Mathematical Tools for Neuroscience (NEU 314)
Princeton University, Spring 2016
Jonathan Pillow

Homework 9: Bootstrap, Linear Shift-Invariant Systems, & Fourier

Due: Tuesday, May 10 (Dean's Day), 11:59pm

Cross-validation and Bootstrap

Ridge Regression

Overfitting occurs when a model has too many degrees of freedom for a given dataset. The parameters end up capturing noise fluctuations in the training data, which degrades the model's ability to generalize to predict test data. Overfitting arises in regression settings when the number of parameters p is too large relative to the number of samples n. (If $p > N$, then the least-squares solution does not even exist, because the $X^\top X$ matrix is not invertible: it's a $p \times p$ matrix with rank only n.) It can also arise when the regressors are highly correlated, meaning they do not "fill out" the full p-dimensional space (or in other words, $X^\top X$ has eigenvalues close to zero, corresponding to directions that have little variance along them).

Ridge regression is the easiest and most common method to combat overfitting in least-squares regression. It can be written:

$$\hat{w}_{ridge} = (X^\top X + \lambda I)^{-1} X^\top Y, \tag{1}$$

where X is the $n \times p$ matrix of regressors, Y is the $n \times 1$ vector of observations, I is the $p \times p$ identity matrix, and λ is known as the "ridge parameter".

Of course, the standard least-squares regression estimator is simply $\hat{w}_{LS} = (X^\top X)^{-1} X^\top Y$, so the ridge regression estimator differs only for adding a scaled copy of the identity matrix, λI, to the $X^\top X$ matrix before taking the inverse. This ensures that the inverse is well conditioned: i.e.,instead of dividing by the singular value s_i of $X^\top X$, we divide by $s_i + \lambda$. This has negligible effect when s_i is large, but when it is close to zero, it keeps $1/s_i$ from blowing up (which would amplify the noise in the estimate along the direction of that singular vector). Ridge regression is known as a *shrinkage* estimator, because the coefficients of $\hat{w}_{ridge}$ are systematically closer to zero than those of $\hat{w}_{LS}$. Taking the ridge parameter all the way to ∞ would of course shrink them all the way to zero.

Aside: ridge regression has a Bayesian interpretation in terms of an iid Gaussian prior on the weights. Under a Gaussian observation model, i.e., $Y = X\mathbf{w} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$, the least-squares solution given above is the maximum likelihood estimate. But if we add a zero-mean Gaussian prior, $w \sim \mathcal{N}(0, vI)$, where v is the prior variance, then (if we work through the math) the ridge regression estimate emerges as the maximum a posteriori (MAP) estimate, where the ridge parameter is given by $\lambda = \frac{\sigma^2}{v}$, the ratio of the observation noise variance to the prior variance.

This makes clear that larger ridge parameter (which favors the the identity term I over the $X^\top X$ term) corresponds to assuming a narrower prior on w.

1. **Cross-validation**

   Load the file `RidgeRegressionData.mat`, which contains `Xstim`, a $1000 times 25$ design matrix, each row of which is a 20-dimensional stimulus vector (e.g., the light and dark values of a set of vertical stripes presented to the neuron). Make a 80%-20% split of data into training and test sets. Compute the ridge regression estimate using the training data, for a range of range of ridge parameter values from $\lambda = 0.1$ to $\lambda = 5000$. (You might wish to make a log-spaced grid using `logspace`, but you can also use a linearly spaced grid, or specify the grid by hand if you like.) Make a plot of the training error and test error (per sample) as a function of λ. What do you think is the best value of λ?

   Make a plot showing the ridge regression estimate $\hat{w}_{ridge}$ at $\lambda = 1$, $\lambda = 5000$, and whatever λ value you determined above to be (close enough to) optimal. (What does the maximum likelihood estimate for w look like? Show this on a separate plot.)

2. **Bootstrap**

   Compute 95% bootstrap confidence intervals for $\hat{w}_{ridge}$ at the ridge parameter λ selected as optimal in the previous problem. Draw 2000 bootstrap resamplings of the data (you can use all the data here instead of just the training data) and compute the ridge regression estimate for each one. You can use the function `randsample` to get a random sample of the data indices for use in each bootstrap sample. (See the help on this function, and be sure to set the "replace" argument to `true` so that sampling is carried out with replacement; otherwise you'll just get the original sample back each time).

   For each component of the weight vector, sort the bootstrap estimates from smallest to greatest, and select the 50th and 1950th as your estimate of the 2.5% and 97.5% confidence intervals. Use the matlab built-in function `errorbar` to generate a plot showing the ridge regression estimate with error bars.

