1. Consider Section 1.4 of the typed outline (concerning the variance-bias trade-off in prediction). Suppose that in a very simple problem with \( p = 1 \), the distribution \( P \) for the random pair \((x, y)\) is specified by

\[
x \sim U(0,1) \text{ and } y \mid x \sim N(x^2, (1+x))
\]

\((1+x)\) is the conditional variance of the output). Further, consider two possible sets of functions \( S = \{g\} \) for use in creating predictors of \( y \), namely

1. \( S_1 = \{g \mid g(x) = a + bx \text{ for real numbers } a, b\} \), and

2. \( S_2 = \{g \mid g(x) = \sum_{j=10}^{10} a_j I \left[ \frac{j-1}{10} < x \leq \frac{j}{10} \right] \text{ for real numbers } a_j\} \)

Training data are \( N \) pairs \((x_i, y_i)\) iid \( P \). Suppose that the fitting of elements of these sets is done by

1. OLS (simple linear regression) in the case of \( S_1 \), and

2. according to

\[
\hat{a}_j = \begin{cases} 
\bar{y} & \text{if no } x_i \in \left( \frac{j-1}{10}, \frac{j}{10} \right] \\
\frac{1}{\# \{x_i \in \left( \frac{j-1}{10}, \frac{j}{10} \right] \}} \sum_{x_i \in \left( \frac{j-1}{10}, \frac{j}{10} \right]} y_i & \text{otherwise}
\end{cases}
\]

in the case of \( S_2 \)

to produce predictors \( \hat{f}_1 \) and \( \hat{f}_2 \).

a) Find (analytically) the functions \( g^* \) for the two cases. Use them to find the two expected squared model biases \( \text{E}^T \left( \text{E}[y \mid x] - g^*(x) \right)^2 \). How do these compare for the two cases?

b) For the second case, find an analytical form for \( \text{E}^T \hat{f}_2 \) and then for the average squared estimation bias \( \text{E}^T \left( \text{E}^T \hat{f}_2(x) - g^*_2(x) \right)^2 \). (Hints: What is the conditional distribution of the \( y_i \) given that no \( x_i \in \left( \frac{j-1}{10}, \frac{j}{10} \right] \)? What is the conditional mean of \( y \) given that \( x \in \left( \frac{j-1}{10}, \frac{j}{10} \right] \)?)

c) For the first case, simulate at least 1000 training data sets of size \( N = 50 \) and do OLS on each one to get corresponding \( f \)'s. Average those to get an approximation for \( \text{E}^T \hat{f}_1 \). (If you can do
this analytically, so much the better!) Use this approximation and analytical calculation to find the average squared estimation bias $E^*\left(E^T f_1^*(x) - g_1^*(x)\right)^2$ for this case.

d) How do your answers for b) and c) compare for a training set of size \( N = 50 \) ?

e) (This may not be trivial, I haven't completely thought it through.) Use whatever combination of analytical calculation, numerical analysis, and simulation you need to use (at every turn preferring analytics to numerics to simulation) to find the expected prediction variances $E^*\text{Var}^T\left(\hat{f}(x)\right)$ for the two cases for training set size \( N = 50 \).

f) In sum, which of the two predictors here has the best value of Err for \( N = 50 \)?

2. Two files provided at http://thirteen-01.stat.iastate.edu/snoweye/stat602x/ with respectively 50 and then 1000 pairs \((x_i, y_i)\) were generated according to \( P \) in Problem 1. Use 10-fold cross validation to see which of the two predictors in Problem 1 looks most likely to be effective. (The data sets are not sorted, so you may treat successively numbered groups of 1/10th of the training cases as your \( K = 10 \) randomly created pieces of the training set.)

3. (This is another calculation intended to provide intuition in the direction that shows data in \( \mathbb{R}^p \) are necessarily sparse.) Let \( F_p(t) \) and \( f_p(t) \) be respectively the \( \chi^2_p \) cdf and pdf. Consider the \( \text{MVN}_p(0, I) \) distribution and \( Z_1, Z_2, \ldots, Z_N \) iid with this distribution. With

\[
M = \min \left\{ \| Z_i \| : i = 1, 2, \ldots, N \right\}
\]

Write out a one-dimensional integral involving \( F_p(t) \) and \( f_p(t) \) giving \( EM \). Evaluate this mean for \( N = 100 \) and \( p = 1, 5, 10, \) and 20 either numerically using simulation.

