Stats 202 Practice Problems

Stats 202 Course Staff

Fall 2013 and 2014

1. Perform K-means clustering on the following 2-dimensional observations with $K = 3$ and initial labels $(1, 1, 2, 3, 3, 2)$. Use the *Manhattan* distance between a pair of points: $d(A, B) = |X_{1A} - X_{1B}| + |X_{2A} - X_{2B}|$, instead of the Euclidean distance. With this distance, the centroid of a cluster is obtained by taking the median of the samples in each dimension.

Obs.	X_1	X_2
A	1	4
B	1	3
C	3	4
D	5	2
E	3	2
F	3	0

Solution. The centroids of the three clusters are $(1, 3.5), (3, 2)$ and $(4, 2)$. The distances between the observations and these centroids is shown in the following table.

dist.	C1	C2	C3
A	0.5	4	5
B	0.5	3	4
C	2.5	2	3
D	5.5	2	1
E	3.5	0	1
F	5.5	2	3

So the new labels after the first iteration are $(1, 1, 2, 3, 2, 2)$, and the centroids of the new clusters are $(1, 3.5), (3, 2)$ and $(5, 2)$. Since only one centroid changed, we just need to compute the distances between the third centroid and the data points.

dist.	C1	C2	C3
A	0.5	4	6
B	0.5	3	5
C	2.5	2	4
D	5.5	2	0
E	3.5	0	2
F	5.5	2	4

The clusters remain the same, so the result is $\{A, B\}, \{C, E, F\}$ and $\{D\}$.

2. Perform single-linkage hierarchical clustering on the data of problem 1, using the Manhattan distance.

   Solution. We compute the distance matrix between the samples:

dist.	A	B	C	D	E	F
A		1	2	6	4	6
B			3	5	3	5
C				4	2	4
D					2	4
E						2
F						

   The smallest distance is that between A and B, so we form a new cluster {A,B}. Then, we recompute the distance matrix between our current set of clusters:

dist.	{A, B}	C	D	E	F
{A,B}		2	5	3	5
C			4	2	4
D				2	4
E					2
F					

   At this second step, we link all the clusters, because every cluster is within a distance 2 to another cluster. We obtain the single cluster {A, B, C, D, E, F}.

3. Every single linkage hierarchical clustering satisfies the following property:

   Let ℓ be the smallest single-linkage distance between any pair of clusters. For any pair of samples i and j in the same cluster, we can find a chain of samples within the same cluster connecting i and j such that the distance between two consecutive samples is at most ℓ.

   Provide a proof for this fact.

   The fact obviously holds for the clustering which has one sample per cluster. We prove the fact by induction, showing that the property is preserved in each step of the single-linkage agglomerative algorithm.

   Suppose the property holds for a clustering with k clusters. Let ℓ_m be the minimal intercluster single-linkage distance for a clustering with m clusters. Suppose clusters A and B merge when we go from k clusters to $k-1$ clusters. Clearly, $\ell_{k-1} \geq \ell_k$.

   For any pair of samples i and j which are both in the same cluster different from A and B, both in A, or both in B, we have by assumption a chain of states connecting i and j where consecutive samples are separated at most by ℓ_k, so they are also separated at most by ℓ_{k-1}. On the other hand, for a pair of samples $i \in A$ and $j \in B$, we can define a chain in the following way. Let i' and j' be the points in A and B, respectively, that are closest together. Then, we have a chain from i to i' and a chain from j to j' which satisfy the property, and if we concatenate them, we obtain a chain from i to j which satisfies the property, because the distance between i' and j' is $\ell_k \leq \ell_{k+1}$.

4. We fit a linear regression model $Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$ to some data. Suppose we change the units of the predictors X_i, to obtain a new set of predictors $Z_i = cX_i$. Then, we fit the same data to the model: $Y = \alpha_0 + \alpha_1 Z_1 + \ldots + \alpha_p Z_p$.

   (a) What is the relationship between the least squares coefficients $\hat{\alpha} = (\hat{\alpha}_1, \ldots, \hat{\alpha}_p)$ and $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_p)$? Provide a proof.

   (b) What is the relationship between the fitted values in the two models?

   Solution.

   (a) In the first case, we are solving the optimization:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 X_1 - \cdots - \beta_p X_p)^2.$$

   In the second case, we are solving the optimization:

$$\min_{\alpha} \sum_{i=1}^{n} (y_i - \alpha_0 - \alpha_1 c X_1 - \cdots - \alpha_p c X_p)^2.$$

   The second problem is equivalent to the first after the change of variables $\beta_0 = \alpha_0$, $\beta_i = \alpha_i c$ for $i > 0$. This change of variables is one-to-one, i.e. every vector β maps to a unique vector α and vice versa. Therefore, the minimizer of the first problem maps to the minimizer of the second problem, or $\hat{\beta}_0 = \hat{\alpha}_0$, $\hat{\beta}_i = c\hat{\alpha}_i$ for $i > 0$.

   (b) By part (a), the fitted values for the two models are equal: $\hat{y}_i = x_i^T \hat{\beta} + \hat{\beta}_0 = c x_i^T \hat{\beta}/c + \hat{\beta}_0 = z_i^T \hat{\alpha} + \hat{\alpha}_0$. One can also recall that the fitted values are a projection of the response vector y onto the column space of the predictor matrix, and $\mathbf{X}$ and $\mathbf{Z}$ have the same column space.

