Assignment #1 (Due 1/29/16)

Regarding the curse of dimensionality/necessary sparsity of data sets in $\mathbb{R}^p$ for moderate to large $p$:

1. For each of $p = 10, 20, 50, 100, 500, 1000$ make $n = 10,000$ draws of distances between pairs of independent points uniform in the cube in $p$-space, $[0,1]^p$. Use these to make 95% confidence limits for the ratio

$$\frac{\text{mean distance between two random points in the cube}}{\text{maximum distance between two points in the cube}}$$

2. For each of $p = 10, 20, 50$ make $n = 10,000$ random draws of $N = 100$ independent points uniform in the cube $[0,1]^p$. For each sample of 100 points, find the distance from the first point drawn to the 5th closest point of the other 99. Use these values to make 95% confidence limits for the ratio

$$\frac{\text{mean diameter of a 5-nearest neighbor neighborhood if } N = 100}{\text{maximum distance between two points in the cube}}$$

3. What fraction of a distribution uniform on the unit cube $[0,1]^p$ in $p$-space lies in the "middle part" of the cube, $[\varepsilon, 1-\varepsilon]^p$, for a small positive number $\varepsilon$? Evaluate this for $\varepsilon = .05$ and $p = 100$.

Regarding the typical standardization of input variables in predictive analytics:

4. Consider the small ($N = 11$) fake $p = 2$ set of predictors that can be entered into R using:

```r
x1<-c(11,12,13,14,13,15,17,16,17,18,19)
x2<-c(18,12,14,16,6,10,14,4,6,8,2)
```

One can standardize variables in R using the `scale()` function.

a) Plot raw and standardized versions of 11 predictor pairs $(x_1, x_2)$ on the same set of axes (using different plotting symbols for the two versions and a 1:1 aspect ratio for the plotting).

b) Find sample means, sample standard deviations, and the sample correlations for both versions of the predictor pairs.
Regarding optimal predictors and decomposition of $Err$:

First, study the solutions to problems 1, 2, and 3 of the 2014 Stat 502X Mid-Term Exam, then consider the following.

5. Suppose that (unknown to a statistician) a mechanism generates iid data pairs $(x, y)$ according to the following model:

$$x \sim U(-\pi, \pi)$$

$$y \mid x \sim N(\sin(x), 0.25(|x|+1)^2)$$

(The conditional variance is $0.25(|x|+1)^2$.)

**a)** What is an absolute minimum value of $Err$ possible regardless what training set size, $N$, is available and what fitting method is employed?

**b)** What linear function of $x$ (which $g(x) = a + bx$) has the smallest "average squared bias" as a predictor for $y$? What cubic function of $x$ (which $g(x) = a + bx + cx^2 + dx^3$) has the smallest average squared bias as a predictor for $y$? Is the set of cubic functions big enough to eliminate model bias in this problem?

6. Consider a 0-1 loss $K = 2$ classification problem with $p = 1$, $\pi_0 = \pi_1 = \frac{1}{2}$, and pdfs

$$g(x \mid 0) = I[-.5 < x < .5] \quad \text{and} \quad g(x \mid 1) = 12x^2I[-.5 < x < .5]$$

**a)** What is the optimal classification rule in this problem? (In the notation of the slides, this is $f(x)$). What is the "minimum expected loss" part of $Err$ in this problem?

**b)** Identify the best rule of the form $g_c(x) = I[x > c]$. (In the notation of the slides, this is $g^*(x)$ for $S = \{g_c\}$. This could be thought of as the 1-d version of a "best linear classification rule" here ... where linear classification is not so smart.) What is the "modeling penalty" part of $Err$ in this situation?

**c)** Suggest a way that you might try to choose a classification rule $g_c$ based on a very large training sample of size $N$. Notice that a large training set would allow you to estimate cumulative conditional probabilities $G(c \mid y) = P[x \leq c \mid y]$ by relative frequencies

$$\frac{\# \text{ training cases with } x_i \leq c \text{ and } y_i = y}{\# \text{ training cases with } y_i = y}$$
d) If one were to do "feature selection" here, adding some function of \( x \), say \( t(x) \), to make a vectors of predictors \((x, t(x))\) for classification purposes, hoping to eventually employ a good "linear classifier"

\[
\hat{f}(x, t(x)) = I[a + bx + ct(x) > 0]
\]

for appropriate constants \( a, b, \) and \( c \), what (knowing the answer to a) would be a good choice of \( t(x) \)? (Of course, one doesn't know the answer to a) when doing feature selection!)