3. **Permutation test**

   Use permutation methods to compute a null distribution for each regression weight in ridge regression, once again using the ridge parameter selected above. Draw 2000 artificial samples of the data in which the Y's are randomly permuted with respect to the X's (i.e., breaking the relationship between regressors and observation) and compute the ridge regression estimate for each. You may wish to use the matlab function `randperm`. Compute the 95% confidence interval for the null distribution of each weight. Plot these intervals using `errorbar` on the same axis with the ridge regression estimate. How many of the ridge regression weights are "significant" at the 5% level (i.e., how many lie outside the 95% intervals of the null distribution?) Replot these weights with a different colored symbol.

   (Note: this is not actually an analysis I'd recommend using in your own scientific research, but

it's an extremely common approach and therefore worth knowing about (and a simple example of a permutation test). Note that if we were going to use this to determine which regression weights are significant, at the very least we'd need to correct for multiple comparisons, since with 25 weights we'd expect at least 1 to be "significant" at the 5% level just by chance. But if your goal is to determine which weights are "meaningful", then there other methods—e.g., Bayesian approaches or sparse regression methods like the lasso—that I'd recommend more highly.)

Linear Shift Invariant Systems and Convolution

4. **Linear shift-invariant systems**. *Written exercises*: Oppenheim & Schafer, problems 2.35 and 2.36 [see attached pages]. Note: $\delta[n]$ indicates a signal that contains a single impulse (a "1") at location n, and zero elsewhere.

   (Feel free to answer these in commented text in your submitted m-file if that's easier than submitting something separate to your TA).

5. **Convolution in matlab**. Create a random vector of length 3, `r` = rand(3,1), and suppose this is finite-length impulse response of a linear shift-invariant system. Because it is LSI, the response of this system to any input vector `in` can be computed as a convolution.

   (a) Compute responses for some test input vectors of length 8 using matlab's `conv` function: `xout` = conv(xin, r). Try using:
   (i) `xin = zeros(8,1); xin(4) = 1;`
   (ii) `xin = zeros(8,1); xin(3) = 1;`
   (iii) `xin = zeros(8,1); xin(3:4) = 1;`
   (iv) `xin = zeros(8,1); xin([1,8]) = 1;`

   (b) Generate the Toeplitz matrix that implements this LSI system, accepting length-8 vectors as input. What is the size, and organization of this matrix? Compare its output (multiplying by an input vector) to the output of `conv` to make sure it matches.

   (c) How does matlab's `conv` function handle boundaries?

   (d) Make a 48-sample signal that consists of 4 periods of a sine wave (so the period of the sine wave is 12). Using `conv`, compute the response of the filter defined above to this input vector. Is this a 4-cycle sinusoid? Why or why not? If not, what modification would be necessary to the `conv` function to ensure that it would behave according to the "sine-in, sine-out" behavior expected of LSI systems?

Fourier transform

6. Write a function called `makeDFTbasis.m` that takes a single integer n as input and generates the $n \times n$ matrix that performs the discrete Fourier transform on vectors of length n. The k'th row of this matrix is given by $\exp(-2\pi ikt)$, for $k \in [0, ..., n-1]$ (where we assume indexing starts at 0 instead of 1), and t is a row vector `t=0:n-1;`. You can check that your function works correctly by comparing it to the output of `fft(eye(n))`, which is the Fourier transform of the $n \times n$ identity matrix.

7. Generate the matrix `M = makeDFTbasis(100)`. Make plots showing the real and imaginary components of the first 4 rows of this matrix. (In matlab you can use `real` and `imag` to extract the real or imaginary component of a complex number). Is M an orthogonal matrix? If not, how does it fall short, and what could you do to make it orthogonal?

8. Use the convolution theorem to convolve the 48-sample sinusoidal signal defined above with the 3-sample filter you generated. You'll want to pad your filter with zeros by creating a new vector with 45 zeros after the three elements of your filter, but you can directly take the Fourier transform of the padded filter by passing in a second argument to `fft` specifying the desired length of the signal to be transformed, e.g., `f_hat = fft(f,48);`. Perform pointwise multiplication of your Fourier-transformed filter with the Fourier-transformed sine wave, then take the inverse Fourier transform (and you may need to take the "real" part only to get rid of an infinitesimal imaginary component containing roundoff errors). Plot the output. Is it a pure sinusoid?

   (Note: the discrete Fourier transform effectively assumes your signal is periodic with period equal to the length of the sample!)