5. Your colleague fitted a multivariate linear regression model $Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$ and found all but three p-values are significant in the t-test. He decides to drop those three variables and keep all the remaining predictors. What do you think of your colleague's method?

   Solution. The colleague's method is not reasonable because the t-test only tests the marginal effect of each predictor. If there is collinearity, it is possible that after removing one predictor, formerly insignificant predictors become significant. If p is large, choosing significant predictors at a fixed significance level could also lead to a large number of false positives. A better approach would be to apply backward stepwise selection, for example.

6. Explain the purpose of an F-test for multiple linear regression.

   Solution. The F-test tests the hypothesis that a set of coefficients in the linear model are equal to zero. The t-test is a special case of the F-test when the set contains a single coefficient.

7. True or false: The variance of a regression estimator $\hat{f}$ in the bias-variance decomposition can be written:

$$\frac{1}{n-1} \sum_{i=1}^{n} (\hat{f}(x_i) - m)^2,$$

where

$$m = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_i),$$

and $x_1, \ldots, x_n$ are the inputs of the training data.

Solution. False. This is the variance of the fitted values for a specific training sample. The variance in the bias-variance decomposition measures the variability of the function $\hat{f}$ with respect to resampling the training data.

8. Suppose the data $(\mathbf{X}, \mathbf{y})$ are well fit by a linear model, how would you diagnose if the data point (x_i, y_i) is an outlier or a high leverage point?

   Solution. An outlier is a point for which y_i is far from the value predicted by the model. We can look at the studentized residual to check if a point is an outlier. In order to know if (x_i, y_i) has high leverage, we need to compute its leverage statistic h_i.

9. Suppose we have a dataset with N observations, and each observation consists of three values:

   - y: binary variable that is 1 if a student passed and 0 if a student failed the exam
   - x_1: the number of hours spent studying for the exam
   - x_2: a binary variable indicating whether or not the student passed the previous exam.

   Suppose upon fitting a logistic regression of the y on x_1, x_2, and an intercept, the estimates for $\beta = (\beta_0, \beta_1, \beta_2)$ are

$$\begin{aligned} \hat{\beta}_0 &= -1.2 \\ \hat{\beta}_1 &= 0.3 \\ \hat{\beta}_2 &= 1.2 \end{aligned}$$

   Now suppose instead of using the number of hours spent studying, we used the number of minutes spent using for the exam. Can you identity what the new β_0, β_1, and β_2 would be? Why or why not? Justify your claim.

   Solution.

   Logistic regression is fit by maximizing the likelihood function, which can be written in the following way:

$$\hat{\beta} = \arg\max_{\beta} \prod_{i,y_i=1} \frac{e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}}}{1+e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}}} \prod_{i,y_i=0} \frac{1}{1+e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}}}.$$

   Since this only depends on the parameters through a linear function, we can apply the same argument used in problem 3. Letting

$$\tilde{\beta} = \arg\max_{\beta} \prod_{i,y_i=1} \frac{e^{\beta_0+\beta_1 60 x_{i1}+\beta_2 x_{i2}}}{1+e^{\beta_0+\beta_1 60 x_{i1}+\beta_2 x_{i2}}} \prod_{i,y_i=0} \frac{1}{1+e^{\beta_0+\beta_1 60 x_{i1}+\beta_2 x_{i2}}},$$

   the maximizers $\hat{\beta}$ and $\tilde{\beta}$ are related by

$$\begin{aligned} \tilde{\beta}_0 &= \hat{\beta}_0 = -1.2 \\ \tilde{\beta}_1 &= \frac{1}{60}\hat{\beta}_1 = 0.005 \\ \tilde{\beta}_2 &= \hat{\beta}_2 = 1.2 \end{aligned}$$

10. Suppose we have a classification problem with a binary response Y and a p-dimensional predictor variable $X = (X_1, \ldots, X_p)$. Logistic regression is fitted to a set of n samples. Then, logistic regression is fitted again to the same observations, where we include one additional predictor, such that:

$$X = (X_1, \ldots, X_p, X_{p+1}).$$

Explain how the training error, test error, and coefficients change in each of the following cases:

(a) $X_{p+1} = X_1 + 2X_p$.

(b) X_{p+1} is a random variable independent of Y.

(a) Since the new predictor is exactly collinear with 2 of the old predictors, the coefficients β_1, β_p, and β_{p+1} are unidentifiable, as logistic regression maximizes a likelihood which only depends on a linear combination of the predictors. The predictions remain unchanged, and therefore so do the training and test errors.

(b) Since the number of samples is finite, logistic regression may assign a positive coefficient to X_{p+1} even though it is independent of the response; this will likely affect other coefficients as well. The training error can only decrease, whereas the test error will increase because the bias remains the same while variance increases.

11. A scientist performs a ridge regression in R and estimates the optimal parameter λ by 10-fold cross-validation. She uses the following code:

```
> library(glmnet)
> X = as.matrix(data[,-1])
> Y = as.vector(data[,1])
> cv.out = cv.glmnet(X,Y,alpha = 0)
> cv.out$lambda.min
[1] 0.5
> min(cv.out$cvm)
[1] 485.1199
```

The scientist concludes that the test MSE of ridge regression with $\lambda = 0.5$ is approximately 485.119. Explain the problem with this estimate and suggest a better way to estimate the test MSE.

