Homework 6

Don’t forget to include a print copy of your Julia code where required.

1. Some properties of bi-objective least squares. Consider the bi-objective least-squares problem with objectives $J_i(x) = \| A_i x + b_i \|^2$, $i = 1, 2$. Let $\hat{x}(\lambda)$ denote the minimizer of $J_1(x) + \lambda J_2(x)$, for $\lambda > 0$. (We assume the columns of the stacked matrix are linearly independent.) We define $J^*_i(\lambda) = J_i(\hat{x}(\lambda))$, for $i = 1, 2$, the value of the two objectives as a function of the weight parameter. The optimal trade-off curve is the set of points $(J^*_1(\lambda), J^*_2(\lambda))$, as $\lambda$ varies over all positive numbers.

(a) Effect of weight on objectives in a bi-objective problem. Show the following: for $\lambda < \mu$, we have $J^*_1(\lambda) \leq J^*_1(\mu)$ and $J^*_2(\lambda) \geq J^*_2(\mu)$, which means if you increase the weight (on the second objective), the second objective goes down (or stays the same), and the first objective goes up (or stays the same). This means that the trade-off curve always slopes downward.

Hint. Resist the urge to write out any equations or formulas. Use the fact that $\hat{x}(\lambda)$ minimizes $J_1(x) + \lambda J_2(x)$, and similarly for $\hat{x}(\mu)$, to deduce the inequalities

$$J^*_1(\mu) + \lambda J^*_2(\mu) \geq J^*_1(\lambda) + \lambda J^*_2(\lambda), \quad J^*_1(\lambda) + \mu J^*_2(\lambda) \geq J^*_1(\mu) + \mu J^*_2(\mu).$$

Use these to show $J^*_1(\lambda) \leq J^*_1(\mu)$ and $J^*_2(\lambda) \geq J^*_2(\mu)$.

(b) Slope of the trade-off curve. The slope of the trade-off curve at the point $(J^*_1(\lambda), J^*_2(\lambda))$ is given by

$$S = \lim_{\mu \to \lambda} \frac{J^*_2(\mu) - J^*_2(\lambda)}{J^*_1(\mu) - J^*_1(\lambda)}. $$

(This limit is the same if $\mu$ approaches $\lambda$ from below or from above.) Show that $S = -1/\lambda$. This gives another interpretation of the parameter $\lambda$: $(J^*_1(\lambda), J^*_2(\lambda))$ is the point on the trade-off curve that has slope $-1/\lambda$.

Hint. First assume that $\mu$ approaches $\lambda$ from above (meaning, $\mu > \lambda$), and use the inequalities in the hint for problem 1 to show that $S \geq -1/\lambda$. Then assume that $\mu$ approaches $\lambda$ from below, and show that $S \leq -1/\lambda$.

2. Auto-regressive time series prediction. Suppose that $x$ is an $N$-vector representing time series data. The (one step ahead) prediction problem is to guess $x_{t+1}$, based on $x_1, \ldots, x_t$. We will base our prediction $\hat{x}_{t+1}$ of $x_{t+1}$ on the previous $M$ values, $x_t, x_{t-1}, \ldots, x_{t-M+1}$. (The number $M$ is called the memory length of our predictor.) When the prediction is a linear function,

$$\hat{x}_{t+1} = \beta_1 x_t + \beta_2 x_{t-1} + \cdots + \beta_M x_{t-M+1},$$
it is called an auto-regressive predictor. (It is possible to add an offset to $\hat{x}_{t+1}$, but we will leave it out for simplicity.) Of course we can only use our auto-regressive predictor for $M \leq t \leq N - 1$.

Some very simple and natural predictors have this form. One example is the predictor $\hat{x}_{t+1} = x_t$, which guesses that the next value is the same as the current one. Another one is $\hat{x}_{t+1} = x_t + (x_t - x_{t-1})$, which guesses what $x_{t+1}$ is by extrapolating a line that passes through $x_t$ and $x_{t-1}$.