Regarding cross-validation as a method of choosing a predictor:

7. Vardeman will send out an \( N = 100 \) data set generated by the model of problem 5. Use ten-fold cross validation (use the 1\textsuperscript{st} ten points as the first test set, the 2\textsuperscript{nd} 10 points as the second, etc.) based on the data set to choose among the following methods of prediction for this scenario:
   - polynomial regressions of orders 0,1,2,3,4, and 5
   - regressions using sets of predictors \{1,sinx,cosx\} and \{1,sinx,cosx,sin2x,cos2x\}
   - a regression with the set of predictors \{1, x, x^2, x^3, x^4, sinx, cosx, sin2x, cos2x\}

(Use ordinary least squares fitting.) Which predictor looks best on an empirical basis? Knowing how the data were generated (an unrealistic luxury) which methods here are without model bias?

Regarding some linear algebra and principal components analysis:

8. Consider the small \((11 \times 2\) ) fake \( \mathbf{X} \) matrices corresponding to the raw and standardized versions of the data of problem 4. Interpret the first principal component direction vectors for the two versions and say why (in geometric terms) they are much different.

9. Consider the small \((7 \times 3\) ) fake \( \mathbf{X} \) matrix below.

\[
\mathbf{X} = \begin{pmatrix}
10 & 10 & .1 \\
11 & 11 & -.1 \\
9 & 9 & 0 \\
11 & 9 & -2.1 \\
9 & 11 & 2.1 \\
12 & 8 & -4.0 \\
8 & 12 & 4.0
\end{pmatrix}
\]

(Note, by the way, that \( \mathbf{x}_3 \approx \mathbf{x}_2 - \mathbf{x}_1 \).)

a) Find the QR and singular value decompositions of \( \mathbf{X} \). Use the latter and give best rank 1 and rank 2 approximations to \( \mathbf{X} \).
b) Subtract column means from the columns of $X$ to make a centered data matrix. Find the singular value decomposition of this matrix. Is it approximately the same as that in part a)? Give the 3 vectors of principal component scores. What are the principal components for case 1?

Henceforth consider only the centered data matrix of b).

c) What are the singular values? How do you interpret their relative sizes in this context? What are the first two principal component directions? What are the loadings of the first two principal component directions on $x_3$? What is the third principal component direction?

c') Find the matrices $Xv_jv'_j$ for $j = 1, 2, 3$ and the best rank 1 and rank 2 approximations to $X$.

How are the latter related to the former?

d) Compute the ($N$ divisor) $3 \times 3$ sample covariance matrix for the 7 cases. Then find its singular value decomposition and its eigenvalue decomposition. Are the eigenvectors of the sample covariance matrix related to the principal component directions of the (centered) data matrix? If so, how? Are the eigenvalues/singular values of the sample covariance matrix related to the singular values of the (centered) data matrix. If so, how?

e) The functions

$$K_1(x, z) = \exp\left(-\nu \|x - z\|^2\right) \quad \text{and}$$

$$K_2(x, z) = (1 + \langle x, z \rangle)^d$$

are legitimate kernel functions for choice of $\nu > 0$ and positive integer $d$. Find the first two kernel principal component vectors for $X$ for each of cases

1. $K_1$ with two different values of $\nu$ (of your choosing), and
2. $K_2$ for $d = 1, 2$.

If there is anything to interpret (and there may not be) give interpretations of the pairs of vectors for each of the 4 cases. (Be sure to use the vectors for "centered versions" of latent feature vectors.)