Solution. This estimate is slightly biased downward. The reason is that we have used all the training data to select the optimal λ, and then used the same data to estimate the test error with $\lambda = 0.5$. The model with the optimal λ is already fit to all the data, so the left-out folds are not entirely independent of the model. A better estimate of the test MSE could be produced by first splitting the data into training and validation sets, selecting the optimal λ by 10-fold cross validation on the training set, and calculating the test MSE on the validation set.

12. On the make believe island of Statlantis, there's a volcano which periodically erupts. It's known the times between eruptions are random variables $X_1, X_2, \ldots$, which are independent and uniformly distributed in the interval $[0, \theta]$, but θ is unknown. Suppose we observe n of the times between eruptions $x_1, x_2, \ldots, x_n$, and we wish estimate θ using the estimator

$$\hat{\theta} = \frac{2}{n} \sum_{i=1}^{n} x_i,$$

that is, the average of the observations multiplied by 2.

(a) We can estimate the standard error of $\hat{\theta}$ using the bootstrap. In this example, it is possible to compute the boostrap error analytically (the error we would estimate from an infinite number of bootstrap resamplings). Find an expression for this standard error as a function of the sample variance $\hat{\sigma}^2$ of the observed times $x_1, \ldots, x_n$.

(b) Find the standard error of $\hat{\theta}$ analytically. Express your answer as a function of the variance of the time between eruptions σ^2.

(c) Argue that the bootstrap standard error will be similar to the true standard error when n is large.

Hint: The variance of a sum of independent random variables is the sum of their variances.

Solution.

(a) Let $Z_1, \ldots, Z_n$ be a bootstrap resampling of the observations $\{x_1, \ldots, x_n\}$. The estimator of θ derived from the bootstrap sample is:

$$\hat{\theta}_Z = \frac{2}{n} \sum_{i=1}^{n} Z_i$$

The bootstrap approximates the variance of this estimator using a Monte Carlo method. In this case, we can derive the true variance of the estimator using the linearity of variance in the hint above:

$$\mathrm{Var}(\hat{\theta}_Z) = \frac{4}{n^2} \mathrm{Var}\left(\sum_{i=1}^{n} Z_i \right) = \frac{4}{n^2} \sum_{i=1}^{n} \mathrm{Var}(Z_i) = \frac{4\hat{\sigma}^2}{n}.$$

The standard error is the standard deviation of the estimator or $2\sqrt{\hat{\sigma}^2/n}$.

(b) The actual standard error of the estimator $\hat{\theta}$ can be calculated in a similar fashion. The variance of the sampling distribution of $\hat{\theta}$ is

$$\mathrm{Var}(\hat{\theta}) = \frac{4}{n^2} \sum_{i=1}^{n} \mathrm{Var}(X_i) = \frac{4\sigma^2}{n}.$$

So, the standard error of the estimator is $2\sqrt{\sigma^2/n}$.

(c) When n is large, the sample variance of $x_1, \ldots, x_n$ will be close to the actual variance of the time between eruptions. Thus, parts (a) and (b) imply that the bootstrap standard error will be close to the actual standard error.

13. What are the key differences between LOOCV and k-fold cross validation for estimating the true test error? Is one better than the other, and if so, why?

    In LOOCV, because (nearly) the whole dataset is used to train the model, it will be almost unbiased for estimating the test error. This is because training models with more data generally leads to a lower test error, and because k-fold CV uses less training data than LOOCV, it will overestimate the test error more than LOOCV.

    However, because the LOOCV estimates share a large chunk of the training set for each iteration, the estimates of the test error are highly correlated. By construction, the k-fold CV estimates are less correlated (since they share less of the training set across folds), and thus the variance of the average of the k-fold estimates might have much less variance than the average of the LOOCV estimates. Hence there's a tradeoff between the two.

    In summary, neither is definitively "better" than the other, although k-fold CV is used more often in practice.

14. Suppose we are trying to model a classification problem with two labels: 'sick' and 'healthy.' For the purpose of this test, we consider a positive result to be testing sick and a negative to test as healthy. After fitting the model with LDA in R, we compare predicted values from the actual values, as shown below:

```
> lda.fit = lda(test.result ~ x1 + x2, data=mydata)
> lda.pred = predict(lda.fit)
> lda.class=lda.pred$class
> table(lda.class, mydata$test.result)
         mydata
lda.pred sick healthy
sick     40   32
healthy  25   121
```

    What is the misclassification rate for the model above? How can we decrease the rate of false positives to false negatives in LDA? Why might we want to do that and, assuming LDA is a good model for the data, how is this likely to affect the misclassification rate?

    Solution. The misclassification rate is the proportion of observations that are misclassified:

$$\frac{32+25}{32+40+25+121} = \frac{57}{218}.$$

    LDA classifies to positive if the estimated posterior $P(\text{sick}|x_1, x_2)$ is greater than 0.5. We can decrease the rate of false positives if we make it harder to classify to positive by increasing this threshold. We might want to do that if a false positive causes more harm than a false negative. If LDA is a good model for the data, the threshold of 0.5 optimizes the misclassification rate, so increasing the threshold would likely decrease the rate of false positives at the expense of increasing the misclassification rate.