We judge a predictor (i.e., the choice of coefficients $\beta_i$) by the mean-square prediction error

$$J = \frac{1}{N - M} \sum_{t=M}^{N-1} (\hat{x}_{t+1} - x_{t+1})^2.$$ 

A sophisticated choice of the coefficients $\beta_i$ is the one that minimizes $J$. We will call this the least-squares auto-regressive predictor.

(a) Find the matrix $A$ and the vector $b$ for which $J = \|A\beta - b\|^2/(N - M)$. This allows you to find the coefficients that minimize $J$, i.e., the auto-regressive predictor that minimizes the mean-square prediction error on the given time series. Be sure to give the dimensions of $A$ and $b$.

(b) For $M = 2, \ldots, 12$, find the coefficients that minimize the mean-square prediction error on the time series $x_{\text{train}}$ given in time_series_data.jl. The same file has a second time series $x_{\text{test}}$ that you can use to test or validate your predictor on. Give the values of the mean-square error on the train and test series for each value of $M$. What is a good choice of $M$? Also find $J$ for the two simple predictors described above.

Hint. Be sure to use the toeplitz function contained in time_series_data.jl. It’ll make your life a lot easier. Documentation for the function is also contained in time_series_data.jl.

3. Trading off tracking error and input size in control. A system that we want to control has input (time series) $u$ and output (time series) $y$, related by convolution: $y = h \ast u$, where

$$h = (0.3, 0.5, 0.6, 0.4, 0.3, 0.2, 0.1).$$

(See §4.6.4 in the textbook.) We are given $y_{\text{des}}$, the (time series of) desired or target output values, and we will choose the input $u$ to minimize $\|y - y_{\text{des}}\|^2 + \lambda \|u\|^2$, where $\lambda > 0$ is a parameter we use to trade off tracking error (i.e., $\|y - y_{\text{des}}\|^2$) and input size (i.e., $\|u\|^2$). We will take the desired output to be the 100-vector

$$y_{\text{des}}^t = \begin{cases} 10 & 10 \leq t < 40 \\ -5 & 40 \leq t < 80 \\ 0 & \text{otherwise}. \end{cases}$$

2
Plot a trade off curve of the tracking error ($\text{rms}(y - y^{\text{des}})$) versus the regularization error ($\text{rms}(u)$).

Pick 3 values of $\lambda$ that correspond to too little regularization, too much regularization, and a reasonable amount of regularization. (Reasonable might correspond to RMS tracking error around 0.3) Plot the input $u$ found for each choice of $\lambda$ on the same figure. Also plot the output $y$ found for each $\lambda$ on the same figure, along with $y^{\text{des}}$.

Hint. Make sure to use the `toeplitz` function described in the hint for problem 2.

4. **Least-squares with smoothness regularization.** Consider the weighted sum least-squares objective

$$\|Ax - b\|^2 + \lambda\|Dx\|^2,$$

where the $n$-vector $x$ is the variable, $A$ is an $m \times n$ matrix, $D$ is the $(n - 1) \times n$ difference matrix, with $i$th row $(e_{i+1} - e_i)^T$, and $\lambda > 0$. Although it does not matter in this problem, this objective is what we would minimize if we want an $x$ that satisfies $Ax \approx b$, and has entries that are smoothly varying. We can express this objective as a standard least-squares objective with a stacked matrix of size $(m + n - 1) \times n$.

What are the conditions under which the stacked matrix has independent columns? Express your answer in the simplest way possible, in terms of the rows or columns of $A$, and the dimensions $m$ and $n$. (You may not need to mention all of these.) We will deduct points from solutions that are correct, but more complicated than they need to be. You must justify your answer.

5. **Conclusions from 5-fold cross validation.** You have developed a regression model for predicting a scalar outcome $y$ from a feature vector $x$ of dimension 20, using a collection of $N = 600$ data points. The mean of the outcome variable $y$ across the given data is 1.85, and its standard deviation is 0.32. After running 5-fold cross validation we get the following RMS test errors (based on forming a model based on the data excluding fold $i$, and testing it on fold $i$).