Assignment #2. (Due 3/2/16)

10. Return to the context of problem 7 and the last/largest set of predictors. Center the $y$ vector to produce (say) $Y^*$, remove the column of 1's from the $X$ matrix (giving a $100 \times 9$ matrix) and standardize the columns of the resulting matrix, to produce (say) $X^*$.

a) If one somehow produces a coefficient vector $\beta^*$ for the centered and standardized version of the problem, so that

$$\hat{y} = \beta_1^* x_1^* + \beta_2^* x_2^* + \cdots + \beta_9^* x_9^*$$
what is the corresponding predictor for $y$ in terms of $\{1, x, x^2, x^3, x^4, x^5, \sin x, \cos x, \sin 2x, \cos 2x\}$?

b) Do the transformations and fit the equation in a) by OLS. How do the fitted coefficients and error sum of squares obtained here compare to what you get simply doing OLS using the raw data (and a model including a constant term)?

c) Augment $Y^*$ to $Y^{**}$ by adding 9 values 0 at the end of the vector (to produce a 109×1 vector) and for value $\lambda = 4$ augment $X^*$ to $X^{**}$ (a 109×9 matrix) by adding 9 rows at the bottom of the matrix in the form of $\sqrt{\lambda} \cdot \mathbf{I}$. What quantity does OLS based on these augmented data seek to optimize? What is the relationship of this to a ridge regression objective?

d) Use trial and error and matrix calculations based on the explicit form of $\hat{\beta}_{\lambda}^{\text{ridge}}$ given in the slides for Module 5 to identify a value of $\lambda$ for which the error sum of squares for ridge regression is about 1.5 time that of OLS in this problem. Then make a series of at least 5 values from 0 to $\tilde{\lambda}$ to use as candidates for $\lambda$. Choose one of these as an "optimal" ridge parameter $\lambda_{\text{opt}}$ here based on 10-fold cross-validation (as was done in problem 7). (In light of the class discussion of the meaning of "real" cross-validation, you'll need to redo the standardization before making predictions for each fold). Compute the corresponding predictions $\hat{y}_{\text{ridge}}^i$ and plot both them and the OLS predictions as functions of $x$ (connect successive $(x, \hat{y})$ points with line segments). How do the "optimal" ridge predictions based on the 9 predictors compare to the OLS predictions based on the same 9 predictors?

11. In light of the idea in part c) of problem 10, if you had software capable of doing lasso fitting of a linear predictor for a penalty coefficient $\lambda$, how can you use that routine to do elastic net fitting of a linear predictor for penalty coefficients $\lambda_1$ and $\lambda_2$ in

$$\sum_{i=1}^{N} (y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^{p} |\hat{\beta}_j| + \lambda_2 \sum_{j=1}^{p} \hat{\beta}_j^2$$?

12. Here is a small fake data set with $p = 4$ and $N = 8$.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$y$</td>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$x_4$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-5</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>-3</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>-11</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-5</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Notice that the $y$ is centered and the $x$'s are orthogonal (and can easily be made orthonormal by dividing by $\sqrt{8}$). Use the explicit formulas for fitted coefficients in the orthonormal features context to make plots (on a single set of axes for each fitting method, 5 plots in total) of

1. $\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3$, and $\hat{\beta}_4$ versus $M$ for best subset (of size $M$) regression,
2. $\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3$, and $\hat{\beta}_4$ versus $\lambda$ for ridge regression,
3. $\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3$, and $\hat{\beta}_4$ versus $\lambda$ for lasso,
4. $\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3$, and $\hat{\beta}_4$ versus $\lambda$ for $\alpha = .2$ in the elastic net penalty

$$\sum_{i=1}^{N} (y_i - \hat{y}_i)^2 + \lambda \left( (1-\alpha) \sum_{j=1}^{p} |\hat{\beta}_j| + \alpha \sum_{j=1}^{p} \hat{\beta}_j^2 \right)$$

5. $\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3$, and $\hat{\beta}_4$ versus $\lambda$ for the non-negative garrote.

Make 5 corresponding plots of the error sum of squares versus the corresponding parameter.

13. For the data set of problem 7 make up a matrix of inputs based on $x$ consisting of the values of Haar basis functions up through order $m = 3$. (You will need to take the functions defined on $[0,1]$ and rescale their arguments to $[-\pi, \pi]$. For a function $g : [0,1] \rightarrow \mathbb{R}$ this is the function $g^* : [-\pi, \pi] \rightarrow \mathbb{R}$ defined by $g^*(x) = g\left(\frac{x}{2\pi} + .5\right)\). This will produce a 100\times16 matrix $X_h$.

a) Find $\hat{\beta}^{\text{OLS}}$ and plot the corresponding $\hat{y}$ as a function of $x$ with the data also plotted in scatterplot form.

b) Center $y$ and standardize the columns of $X_h$. Find the lasso coefficient vectors $\hat{\beta}$ with exactly $M = 2, 4, \text{ and } 8$ non-zero entries with the largest possible $\sum_{j=2}^{16} |\hat{\beta}_j^{\text{lasso}}| \text{ (for the counts of non-zero entries)}. Plot the corresponding $\hat{y}$'s as functions of $x$ on the same set of axes, with the data also plotted in scatterplot form.