15. Consider the dataset

x	y
-2	'slow'
5	'fast'
-1	'slow'
10	'fast'
4	'fast'

Suppose we used logistic regression to fit this model: that is, if y is a binary variable that is either 'fast' or 'slow', we wish to fit the model

$$\mathbb{P}(y_i = \text{fast}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_i)}} \qquad \mathbb{P}(y_i = \text{slow}) = \frac{e^{-(\beta_0 + \beta_1 x_i)}}{1 + e^{-(\beta_0 + \beta_1 x_i)}}$$

for all $i = 1, \ldots, 5$. What value(s) of β would maximize the likelihood (and thus be the estimates returned from fitting this model)?

Solution. Notice that y is the label 'slow' if and only if $x < 4$. That is, the two classes can be perfectly separated. We know that in this case, logistic regression produces unstable results. In particular, if we choose $\beta_0 = 0$ and let $\beta_1 \to \infty$, the probability of each observation approaches 1. Therefore, our objective can only get better as we increase β_1, and there is no maximizer of the likelihood.

16. Suppose that $\mathbf{X}$ is an $n \times p$ matrix of predictors and $\mathbf{y}$ is a quantitative response. Suppose that $p > n \geq 1000$. You want to fit a linear model that helps you make predictions with new data. Explain which of the following methods could be applied, and the advantages of each one.

    (a) Least squares

    (b) Lasso

    (c) Ridge regression

    (d) Backward stepwise selection

    (e) Forward stepwise selection

    (f) Best subset selection

    Solution. The least squares problem does not have a solution when $p > n$. Backward stepwise selection cannot be applied because you need to start with a model with all predictors, which is not well-defined. Best subset selection cannot be applied because there is a huge number of models with k predictors, even for small values of k.

    The Lasso will produce sparse solutions, setting some coefficients to zero, which is desirable from the point of view of interpretation. Ridge regression is another way of regularizing the problem, but redundant or collinear predictors will be given positive coefficients. Forward stepwise selection is another way to produce a sparse model.

17. Consider selecting subsets of predictors in the standard linear model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$$

We run best subset selection, forward stepwise selection, and backward stepwise selection using RSS as the criterion and get the list of models:

$$\text{Best subset selection} : \mathcal{M}_0^{(bs)}, \mathcal{M}_1^{(bs)}, \cdots \mathcal{M}_p^{(bs)}$$
$$\text{Forward stepwise selection} : \mathcal{M}_0^{(forward)}, \mathcal{M}_1^{(forward)}, \cdots, \mathcal{M}_p^{(forward)}$$
$$\text{Backward stepwise selection} : \mathcal{M}_p^{(backward)}, \mathcal{M}_{p-1}^{(backward)}, \cdots, \mathcal{M}_0^{(backward)}$$

For example, $\mathcal{M}_k^{(bs)}$ is the 'best' model among all the models with k predictors and $\mathcal{M}_k^{(forward)}$ is the 'best' model among the $p-k$ candidate models after choosing $\mathcal{M}_{k-1}^{(forward)}$.
Let $RSS_{\mathcal{M}}$ the training error fitted by model $\mathcal{M}$. Show that

(a) $RSS_{\mathcal{M}_p^{(forward)}} = RSS_{\mathcal{M}_p^{(backward)}}$.
(b) $RSS_{\mathcal{M}_1^{(forward)}} \leq RSS_{\mathcal{M}_1^{(backward)}}$.

Solution.

(a) $\mathcal{M}_p^{(forward)}$ and $\mathcal{M}_p^{(backward)}$ contains all p predictors in the model. Thus, they give the same RSS.

(b) $\mathcal{M}_1^{(forward)}$ is the best model among all the models with 1 predictor because $\mathcal{M}_0^{(forward)}$ is the null model. Thus $\mathcal{M}_1^{(forward)} = \mathcal{M}_1^{(bs)}$ and they have the same RSS. Noting that the best subset method selects the model with smallest RSS, we get $RSS_{\mathcal{M}_1^{(forward)}} = RSS_{\mathcal{M}_1^{(bs)}} \leq RSS_{\mathcal{M}_1^{(backward)}}$.

18. Ridge regression solves the following optimization problem:

$$\min_\beta \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Suppose we fit a ridge regression model for a single predictor X_1 and intercept with parameter $\lambda = \delta > 0$ and get $\hat{\beta}_0 = a$ and $\hat{\beta}_1 = b$. Now we include additional predictor X_2 which satisfies $X_2 = X_1$ and fit a ridge regression model with parameter 2δ. Express the fitted model in terms of a and b and compare with the previous fitted model.

Hint: $2(\beta_1^2 + \beta_2^2) \geq (\beta_1 + \beta_2)^2$ with equality only when $\beta_1 = \beta_2$.

Solution. Note that

$$\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i,1} - \beta_2 x_{i,2})^2 + 2\delta(\beta_1^2 + \beta_2^2)$$
$$= \sum_{i=1}^{n} (y_i - \beta_0 - (\beta_1 + \beta_2) x_{i,1})^2 + 2\delta(\beta_1^2 + \beta_2^2) \geq \sum_{i=1}^{n} (y_i - \beta_0 - (\beta_1 + \beta_2) x_{i,1})^2 + \delta(\beta_1 + \beta_2)^2$$

and the right hand side is minimized when $\beta_0 = a,\ \beta_1 + \beta_2 = b$. Since equality holds if and only if $\beta_1 = \beta_2$, the minimizer is $\beta_0 = a,\ \beta_1 = \beta_2 = \frac{b}{2}$.
Thus, the fitted model is

$$\hat{Y} = a + \frac{b}{2}X_1 + \frac{b}{2}X_2$$

which is the same model as $\hat{Y} = a + bX_1$. Note that collinearity did not affect the solution.