**HW#2 (Due 2/9/11)**

4. Show the equivalence of the two forms of the optimization used to produce the fitted ridge regression parameter. (That is, show that there is a \( t(\lambda) \) such that \( \hat{\beta}^{\text{ridge}}_\lambda = \hat{\beta}^{\text{ridge}}_{t(\lambda)} \) and a \( \lambda(t) \) such that \( \hat{\beta}^{\text{ridge}}_t = \hat{\beta}^{\text{ridge}}_{\lambda(t)} \).)

5. Consider the linear space of functions on \([0,1]\) of the form

\[ f(t) = a + bt + ct^2 + dt^3 \]

Equip this space with the inner-product \( \langle f, g \rangle \equiv \int_0^1 f(t)g(t) dt \) and norm \( \| f \| = \langle f, f \rangle^{1/2} \) (to create a small Hilbert space). Use the Gram-Schmidt process to orthogonalize the set of functions \( \{1, t, t^2, t^3\} \) and produce an orthonormal basis for the space.
6. Consider the matrix
\[ X = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 1 & 2 & 1 \\ 2 & 2 & 1 \end{bmatrix} \]
a) Use \( \mathbb{R} \) and find the QR and singular value decompositions of \( X \). What are the two corresponding bases for \( C(X) \)?
b) Use the singular value decomposition of \( X \) to find the eigen (spectral) decomposition of \( X'X \) (what are the eigenvalues and eigenvectors?).
c) Find the best \( rank = 1 \) and \( rank = 2 \) approximations to \( X \).

7. Consider "data augmentation" methods of penalized least squares fitting.
a) Augment a centered \( X \) matrix with \( p \) new rows \( \sqrt{\lambda} \mathbf{I}_{pp} \) and \( Y \) by adding \( p \) new entries 0. Argue that OLS fitting with the augmented data set returns \( \hat{\beta}_{ridge} \) as a fitted coefficient vector.
b) Show how the "elastic net" fitted coefficient vector \( \hat{\beta}_{elastic\ net} \) could be found using lasso software and an appropriate augmented data set.

8. Beginning in Section 5.6, Izenman's book uses an example where PET yarn density is to be predicted from its NIR spectrum. This is a problem where \( N = 21 \) data vectors \( x_j \) of length \( p = 268 \) are used to predict the corresponding outputs \( y_i \). Izenman points out that the yarn data are to be found in the \texttt{pls} package in \( \mathbb{R} \). (The package actually has \( N = 28 \) cases. Use all of them in the following.) Get those data and make sure that all inputs are standardized and the output is centered.
a) Using the \texttt{pls} package, find the 1, 2, 3, and 4 component PCR and PLS \( \hat{\beta} \) vectors.
b) Find the singular values for the matrix \( X \) and use them to plot the function \( df(\lambda) \) for ridge regression. Identify values of \( \lambda \) corresponding to effective degrees of freedom 1, 2, 3, and 4. Find corresponding ridge \( \hat{\beta} \) vector.
c) Plot on the same set of axes \( \hat{\beta}_j \) versus \( j \) for the PCR, PLS and ridge vectors for number of components/degrees of freedom 1. (Plot them as "functions," connecting consecutive plotted
\((j, \hat{\beta}_j)\) points with line segments.) Then do the same for 2, 3, and 4 numbers of components/degrees of freedom.

d) It is (barely) possible to find that the best (in terms of \(R^2\)) subsets of \(M = 1, 2, 3,\) and 4 predictors are respectively, \(\{x_{40}\}, \{x_{212}, x_{246}\}, \{x_{25}, x_{160}, x_{215}\}\) and \(\{x_{160}, x_{169}, x_{231}, x_{243}\}\). Find their corresponding coefficient vectors. Use the lars package in R and find the lasso coefficient vectors \(\hat{\beta}\) with exactly \(M = 1, 2, 3,\) and 4 non-zero entries with the largest possible \(\sum_{j=1}^{268} |\hat{\beta}_{lasso,j}|\) (for the counts of non-zero entries).

