively. Define $z = x \circ y$. Show that $\mathbf{E}\, z = \bar{x} \circ \bar{y}$, and $\mathbf{E}\, zz^T = X \circ Y$.

   Remark. This implies that the Hadamard product of two positive semidefinite matrices is positive semidefinite, which is not obvious.

   Solution. To show that $\mathbf{E}(x \circ y) = \bar{x} \circ \bar{y}$, we note that

$$\mathbf{E}\, z_i = \mathbf{E}\, x_i y_i = \mathbf{E}\, x_i\, \mathbf{E}\, y_i = \bar{x}_i \bar{y}_i.$$

   (The middle inequality follows from independence of x_i and y_i.) In a similar way we have

$$\mathbf{E}\, z_i z_j = \mathbf{E}\, x_i y_i x_j y_j = (\mathbf{E}\, x_i x_j)(\mathbf{E}\, y_i y_j) = X_{ij} Y_{ij}.$$