<table>
<thead>
<tr>
<th>Fold excluded</th>
<th>RMS test error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.13</td>
</tr>
<tr>
<td>2</td>
<td>0.11</td>
</tr>
<tr>
<td>3</td>
<td>0.09</td>
</tr>
<tr>
<td>4</td>
<td>0.11</td>
</tr>
<tr>
<td>5</td>
<td>0.14</td>
</tr>
</tbody>
</table>

(a) How would you expect your model to do on new, unseen (but similar) data? Respond briefly. Feel free to say that you are not comfortable making any prediction at all, and why, if this is the appropriate response. You should also feel free to give a guarantee, if you think it is warranted based on the results above. In any case, briefly justify your response.
(b) A co-worker observes that the regression model parameters found in the 5 different folds are quite close, but not the same. He says that for the production system, you should use the regression model parameters found when you excluded fold 3 from the original data, since it achieved the best RMS test error. Comment briefly.

(c) Another co-worker says that you might get a much better model (in terms of predicting new, unseen data) by adding a new 21st feature, which is \( x_{21} = \max\{x_1, \ldots, x_{20}\} \). Give an appropriate, and brief, response to her.

6. **Equalization in communication.** A communication system (such as a wireless, wired, or fiber optic link) is used to transfer a message, given as a Boolean \( N \)-vector \( s \), to a receiver, over a channel. Boolean means that the entries of \( s \) are all either 0 or 1. (Another common choice is \(-1\) or \(+1\).) The entries of \( s \) are called symbols; the set of possible values, in this case, 0 and 1, is called the symbol constellation.

Convolution is a very common model of the effect of the channel on the transmitted message \( s \). This means that the receiver gets the received signal \( y = c \ast s \), where \( c \) is an \( n \)-vector called the channel impulse response, and \( n \) is called the channel length. In typical channels \( n \) is small, say, no more than 10, and \( c_1 \approx 1 \) is the largest entry. When the coefficients \( c_2, \ldots, c_n \) are nonzero, \( y_i \) depends not just on \( s_i \), but also \( s_{i-1}, \ldots, s_{i-n+1} \). This effect is called inter-symbol interference (ISI).

A common method to estimate or guess the transmitted signal is \( \hat{s} = \text{round}(y_{1:N}) \), where the \text{round} function rounds each entry to 0 or 1 (with threshold 1/2). This method works well when the channel impulse response satisfies \( c \approx e_1 \), since then we have \( y \approx (s, 0_{n-1}) \), and so \( \hat{s}_i = s_i \), after rounding. When \( \hat{s}_i \neq s_i \), we say that a bit error has occurred in the transmission of bit \( i \). The bit error rate (BER) is the number of bit errors, divided by \( N \), the message length.

**Equalization** is a widely used method to combat the effects of ISI. An \( n \)-vector \( h \) (called the equalizer impulse response) is chosen or designed, so that \( h \ast c \approx e_{1,2n-1} \). (The vector \( h \ast c \) is sometimes called the equalized channel impulse response.) The receiver then forms the vector \( \tilde{y} = h \ast y \). This means that

\[
\tilde{y} = h \ast y = h \ast (c \ast s) = (h \ast c) \ast s \approx e_{1,2n-1} \ast s = (s_{1:N}, 0_{2n-2}).
\]

It follows that the signal can be decoded at the receiver using \( \hat{s}^{eq} = \text{round}(\tilde{y}_{1:N}) \). The interpretation is that the equalizer ‘undoes’ the affect of the channel.

Run the file \texttt{channel\_equalization\_data.jl}, which will define a message \( s \), a channel \( c \), and an equalizer \( h \). (Your are welcome to look inside the file to see how we designed the equalizer.)

Plot \( c \), \( h \), and \( h \ast c \). Make a brief comment about the channel and equalized channel impulse responses.

Plot \( s \), \( y \), and \( \tilde{y} \) over the index range \( i = 1, \ldots, 100 \). Is it clear from this plot that \( \hat{s} = \text{round}(y_{1:N}) \) will be worse estimate of \( s \) than \( \hat{s}^{eq} = \text{round}(\tilde{y}_{1:N}) \)?
Report the BER for \( \hat{s} \) (estimating the message without equalization), and for \( \hat{s}^{eq} \) (estimating the message with equalization).