e) If necessary, re-order/sort the cases by their values of \(i y_i\) (from smallest to largest) to get a new indexing. Then plot on the same set of axes \(i y_i\) versus \(i\) and \(\hat{y}_i\) versus \(i\) for ridge, PCR, PLS, best subset, and lasso regressions for number of components/degrees of freedom/number of nonzero coefficients equal to 1. (Plot them as "functions," connecting consecutive plotted \((i, y_i)\) or \((i, \hat{y}_i)\) points with line segments.) Then do the same for 2, 3, and 4 numbers of components/degrees of freedom/counts of non-zero coefficients.

**HW#3 (Due 2/28/11)**

9. Find a set of basis functions for the natural (linear outside the interval \((\xi_1, \xi_K)\)) quadratic regression splines with knots at \(\xi_1 < \xi_2 < \cdots < \xi_K\).

10. (B-Splines) For \(a < \xi_1 < \xi_2 < \cdots < \xi_K < b\) consider the B-spline bases of order \(m\), \(\{B_{i,m}(x)\}\) defined recursively as follows. For \(j < 1\) define \(\xi_j = a\), and for \(j > K\) let \(\xi_j = b\). Define \(B_{i,1}(x) = I[\xi_i \leq x < \xi_{i+1}]\) (in case \(\xi_i = \xi_{i+1}\) take \(B_{i,1}(x) \equiv 0\)) and then

\[
B_{i,m}(x) = \frac{x - \xi_i}{\xi_{i+m} - \xi_i} B_{i,(m-1)}(x) + \frac{\xi_{i+m} - x}{\xi_{i+m} - \xi_{i+1}} B_{i+1,(m-1)}(x)
\]

(where we understand that if \(B_{i,j}(x) \equiv 0\) its term drops out of the expression above). For \(a = -0.1\) and \(b = 1.1\) and \(\xi_i = (i-1)/10\) for \(i = 1, 2, \ldots, 11\), plot the non-zero \(B_{i,3}(x)\). Consider all linear combinations of these functions. Argue that any such linear combination is piecewise quadratic with first derivatives at every \(\xi_i\). If it is possible to do so, identify one or more linear constraints on the coefficients (call them \(c_i\)) that will make \(q_e(x) = \sum_i c_i B_{3,i}(x)\) linear to the left of \(\xi_1\) (but otherwise minimally constrain the form of \(q_e(x)\)).
11. (3.23 of HTF) Suppose that columns of $X$ with rank $p$ have been standardized, as has $Y$. Suppose also that
\[ \frac{1}{N}|\langle x_j, Y \rangle| = \lambda \quad \forall j = 1, \ldots, p \]
Let $\hat{\beta}_{\text{OLS}}$ be the usual least squares coefficient vector and $\hat{Y}_{\text{OLS}}$ be the usual projection of $Y$ onto the column space of $X$. Define $\hat{\lambda}(\alpha) = \alpha X \hat{\beta}_{\text{OLS}}$ for $\alpha \in [0,1]$. Find
\[ \frac{1}{N}|\langle x_j, Y - \hat{Y}(\alpha) \rangle| \quad \forall j = 1, \ldots, p \]
in terms of $\alpha, \lambda$, and $\left( Y - \hat{Y}_{\text{OLS}} \right)' \left( Y - \hat{Y}_{\text{OLS}} \right)$. Show this is decreasing in $\alpha$. What is the implication of this as regards the LAR algorithm?