19. In this problem, we compare the lasso estimator with the best subset estimator.
Lasso regression solves the following optimization problem:

$$\min_\beta \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

Let $\hat{\beta}^{lasso}$ the solution of this problem with s nonzero $\hat{\beta}_j^{lasso}$s. i.e., $\sum_{j=1}^{p} 1(\hat{\beta}_j^{lasso} \neq 0) = s$.
Consider best subset method of size s which solves the following optimization problem:

$$\min_\beta \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j}\right)^2$$
$$\text{subject to } \sum_{j=1}^{p} 1(\beta_j \neq 0) = s$$

Let $\hat{\beta}^{bs}$ the solution of this problem. Prove that

$$\sum_{j=1}^{p}|\hat{\beta}_j^{lasso}| \leq \sum_{j=1}^{p}|\hat{\beta}_j^{bs}|$$

Can we say that the lasso estimator is a shrinkage estimator of the best subset estimator?

Solution.

$$\sum_{i=1}^{n}\left(y_i - \hat{\beta}_0^{lasso} - \sum_{j=1}^{p}\hat{\beta}_j^{lasso}x_{i,j}\right)^2 + \lambda\sum_{j=1}^{p}|\hat{\beta}_j^{lasso}|$$
$$\leq \sum_{i=1}^{n}\left(y_i - \hat{\beta}_0^{bs} - \sum_{j=1}^{p}\hat{\beta}_j^{bs}x_{i,j}\right)^2 + \lambda\sum_{j=1}^{p}|\hat{\beta}_j^{bs}|$$
$$\leq \sum_{i=1}^{n}\left(y_i - \hat{\beta}_0^{lasso} - \sum_{j=1}^{p}\hat{\beta}_j^{lasso}x_{i,j}\right)^2 + \lambda\sum_{j=1}^{p}|\hat{\beta}_j^{bs}|$$

The first inequality holds because $\hat{\beta}_j^{lasso}$ is the optimal solution of the first optimization problem. The second inequality holds because $\hat{\beta}_j^{lasso}$ has s non-zero elements and thus $\hat{\beta}_j^{bs}$ gives us a smaller RSS. Therefore,

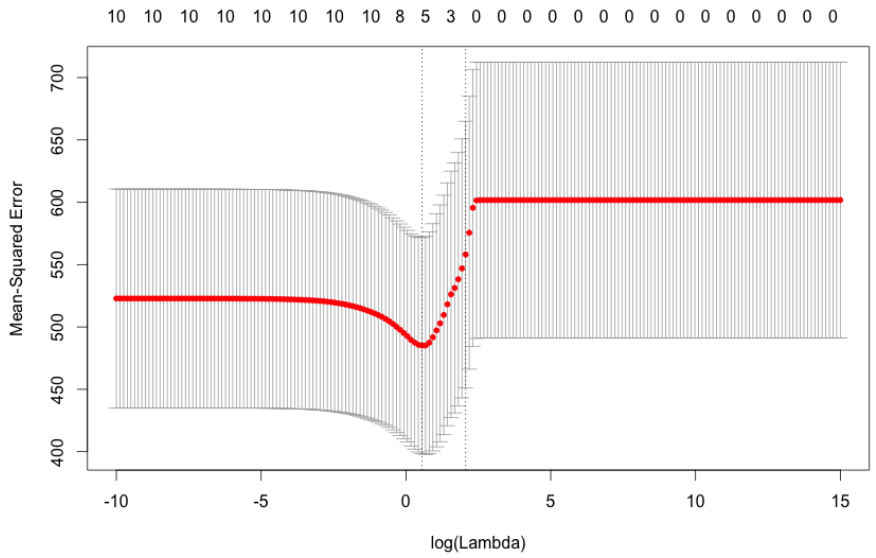
$$\sum_{j=1}^{p}|\hat{\beta}_j^{lasso}| \leq \sum_{j=1}^{p}|\hat{\beta}_j^{bs}|$$

However, there is no guarantee that the predictors with non-zero estimated coefficients match in lasso regression and best subset method.

20. When the number of observations and predictors are large, best subset selection can be computationally intensive. However, if we ignore this issue, best subset selection might be preferred to forward stepwise selection because every model that appears in forward stepwise selection algorithm is considered in best subset selection. Explain the reason that forward selection can still be preferred.

    Solution. If we compare all the possible models, we increase our chance of overfitting. Restricting our search space for the best model can reduce the variance of the model, which can give a smaller prediction error.

21. The plot below displays the cross-validation errors of lasso regression computed on a range of tuning parameter λ. There are $p = 10$ parameters in the model, and the training set and test set has $n = 100$ observations each.



Cross-validation error plot for lasso regression

```
> cv.out$cvm[1]
[1] 601.7324
> cv.out$cvm[200]
[1] 522.8358
> min(cv.out$cvm)
[1] 485.1199
```

What are the two models marked with dashed vertical lines in the plot? Would you expect an appropriate shrinkage method to give a smaller test MSE than the test MSE of multiple linear regression for this data?