*Hint:* To round a real vector \( x \) to \( \{0,1\} \) in Julia you can use \( (x \ .>0.5) \), which yields a Boolean vector. You can convert it to an integer vector (say, for plotting) using \( \text{int}(x \ .>0.5) \).

7. **Approximate right inverse.** Suppose the tall \( m \times n \) matrix \( A \) has independent columns. Unless it is square, it does not have a right inverse, i.e., there is no \( n \times m \) matrix \( X \) for which \( AX = I \). So instead we seek the matrix \( X \) for which the residual matrix \( R = AX - I \) is as small as possible, where we measure the size of \( R \) by the sum of the squares of its entries, \( \sum_{i,j} R_{ij}^2 \). We call this matrix the least-squares approximate right inverse of \( A \).

Show that the least-squares right inverse of \( A \) is given by \( X = A^t \).

*Hint.* Let \( x_i \) denote the \( i \)th column of \( X \). Show that the objective can be expressed as

\[
\sum_{i,j} R_{ij}^2 = \|Ax_1 - e_1\|^2 + \cdots + \|Ax_m - e_m\|^2.
\]

8. **State feedback control.** Consider a time-invariant linear dynamical system with \( n \)-vector state \( x_t \) and \( m \)-vector input \( u_t \), with dynamics

\[
x_{t+1} = Ax_t + Bu_t, \quad t = 1, 2, \ldots.
\]

The entries of the state often represent deviations of \( n \) quantities from their desired values, so \( x_t \approx 0 \) is a goal in operation of the system. The entries of the input \( u_t \) are deviations from the standard or nominal values. For example, in an aircraft model, the states might be the deviation from the desired altitude, climb rate, speed, and angle of attack; the input \( u_t \) represents changes in the control surface angles or engine thrust from their normal values.

In state feedback control, the states are measured and the input is a linear function of the state, \( u_t = Kx_t \). The \( m \times n \) matrix \( K \) is called the state feedback gain matrix. The state feedback gain matrix is very carefully designed, using several methods. State feedback control is very widely used in many application areas (including, for example, control of airplanes).

(a) **Open and closed-loop dynamical system.** With \( u_t = 0 \), the system satisfies \( x_{t+1} = Ax_t \), which is called the open-loop dynamics. When \( u_t = Kx_t \), the system dynamics can be expressed as \( x_{t+1} = \tilde{A}x_t \), where the matrix \( \tilde{A} \) is the closed-loop dynamics matrix. Find an expression for \( \tilde{A} \) in terms of \( A, B, \) and \( K \).
(b) Aircraft control. The longitudinal dynamics of a 747 flying at 40000 ft at Mach 0.81 is given by

\[ A = \begin{bmatrix} 0.99 & 0.03 & -0.02 & -0.32 \\ 0.01 & 0.47 & 4.7 & 0.00 \\ 0.02 & -0.06 & 0.40 & -0.00 \\ 0.01 & -0.04 & 0.72 & 0.99 \end{bmatrix}, \quad B = \begin{bmatrix} 0.01 & 0.99 \\ -3.44 & 1.66 \\ -0.83 & 0.44 \\ -0.47 & 0.25 \end{bmatrix}, \]

where the sampling time is one second. (The state and control variables are described in more detail in the lecture on control.) We will use the state feedback matrix

\[ K = \begin{bmatrix} -0.038 & 0.021 & 0.319 & -0.270 \\ -0.061 & -0.004 & -0.120 & 0.007 \end{bmatrix}. \]

(The matrices \( A \), \( B \), and \( K \) can be found in 747_cruise_dyn_data.jl, so you don’t have to type them in.) Plot the open-loop and closed-loop state trajectories from several nonzero initial states, such as \( x_1 = (1, 0, 0, 0) \), or ones that are randomly generated, from \( t = 1 \) to \( t = 100 \) (say). Would you rather be a passenger in the plane with the state feedback control turned off (i.e., open-loop) or on (i.e., closed-loop)?