12. a) Suppose that $a < x_1 < x_2 < \cdots < x_N < b$ and $s(x)$ is a natural cubic spline with knots at the $x_i$ interpolating the points $(x_i, y_i)$ (i.e. $s(x_i) = y_i$). Let $z(x)$ be any twice continuously differentiable function on $[a,b]$ also interpolating the points $(x_i, y_i)$. Then
\[ \int_a^b (s''(x))^2 \, dx \leq \int_a^b (z''(x))^2 \, dx \]
(Hint: Consider $d(x) = z(x) - s(x)$, write
\[ \int_a^b (d''(x))^2 \, dx = \int_a^b (z''(x))^2 \, dx - \int_a^b (s''(x))^2 \, dx - 2 \int_a^b s''(x) \, dx \]
and use integration by parts and the fact that $s''(x)$ is piecewise constant.)

b) Use a) and prove that the minimizer of $\sum_{i=1}^N (y_i - h(x_i))^2 + \lambda \int_a^b (h''(x))^2 \, dx$ over the set of twice continuously differentiable functions on $[a,b]$ is a natural cubic spline with knots at the $x_i$.

13. Let $\mathcal{H}$ be the set of absolutely continuous functions on $[0,1]$ with square integrable first derivatives (that exist except possibly at a set of measure 0). Equip $\mathcal{H}$ with an inner product
\[ \langle f, g \rangle_{\mathcal{H}} = f(0) + g(0) + \int_0^1 f'(x)g'(x) \, dx \]

a) Show that
\[ R(x, z) = 1 + \min(x, z) \]
is a reproducing kernel for this Hilbert space.

b) Using Heckman's development, describe as completely as possible
\[ \arg \min_{h \in \mathcal{H}} \left( \sum_{i=1}^N (y_i - h(x_i))^2 + \lambda \int_0^1 (h'(x))^2 \, dx \right) \]
c) Using Heckman's development, describe as completely as possible
\[
\arg \min_{h \in \mathcal{H}} \left( \sum_{i=1}^{N} \left( y_i - \int_{0}^{h(t)} dt \right)^2 + \lambda \int_{0}^{1} (h'(x))^2 dx \right)
\]

HW#4

14. Suppose that with \( p = 1 \),
\[
y \mid x \sim N \left( \frac{\sin(12(x+.2))}{x+.2}, 1 \right)
\]
and \( N = 101 \) training data pairs are available with \( x_i = (i-1)/100 \) for \( i = 1, 2, \ldots, 101 \). Vardeman will send you a text file with a data set like this in it. Use it in the following.

a) Fit all of the following using first 5 and then 9 effective degrees of freedom
   i) a cubic smoothing spline,
   ii) a locally weighted linear regression smoother based on a normal density kernel, and
   iii) a locally weighted linear regression smoother based on a tricube kernel.

Plot for 5 effective degrees of freedom all of \( y_i \) and the 3 sets of smoothed values against \( x_i \).
Connect the consecutive \((x_i, \hat{y}_i)\) for each fit with line segments so that they plot as "functions."
Then redo the plotting for 9 effective degrees of freedom.

b) For all of the fits in a) plot as a function of \( i \) the coefficients \( c_i \) applied to the observed \( y_i \) in order to produce \( \hat{f}(x) = \sum_{i=1}^{101} c_i y_i \) for \( x = .05, .1, .2, .3 \). (Make a different plot of three curves for 5 degrees of freedom and each of the values \( x \) (four in all). Then redo the plotting for 9 degrees of freedom.)

15. The "point and click" JMP software will fit neural nets (with logistic sigmoidal function, \( \sigma(\cdot) \)). It has pretty good online help files, from which you should be able to figure out how to use the software. (If I can do it, you can do it!) Use that software to produce a neural net approximation for the data set in problem 14, with an error sum of squares about like those for the 9 degrees of freedom fits. Provide appropriate JMP reports/summaries. You'll be allowed to vary the number of hidden nodes for a single-hidden-layer architecture and to vary a weight for a penalty made from a sum of squares of coefficients. Each run of the routine makes several random starts of an optimization algorithm. Extract the coefficients from the JMP run and use them to plot the fitted function of \( x \) that you settle on. How does this compare to the plotted fits produced in 14?
16. A $p = 2$ data set consists of $N = 441$ training vectors $(x_{i1}, x_{i2}, y_i)$ for the distinct pairs $(x_{i1}, x_{i2})$ in the set $\{-1.0, -0.9, \ldots, 9.1, 1.0\}^2$ where the $y_i$ were generated as

$$y_i = \frac{\sin\left(10(x_{i1} + x_{i2})\right)}{10(x_{i1} + x_{i2})} + \varepsilon_i$$

for iid $N\left(0, (.02)^2\right)$ variables $\varepsilon_i$. Vardeman will send you such a data file. Use it in the following.