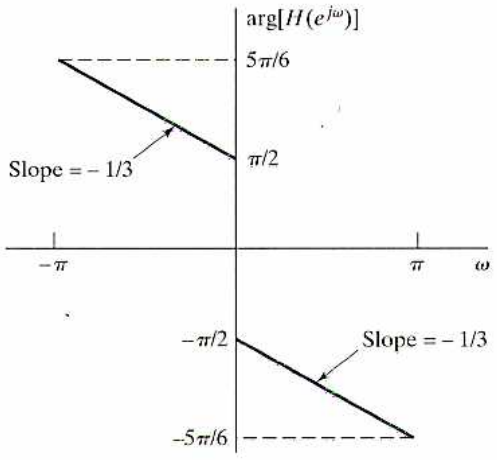

Figure P2.33-1

2.34. The input–output pair shown in Figure P2.34-1 is given for a stable LTI system.

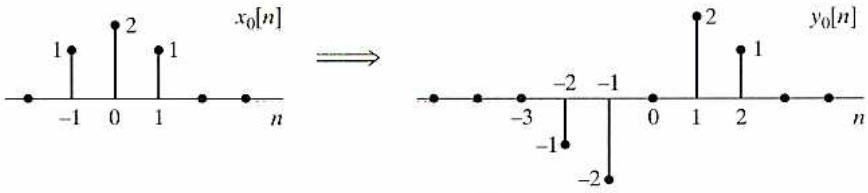

Figure P2.34-1

(a) Determine the response to the input $x_1[n]$ in Figure P2.34-2.

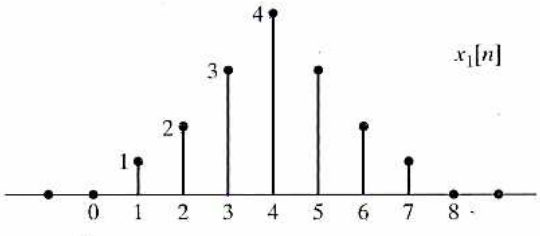

Figure P2.34-2

(b) Determine the impulse response of the system.

Advanced Problems

2.35. The system T in Figure P2.35-1 is known to be *time invariant*. When the inputs to the system are $x_1[n]$, $x_2[n]$, and $x_3[n]$, the responses of the system are $y_1[n]$, $y_2[n]$, and $y_3[n]$, as shown.

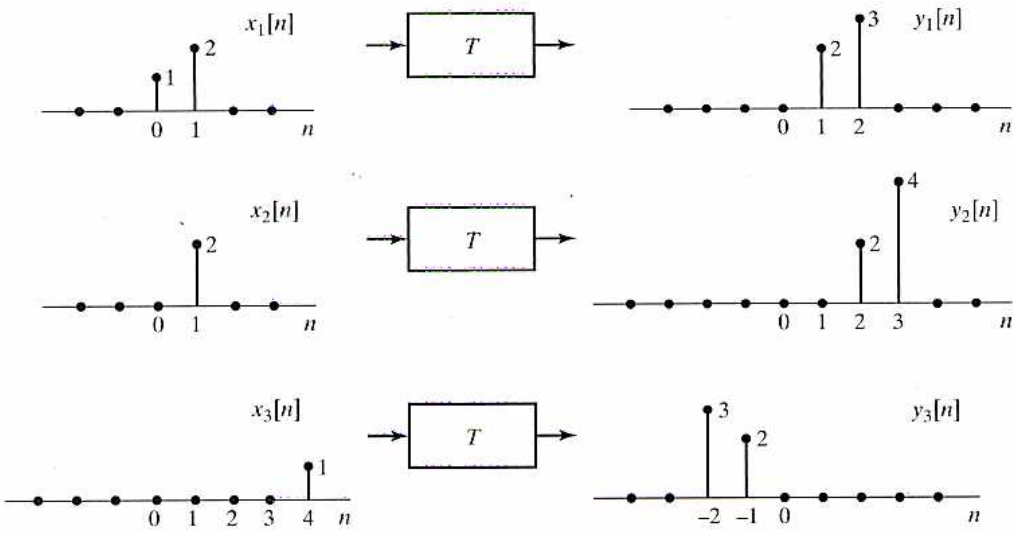

Figure P2.35-1

(a) Determine whether the system T could be linear.
(b) If the input $x[n]$ to the system T is $\delta[n]$, what is the system response $y[n]$?
(c) What are all possible inputs $x[n]$ for which the response of the system T can be determined from the given information alone?

2.36. The system L in Figure P2.36-1 is known to be *linear*. Shown are three output signals $y_1[n]$, $y_2[n]$, and $y_3[n]$ in response to the input signals $x_1[n]$, $x_2[n]$, and $x_3[n]$, respectively.