Solution. The two dashed lines mark the model with the smallest cross validation error and the model chosen with the 1 standard error rule; that is, the simplest model whose cross validation error is within one standard error of the minimum cross validation error. The lasso regression when $\lambda \to 0$ corresponds to the linear regression with (10+1) parameters including

the intercept and the lasso regression when $\lambda \to \infty$ corresponds to the linear regression with intercept only. We observe that the lasso regression with a certain tuning parameter has the smallest cv error. Since the mean-squared errors from cross-validation are estimates of the test MSE, we expect that the lasso regression using the best λ from cross-validation will give a smaller test MSE than the test MSE of multiple linear regression.

22. What is the main advantage of the Lasso with respect to Ridge regression?

    Solution. In addition to reducing the variance of the regression estimate through shrinkage, the Lasso performs variable selection by setting certain coefficient to 0.

23. Assume 2 predictors and a quantitative outcome have a linear relationship $y_i = x_{i1}\beta_1 + x_{i2}\beta_2 + \varepsilon_i$, where ε_i is i.i.d. noise of unit variance. You have a training set $(\mathbf{X}, \mathbf{y})$, where $\mathbf{X}$ has full column rank and

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}.$$

    Let $\hat{y}_\lambda$ be the prediction of ridge regression at a test point with predictors (x_{01}, x_{02}). It can be shown that

$$\begin{aligned} \text{Bias}(\hat{y}_\lambda) &= \lambda\left(\frac{\beta_1 x_{01}}{d_1+\lambda} + \frac{\beta_2 x_{02}}{d_2+\lambda}\right) \\ \text{Var}(\hat{y}_\lambda) &= \left[\frac{d_1 x_{01}^2}{(d_1+\lambda)^2} + \frac{d_2 x_{02}^2}{(d_2+\lambda)^2}\right]. \end{aligned}$$

    Prove that ridge regression achieves a strictly lower test MSE at (x_{01}, x_{02}) than least squares regression for a positive value of λ.

    Solution. The solution of ridge regression for $\lambda = 0$ is identical to the least squares solution. We will prove that the derivative of the test MSE with respect to λ is strictly negative at $\lambda = 0$, so there exist a positive λ which yields a strictly lower test MSE. The test MSE is a sum of squared bias, variance, and the irreducible error, which in this case is 1 and doesn't depend on λ. The derivative of the squared bias is equal to 0 at $\lambda = 0$. On the other hand

$$\frac{d\text{Var}(\hat{y}_\lambda)}{dy} = -2\left[\frac{d_1 x_{01}^2}{(d_1+\lambda)^3} + \frac{d_2 x_{02}^2}{(d_2+\lambda)^3}\right]$$

    is negative at $\lambda = 0$ because at least one sample variance d_1 and d_2 is strictly positive.

24. A new startup has developed a cheap way to measure the expression of antibodies in infants. They hope to use these data to classify infants according to whether they will develop atopic syndrome. The company has collected data from 500 infants and followed them for 5 years to evaluate the incidence of atopy. They ran an initial trial in which they measured just 1,000 antibodies, and then a second trial which considered a more complete library of 100,000 antibodies. In each case, they performed variable selection using forward stepwise selection and produced a classification by logistic regression.

    They found that the classification quality was worse in the second trial. Puzzled by this outcome, they decided to consult with you. How would you explain this observation?

    Solution. This problem is in the realm of high-dimensional statistics, since p is significantly larger than n. Even though some antibodies may be predictive of the disease, adding predictors which are not correlated with the response may hurt the performance of the method.

This is because the number of patients is relatively small and their data are used to perform variable selection, which is more difficult if the significant variables are obscured by many insignificant predictors.

25. Variable Y is generated by the following formula:

$$Y = 1 + X + X^2 + \epsilon$$

where $X, \epsilon \sim N(0, 1)$ and X, ϵ are independent. 100 samples are generated.

```
set.seed(1)
x=rnorm(100)
e=rnorm(100)
y = 1+ x + x^2 + e
```

We apply Principal Component Regression and Partial Least Squares using predictors X, X^2, X^3, X^4 in order to predict Y variable.

```
> pcr.fit<-pcr(y~.,data=x_data,scale=TRUE,validation="CV")
> summary(pcr.fit)
Data:  X dimension: 100 4
Y dimension: 100 1
Fit method: svdpc
Number of components considered: 4

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps
CV           1.733    1.226    1.018   0.9560   0.9780
adjCV        1.733    1.153    1.016   0.9544   0.9754

TRAINING: % variance explained
   1 comps  2 comps  3 comps  4 comps
X    56.37    93.90    98.44    100.0
y    66.77    66.88    70.29     70.3
> pls.fit<-plsr(y~.,data=x_data,scale=TRUE,validation="CV")
> summary(pls.fit)
Data:  X dimension: 100 4
Y dimension: 100 1
Fit method: kernelpls
Number of components considered: 4

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps
CV           1.733    1.003   0.9750   0.9686   0.9868
adjCV        1.733    1.002   0.9718   0.9669   0.9838

TRAINING: % variance explained
   1 comps  2 comps  3 comps  4 comps
X    56.34    68.63    98.43    100.0
y    67.40    69.77    70.30     70.3
```

We observe that when the same number of components are used, the percentage of variance explained in the predictors in PCR are larger or equal than that in PLS, but the percentage

of variance explained in the response in PCR are always smaller or equal than that in PLS. Briefly explain the reasons.