a) Why should you expect MARS to be ineffective in producing a predictor in this context? (You may want to experiment with the earth package in R trying out MARS.)

b) Try 2-D locally weighted regression smoothing on this data set using the loess function in R. Contour plot your fits for 2 different choices of smoothing parameters.

c) Try fitting a thin plate spline to these data. There is an old tutorial at http://www.stat.wisc.edu/~xie/thin_plate_spline_tutorial.html for using the Tps function in the fields package that might make this simple.

d) Use the neural network routines in JMP to fit these data. How complicated does the network architecture have to be in order to do a good job fitting these data? Contour plot your fit.

e) If you were going to use a structured kernel and 1-D smoothing to produce a predictor here, what form for the matrix $A$ would work best? What would be a completely ineffective choice of a matrix $A$? Use the good choice of $A$ and produce a corresponding predictor.

17. Center the outputs for the data set of problem 14. Then derive sets of predictions $\hat{y}_i$ based on $\mu(x) \equiv 0$ Gaussian process priors for $f(x)$. Plot several of those as functions on the same set of axes (along with centered original data pairs) as follows:

a) Make one plot for cases with $\sigma^2 = 1$, $\rho(\Delta) = \exp(-c\Delta^2)$, $\tau^2 = 1, 4$ and $c = 1, 4$.

b) Make one plot for cases with $\sigma^2 = 1$, $\rho(\Delta) = \exp(-c|\Delta|)$, $\tau^2 = 1, 4$, and $c = 1, 4$.

c) Make one plot for cases with $\sigma^2 = .25$, but otherwise the parameters of a) are used.

18. Center the outputs for the data set of problem 16. Then derive a set of predictions $\hat{y}_i$ based on a $\mu(x) \equiv 0$ prior for $f(x)$. Use $\sigma^2 = (.02)^2$, $\rho(x - z) = \exp\left(-2\|x - z\|^2\right)$, and $\tau^2 = .25$. How do these compare to the ones you made in problem 16?
19. Consider radial basis functions built from kernels. In particular, consider the choice
\( D(t) = \phi(t) \), the standard normal pdf.

a) For \( p = 1 \), plot on the same set of axes the 11 functions

\[
K_{\lambda}(x, \xi_j) = D \left( \frac{|x - \xi_j|}{\lambda} \right) \text{ for } \xi_j = \frac{j-1}{10}, j = 1, 2, \ldots, 11
\]

first for \( \lambda = .1 \) and then (in a separate plot) for \( \lambda = .01 \). Then make plots on the a single set of axes the 11 normalized functions

\[
N_{\lambda}(x) = \frac{K_{\lambda}(x, \xi_j)}{\sum_{j=1}^{11} K_{\lambda}(x, \xi_j)}
\]

first for \( \lambda = .1 \), then in a separate plot for \( \lambda = .01 \).

b) For \( p = 2 \), consider the 121 basis functions

\[
K_{\lambda}(x, \xi_{ij}) = D \left( \frac{|x - \xi_{ij}|}{\lambda} \right) \text{ for } \xi_{ij} = \left( \frac{i-1}{10}, \frac{j-1}{10} \right), i = 1, 2, \ldots, 11 \text{ and } j = 1, 2, \ldots, 11
\]

Make contour plots for \( K_{.1}(x, \xi_{6,6}) \) and \( K_{.01}(x, \xi_{6,6}) \). Then define

\[
N_{\lambda ij}(x) = \frac{K_{\lambda}(x, \xi_{ij})}{\sum_{m=1}^{11} \sum_{l=1}^{11} K_{\lambda}(x, \xi_{lm})}
\]

Make contour plots for \( N_{.1,6,6}(x) \) and \( N_{.01,6,6}(x) \).