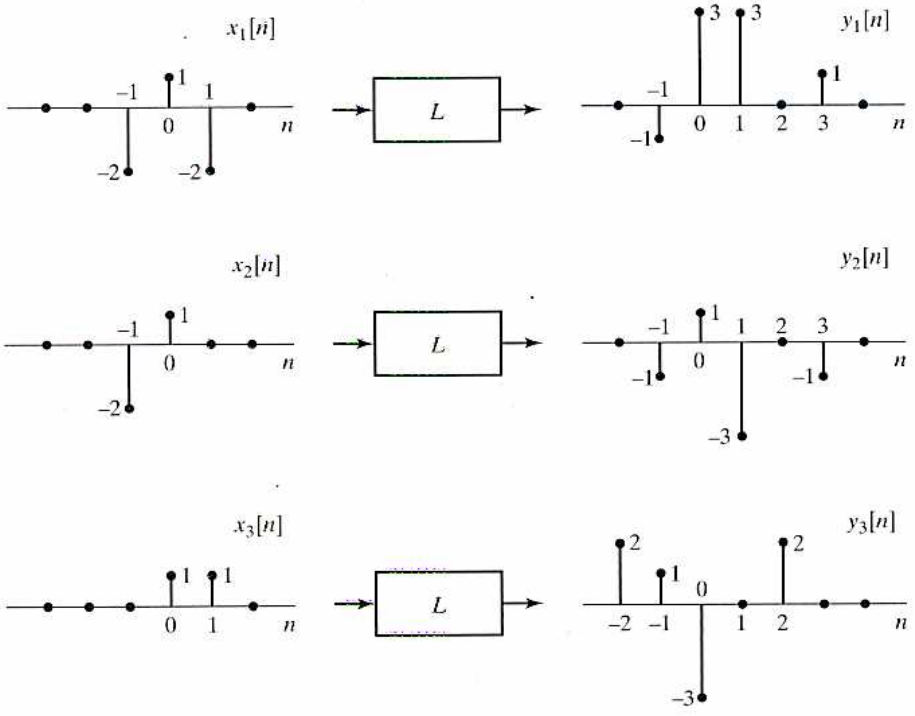

Figure P2.36-1

(a) Determine whether the system L could be time invariant.

(b) If the input $x[n]$ to the system L is $\delta[n]$, what is the system response $y[n]$?

2.37. Consider a discrete-time linear time-invariant system with impulse response $h[n]$. If the input $x[n]$ is a periodic sequence with period N (i.e., if $x[n] = x[n+N]$), show that the output $y[n]$ is also a periodic sequence with period N.

2.38. In Section 2.5, we stated that the solution to the homogeneous difference equation

$$\sum_{k=0}^{N} a_k y_h[n-k] = 0 \tag{P2.38-1}$$

is of the form

$$y_h[n] = \sum_{m=1}^{N} A_m z_m^n, \tag{P2.38-2}$$

with the A_m's arbitrary and the z_m's the N roots of the polynomial

$$\sum_{k=0}^{N} a_k z^{-k} = 0; \tag{P2.38-3}$$

i.e.,

$$\sum_{k=0}^{N} a_k z^{-k} = \prod_{m=1}^{N} (1 - z_m z^{-1}). \tag{P2.38-4}$$

(a) Determine the general form of the homogeneous solution to the difference equation

$$y[n] - \tfrac{3}{4}y[n-1] + \tfrac{1}{8}y[n-2] = 2x[n-1]. \tag{P2.38-5}$$

(b) Determine the coefficients A_m in the homogeneous solution if $y[-1] = 1$ and $y[0] = 0$.

(c) Now consider the difference equation

$$y[n] - y[n-1] + \tfrac{1}{4}y[n-2] = 2y[n-1]. \tag{P2.38-6}$$

If the homogeneous solution contains only terms of the form of Eq. (P2.38-2), show that the initial conditions $y[-1] = 1$ and $y[0] = 0$ cannot be satisfied.

(d) If Eq. (P2.38-3) has two roots that are identical, then, in place of Eq. (P2.38-2), $y_h[n]$ will take the form

$$y_h[n] = \sum_{m=1}^{N-1} A_m z_m^n + nB_1 z_1^n, \tag{P2.38-7}$$

where we have assumed that the double root is z_1. Using Eq. (P2.38-7), determine the general form of $y_h[n]$ for Eq. (P2.38-6). Verify explicitly that your answer satisfies Eq. (P2.38-6) with $x[n] = 0$.

(e) Determine the coefficients A_1 and B_1 in the homogeneous solution obtained in Part (d) if $y[-1] = 1$ and $y[0] = 0$.

2.39. Consider a system with input $x[n]$ and output $y[n]$. The input–output relation for the system is defined by the following two properties:

1. $y[n] - ay[n-1] = x[n]$,
2. $y[0] = 1$.

(a) Determine whether the system is time invariant.

(b) Determine whether the system is linear.