14. Consider the basis functions for natural cubic splines with knots $\xi_j$ given on panel 7 of Module 9:

$$h_1(x) = 1, h_2(x) = x, \text{ and}$$

$$h_{j+2}(x) = \begin{cases} (x - \xi_j)^3 & \text{for } j = 1,2,\ldots,K-2 \\ \left(\frac{\xi_{K-1} - \xi_j}{\xi_K - \xi_{K-1}}\right) (x - \xi_{K-1})^3 + \left(\frac{\xi_j - \xi_{K-1}}{\xi_K - \xi_{K-1}}\right) (x - \xi_K)^3 & \text{for } j = 1,2,\ldots,K-2 \end{cases}$$

Using knots $\xi_1 = -8\pi, \xi_2 = -6\pi, \xi_3 = -4\pi, \xi_4 = -2\pi, \xi_5 = 0, \xi_6 = 2\pi, \xi_7 = 4\pi, \xi_8 = 6\pi, \xi_9 = .8\pi$ fit a natural cubic regression spline to the data of problem 7 using OLS. Plot the fitted function on the same axes as the data points.
15. Now for $p = 1$ suppose that $N$ observations $(x_i, y_i)$ have distinct $x_i$, and for simplicity of notation, suppose that $x_1 < x_2 < \cdots < x_N$. Consider the smoothing spline problem using the "basis functions" of problem 14, with $\xi_j = x_j$. Obviously, $h_1$ and $h_2$ have second derivative functions that are everywhere 0 and the products of these second derivatives with themselves or 2nd derivatives of other basis functions must have 0 integral from $a$ to $b$.

Then for $j = 1, 2, 3, \ldots, N - 2$

$$h_{j+2}^{n}(x) = 6(x - x_j) I \left[ x_j \leq x \leq x_{N-1} \right] + 6 \left( x - x_j \right) - \frac{x - x_j}{x - x_{N-1}} \left( x - x_{N-1} \right) I \left[ x_{N-1} \leq x \leq x_N \right]$$

$$+ 6 \left( x - x_j \right) - \frac{x - x_j}{x - x_{N-1}} \left( x - x_{N-1} \right) + \frac{x - x_j}{x - x_{N-1}} \left( x - x_N \right) I \left[ x_N \leq x \leq b \right]$$

$$= 6 \left( x - x_j \right) I \left[ x_j \leq x \leq x_{N-1} \right] + 6 \left( x - x_j \right) - \frac{x - x_j}{x - x_{N-1}} \left( x - x_{N-1} \right) - \frac{x - x_j}{x - x_{N-1}} \left( x - x_N \right) I \left[ x_{N-1} \leq x \leq x_N \right]$$

$$= 6 \left( x - x_j \right) I \left[ x_j \leq x \leq x_{N-1} \right] + 6 \left( x - x_N \right) \frac{x - x_j}{x - x_{N-1}} I \left[ x_{N-1} \leq x \leq x_N \right]$$

Thus for $j = 1, 2, 3, \ldots, N - 2$

$$\int_a^b \left( h_{j+2}^{n}(x) \right)^2 dx = 12 \left( x_{N-1} - x_j \right)^3 + (x_N - x_{N-1})^3 \left( x - x_N \right)^2$$

$$= 12 \left( x_{N-1} - x_j \right)^3 + (x_N - x_{N-1}) \left( x_N - x_j \right)^2$$

and for positive integers $1 \leq j < k \leq N - 2$

$$\int_a^b h_{j+2}^{n}(x) h_{k+2}^{n}(x) dx = 36 \left( \int_{x_j}^{x_{N-1}} (x - x_j) (x - x_k) dx + \int_{x_{N-1}}^{x_N} (x - x_N \right)^2 \left( x - x_N \right) \left( x_k - x_N \right) dx)$$

$$= 12 \left( x_{N-1}^2 - x_j^2 \right) - 18 \left( x_j + x_k \right) + 36 x_j x_k \left( x_{N-1} - x_k \right)$$

$$+ 12 \left( x_N - x_{N-1} \right) \left( x_N - x_j \right) \left( x_N - x_k \right)$$

(There is no guarantee that I haven't messed up my calculus/algebra here, so check all this!)