HW#5

20. Figure 4.4 of HTF gives a 2-dimensional plot of the "vowel training data" (available on the book’s website at [http://www-stat.stanford.edu/~tibs/ElemStatLearn/index.html](http://www-stat.stanford.edu/~tibs/ElemStatLearn/index.html) or from [http://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+%28Vowel+Recognition+-+Deterding+Data%29](http://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+%28Vowel+Recognition+-+Deterding+Data%29). The ordered pairs of first 2 canonical variates are plotted to give a "best" reduced rank LDA picture of the data like that below (lacking the decision boundaries).

a) Use Section 10.1 of the course outline to reproduce Figure 4.4 of HTF (color-coded by group, with group means clearly indicated). Keep in mind that you may need to multiply one or both of your coordinates by \(-1\) to get the exact picture.
Use the \texttt{R} function \texttt{lda} (MASS package) to obtain the group means and coefficients of linear discriminants for the vowel training data. Save the \texttt{lda} object by a command such as \texttt{LDA=lda(insert formula, data=vowel)}.

\textbf{b)} Reproduce a version of Figure 1a) above. You will need to plot the first two canonical coordinates as in a). Decision boundaries for this figure are determined by classifying to the nearest group mean. Do the classification for a fine grid of points covering the entire area of the plot. If you wish, you may plot the points of the grid, color coded according to their classification, instead of drawing in the black lines.

\textbf{c)} Make a version of Figure 1b) above with decision boundaries now determined by using logistic regression as applied to the first two canonical variates. You will need to create a data frame with columns \texttt{y}, \texttt{canonical variate 1}, and \texttt{canonical variate 2}. Use the \texttt{vglm} function (VGAM package) with \texttt{family=multinomial()} to do the logistic regression. Save the \texttt{lda} object by a command such as \texttt{LR=vglm(insert formula, family=multinomial(), data=data set)}. A set of observations can now be classified to groups by using the command \texttt{predict(LR, newdata, type='response')}, where \texttt{newdata} contains the observations to be classified. The outcome of the predict function will be a matrix of probabilities. Each row contains the probabilities that a corresponding observation belongs to each of the groups (and thus sums to 1). We classify to the group with maximum probability. As in b), do the classification for a fine grid of points covering the entire area of the plot. You may again plot the points of the grid, color-coded according to their classification, instead of drawing in the black lines.

Figure 1: Decision boundaries for the vowel training data, in the two-dimensional subspace spanned by the first two canonical variates, according to a) nearest mean and b) logistic regression.

![Figure 1](image-url)

a) So that you can see and plot results, first use the 2 canonical variates employed in problem 20 and use \texttt{rpart} in R to find a classification tree with empirical error rate comparable to the reduced rank LDA classifier pictured in Figure 1a. Make a plot showing the partition of the region into pieces associated with the 11 different classes. (The intention here is that you show rectangular regions indicating which classes are assigned to each rectangle, in a plot that might be compared to the plots on page 9.)

b) Beginning with the original data set (rather than with the first 2 canonical variates) and use \texttt{rpart} in R to find a classification tree with empirical error rate comparable to the classifier in a). How much (if any) simpler/smaller is the tree here than in a)?


Compare an appropriate SVM based on the original input variables to a classifier based on logistic regression with the same variables. (BTW, there is a nice paper "Support Vector Machines in R" by Karatzoglou, Meyer, and Hornik that appeared in Journal of Statistical Software that you can get to through ISU. And there is an R SVM tutorial at http://planatscher.net/svmtut/svmtut.html.)

23. (Izenman Problem 14.4.) Consider the following 10 two-dimensional points, the first five points, (1, 4),(3.5,6.5),(4.5,7.5),(6,6),(1.5,1.5) belong to Class 1 and the second five points (8,6.5),(3, 4.5),(4.5,4),(8,1.5),(2.5,0) , belong to Class 2. Plot these points on a scatterplot using different symbols or colors to distinguish the two classes. Carry through by hand the AdaBoost algorithm on these points, showing the weights at each step of the process. (Presumably Izenman's intention is that you use simple 2-node trees as your base classifiers.) Determine the final classifier and calculate its empirical misclassification rate.