Solution. Let k the number of components used. PCR uses the first k score vectors as predictors which maximizes the proportion of variance explained in the predictors. However, PLS recursively uses the residuals from fitting the $k-1$ predictors in the previous step, which leads to maximizing the proportion of variance explained in the response at every step.

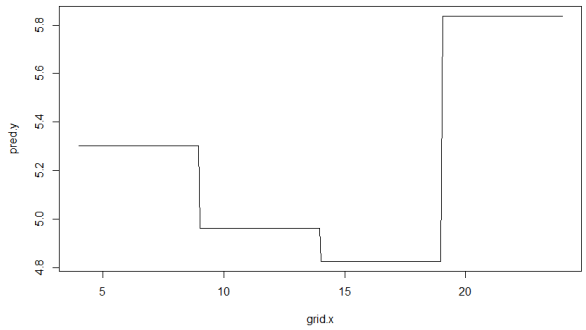
26. We fit a step function regression on a dataset with a single predictor X. 3 cutpoints c_1, c_2, c_3 in the range of X are selected. Construct 4 variables

$$C_0(X) = I(X < c_1),\ C_1(X) = I(c_1 \leq X < c_2),\ C_2(X) = I(c_2 \leq X < c_3), C_3(X) = I(c_3 \leq X)$$

where $I(\cdot)$ is an indicator funtion. Remark that the linear model using $C_0(X), \cdots, C_3(X)$ as predictors is:

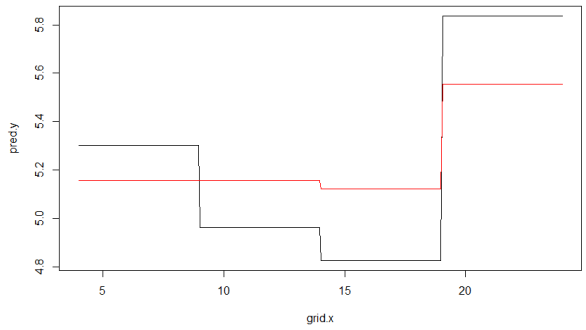
$$Y = \beta_0 + \beta_1 C_1(X) + \beta_2 C_2(X) + \beta_3 C_3(X) + \epsilon$$

The plot below displays the fitted step function on a dataset.



Now we fit a lasso regression using the same predictors. Using the tuning parameter $\lambda = 0.17$, we get 3 nonzero β_is out of 4; $\hat{\beta}_1 = 0$. Sketch the lasso-fitted step function on the plot above.

Solution.



27. Consider two curves, $\hat{g}_1$ and $\hat{g}_2$, defined by

$$\hat{g}_1 = \arg\min_g \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int [g^{(3)}(x)]^2 dx \right)$$

$$\hat{g}_2 = \arg\min_g \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int [g^{(4)}(x)]^2 dx \right)$$

where $g^{(m)}$ represents the mth derivative of g. For each of the following questions, either provide an answer or state that there is not enough information to answer.

(a) As $\lambda \to \infty$, which of $\hat{g}_1$ or $\hat{g}_2$ has the smaller training RSS?

(b) As we increase λ, how do the bias and the variance of $\hat{g}_1$ change?

(c) As $\lambda \to \infty$, which of $\hat{g}_1$ or $\hat{g}_2$ has the smaller test RSS?

Solution. When $\lambda = \infty$, $\hat{g}_1$ is the quadratic polynomial which minimizes the RSS, and $\hat{g}_2$ is the cubic polynomial which minimizes the RSS.

(a) $\hat{g}_2$ will have smaller training RSS.

(b) As we increase λ, the variance of $\hat{g}_1$ will decrease because the penalty for roughness becomes more significant, which prevents overfitting. On the other hand, the bias increases.

(c) There is not enough information to answer this.

28. We can estimate a natural cubic spline with K knots by fixing the locations of the knots at quantiles of X and finding the coefficients $\beta_0, \ldots, \beta_{K+1}$ which minimize the RSS. True or false: Smoothing splines are obtained in the same way by setting K to the number of samples n. Explain your answer.

Solution. This is false. Smoothing splines are natural cubic splines with knots at the data points, but the coefficients $\beta_0, \ldots, \beta_{n+1}$ are estimated differently, using an algorithm similar to Ridge regression, in order to obtain the function g which solves the problem

$$\text{minimize} \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int [g''(x)]^2 dx \right).$$

If we used the algorithm stated in the problem with $K = n$, we would place the knots at the data points, but we would estimate coefficients which minimize the RSS. It is easy to see that we could make the RSS exactly 0, which would clearly overfit.

29. A cubic spline with K knots is a function that:

- is a cubic polynomial between each pair of knots,
- is continuous at the knots, and
- has continuous first and second derivatives at each knot.

Without using the definition of a cubic spline in terms of basis functions, explain why the spline has $K + 4$ free parameters or degrees of freedom.

Solution. If there were no continuity constraints, we would have 4 parameters for each cubic piece, for a total of $4(K + 1)$ paremeters. However, we have 3 constraints for each knot, and each one eliminates one degree of freedom. This leaves us with $4K + 4 - 3K = K + 4$ parameters.