On the course web page, there are "handouts" for a small smoothing example of Prof. Morris. It involves an $N = 11$ point data set. Here is R code for entering it.

```r
> x <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1)
```
> y <- c(1.0030100, 0.8069872, 0.6690364, 0.6281389, 0.5542417, 0.5105527, 0.5306341, 0.5023222, 0.6103748, 0.7008915, 0.9422990)

Do the smoothing spline computations "from scratch" using the above representations of the entries of the matrix $\Omega$. That is,

a) Compute the $11 \times 11$ matrix $\Omega$.

b) For $\lambda = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$, and 0 compute the smoother matrices $S_\lambda$ and the effective degrees of freedom. Compare your degrees of freedom to what Prof. Morris found and compare your $S_{0.01}$ to his.

c) Find the penalty matrix $K$ and its eigen decomposition. Plot as functions of $x$ (or just $i$ assuming that you have ordered the values of $x$) the entries of the eigenvectors of this matrix (connect successive points with line segments so that you can see how these change in character as the corresponding eigenvalue of $K$ increases—the corresponding eigenvalue of $S_\lambda$ decreases). Which $\mathbb{R}^{11}$ components of the observed $Y$ are most suppressed in the smoothing operation? Can you describe them in qualitative terms?

16. Again using the data set of problem 7,

a) Fit with approximately 5 and then 9 effective degrees of freedom
   i) a cubic smoothing spline (using `smooth.spline()`), and
   ii) a locally weighted linear regression smoother based on a tricube kernel (using `loess(...,span= ,degree=1)`).

Plot for approximately 5 effective degrees of freedom all of $y_i$ and the 2 sets of smoothed values against $x_i$. Connect the consecutive $(x_i, y_i)$ for each fit with line segments so that they plot as "functions." Then redo the plotting for 9 effective degrees of freedom.

b) Produce a single hidden layer neural net fit with an error sum of squares about like those for the 9 degrees of freedom fits using `nnet()` . You may need to vary the number of hidden nodes for a single-hidden-layer architecture and vary the weight for a penalty made from a sum of squares of coefficients in order to achieve this. For the function that you ultimately fit, extract the coefficients and plot the fitted mean function. How does it compare to the plots made in a)?

c) Each run of an `nnet()` begins from a different random start and can produce a different fitted function. Make 5 runs using the architecture and penalty parameter (the "decay" parameter) you settle on for part b) and save the 100 predicted values for the 10 runs into 10 vectors. Make a scatterplot matrix of pairs of these sets of predicted values. How big are the correlations between the different runs?

d) Use the `avNNet()` function from the `caret` package to average 20 neural nets with your parameters from part b).
e) Fit radial basis function networks based on the standard normal pdf $\phi$, 

$$f_{\hat{\lambda}}(x) = \beta_0 + \sum_{i=1}^{100} \beta_i K_{\hat{\lambda}}(x, x_i) \quad \text{for} \quad K_{\hat{\lambda}}(x, x_i) = \phi \left( \frac{x - x_i}{\hat{\lambda}} \right)$$

to these data for two different values of $\hat{\lambda}$ using Lasso (with cross-validated choice of the penalty weight). Define normalized versions of the radial basis functions as

$$N_{\hat{\lambda}}(x) = \frac{K_{\hat{\lambda}}(x, x_i)}{\sum_{m=1}^{100} K_{\hat{\lambda}}(x, x_m)}$$

and redo the problem using the normalized versions of the basis functions. Plot all 4 of these fits on the same set of axes.

17. Treat the parameter "degree" in the earth() routine as a complexity parameter (accepting the defaults for all other parameters of the routine). Pick a value for this parameter via 8-fold cross-validation on the Ames Housing data. Compare predictions for this "optimal" MARS fit to the data to the best 7-NN (chosen by cross-validation), best Lasso (chosen by cross-validation), and PLS fits to the data on the basis of honest cross validation. (This means that in the case of PLS you've got to re-standardize and re-center for each fold.)