24. Below is a very small sample of fictitious $p = 1$ training data.

$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
x & 1 & 2 & 3 & 4 & 5 \\
y & 1 & 4 & 3 & 5 & 6 \\
\end{array}
$$

Consider a toy Bayesian model averaging problem where what is of interest is a prediction for $y$ at $x = 3$.

Suppose that Model #1 is as follows. \((x_i,y_i) \sim \text{iid } P\) where

\[
x \sim \text{Discrete Uniform on } \{1, 2, 3, 4, 5\} \text{ and } y \mid x \sim \text{Binomial}(10, p(x)) \quad \text{for } p(x) = \Phi\left(\frac{x-a}{b}\right)
\]

where \(\Phi\) is the standard normal cdf and \(b > 0\). In this model, the quantity of interest is

\[
10 \cdot \Phi\left(\frac{3-a}{b}\right).
\]
On the other hand, suppose that Model #2 is as follows. \((x_i, y_i) \sim \text{iid } P\) where 
\[ x \sim \text{Discrete Uniform on } \{1, 2, 3, 4, 5\} \text{ and } y \mid x \sim \text{Binomial}(10, p(x)) \text{ for } p(x) = 1 - \frac{1}{(c+1)x} \]
where \(c > 0\). In this model, the quantity of interest is 
\[ 10 \left(1 - \frac{1}{3(c+1)}\right). \]

For prior distributions, suppose that for Model #1 \(a\) and \(b\) are a priori independent with 
\(a \sim \text{Unif}(0, 6)\) and \(b^{-1} \sim \text{Exp}(1)\). While for model #2, \(c \sim \text{Exp}(1)\). Further suppose that prior probabilities on the models are \(\pi(1) = \pi(2) = 0.5\). Compute (almost surely you'll have to do this numerically) posterior means of the quantities of interest in the two Bayes models, posterior probabilities for the two models, and the overall predictor of \(y\) for \(x = 3\).

25. Consider the \(N = 50\) training set and predictors of Problem 2. Find (according to Section 9.2.2 of the class outline) \(w^{\text{stacked}}\) and hence identify a stacked predictor of \(y\). How does \(\text{err}\) for this stacked predictor compare to those of the two predictors being combined?

26. Get the famous \(p = 2\) "Ripley dataset" (\texttt{synth.tr}) commonly used as a classification example from the author's site \url{http://www.stats.ox.ac.uk/pub/PRNN/}. Using several different values of \(\lambda\) and constants \(c\), find \(g \in \mathcal{H}_x\) and \(\beta_0 \in \mathbb{R}\) minimizing
\[
\sum_{i=1}^{N} \left(1 - y_i (\beta_0 + g(x_i))\right)^2 + \lambda \|g\|_{\mathcal{H}_x}^2
\]
for the (Gaussian) kernel \(K(x, z) = \exp\left(-c\|x - z\|^2\right)\). Make contour plots for those functions \(g\), and, in particular, show the \(g(x) = 0\) contour that separates \([-1.5, 1.0] \times [-2, 1.3]\) into the regions where a corresponding SVM classifies to classes \(-1\) and \(1\).

**HW#6**

Have a look at the paper "\texttt{ada}: An \texttt{R} Package for Stochastic Boosting" by Mark Culp, Kjell Johnson, and George Michailidis that appeared in the \textit{Journal of Statistical Software}. It provides perspective and help with both the \texttt{R} packages \texttt{ada} and \texttt{randomforest}.

27. Use both an adaBoost algorithm and a random forest to identify appropriate classifiers based on the Ripley dataset. For a fine grid of points in \([-1.5, 1.0] \times [-2, 1.3]\), indicate on a 2-D plot which points get classified to classes \(-1\) and \(1\) so that you can make visual comparisons to the SVM classifiers referred to in problem 26.
28. The Culp et al. paper discusses using a one-versus-all strategy to move adaBoost to a multi-class problem. Apparently, this is known elsewhere as the "adaBoost.mh" algorithm. Return to the use of the first two canonical coordinates of the vowel training data (as in problem 21). Find both an appropriate random forest classifier and a adaBoost.mh classifier for the 11-class problem with $p = 2$, and once more show how the classifiers break the 2-D space up into regions to be compared to the plots in problem 20.