30. (Locally weighted regression) Often a linear regression is inadequate, and we turn to more flexible forms of regression. One such form is locally weighted regression. Consider the case of a single predictor x, and suppose the model is $y = f(x) + \epsilon$, with $\epsilon \sim N(0, \sigma^2)$. Local weighted regression approximates $f(x)$ at x_0 by $\hat{f}(x_0) = a_0 + b_0 x_0$, where a_0 and b_0 depend on x_0 and are chosen to minimize

$$\sum_{i=1}^{n} W\left(\frac{|x_i - x_0|}{\omega}\right)(y_i - a_0 - b_0 x_i)^2.$$

$W(r)$ is a positive weight function. Two options for weight functions are, the uniform weight function:

$$W_u(r) = \begin{cases} 1 & \text{if } |r| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

and the tri-cube weight function, invented by John Tukey:

$$W_{tc}(r) = \begin{cases} (1 - |r|^3)^3 & \text{if } |r| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

The parameter ω is known as the *window size* or bandwidth. Since x_0 is arbitrary, we have an approximation $\hat{f}$ at any value of x.

(a) Explain in words how this method works and why it makes sense.

(b) Explain the role of ω, and describe a principled way of selecting ω.

(c) What would be the main difference in the function $\hat{f}$ when we use the uniform weight function vs. the tri-cube weight function?

Solution.

(a) Local weighted regression is a way to estimate non-linear regression functions $f(x)$. The function at some test point x_0 is estimated by fitting a linear regression which only uses training points in the neighborhood of x_0. When $W(r)$ is the uniform weight function, local weighed regression is similar in spirit to nearest neighbors regression, except that instead of averaging the response of neighbors, we fit a linear model using only the neighbors. Unlike linear regression, this is a fully nonparametric procedure.

(b) ω controls the width of the neighborhood, since the weight for the training sample (x_i, y_i) is 0 when $|x_0 - x_i| > \omega$. It determines the number of neighbors of x_0 that influence the fit at x_0. When ω is small, we only consider points in a very small neighborhood of x_0, which leads to overfitting. We can select the optimal window size by k-fold cross validation.

(c) The uniform weight function produces a piecewise linear estimate $\hat{f}$, because the coefficients a_0 and b_0 are constant in a range of test points x_0 which have the same set of neighbors in a window of width ω. Note that this function could be discontinuous and non-smooth. On the other hand, the tri-cube weight function produces a continuous and smooth estimate $\hat{f}$, because the weight or influence of a training sample on the parameters a_0 and b_0 decrease to 0 gradually as the training sample moves to the edge of the window $[x_0 - \omega, x_0 + \omega]$.

31. Consider a natural cubic spline and a polynomial regression with the same degrees of freedom on the same data set. Which is likely to be more stable for extreme values of the predictor?

    Solution. The natural cubic spline.

32. List 2 regression methods for which it is possible to compute the leave one out cross validation (LOOCV) error analytically without performing n fits.

    Solution. Multiple linear regression and smoothing splines.

33. In class, we studied local linear regression, which solves a separate least squares problem at each target point x_0:

$$\min_{\alpha(x_0),\beta(x_0)} \sum_{i=1}^{N} K(x_0, x_i)[y_i - \alpha(x_0) - \beta(x_0)x_i]^2.$$

    The estimate is then $\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0)x_0$.

    Following this line, we can obtain local quadratic fits by solving the following least squares problems:

$$\min_{\alpha(x_0),\beta(x_0),j=1,\ldots,d} \sum_{i=1}^{N} K(x_0, x_i) \left[y_i - \alpha(x_0) - \beta_1(x_0)x_i - \beta_2(x_0)x_i^2\right]^2$$

    with solution $\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}_1(x_0)x_0 + \hat{\beta}_2(x_0)x_0^2$.

    Please list pros and cons of these two methods.

    Solution. Local quadratic fits can help reduce bias due to curvature in the interior of the domain. However, it increases the variance of the estimation, especially at the boundaries.

34. One-dimensional spline models can be extended to multidimensional inputs. Suppose we have two predictors X_1 and X_2, and the functions $h_k,\ k = 1, \ldots, M$ form a basis for a cubic spline with fixed knots $\xi_1, \ldots, \xi_{M-4}$. We can define two dimensional basis functions by the tensor product

$$g_{jk}(X) = h_j(X_1)h_k(X_2),\ j, k = 1, \ldots, M,$$

    and a two-dimensional function can be modeled as a linear combination of these:

$$g(X) = \sum_{j_1=1}^{M} \sum_{j_2=1}^{M} \theta_{jk} g_{jk}(X).$$

    How does this model compare in terms of bias and variance with a GAM which represents $g(X)$ as a sum of a cubic spline in X_1 and a cubic spline in X_2 with knots at $\xi_1, \ldots, \xi_{M-4}$ in each case?

Solution. The GAM uses $2M$ basis functions as opposed to M^2 in the multidimensional model, which does not constrain the effects for each input to be additive. Therefore, the GAM has higher bias but lower variance.

35. Imagine starting to grow a 2-class classification tree. We have 80 points total, with 3 possible values of x. In class 1, 30 of the points are at $x = 0$, 10 points are at $x = 1$, and no points are at $x = 2$. In class 2, 10 of the points are at $x = 0$, 10 points are at $x = 1$, and 20 points are at $x = 2$. There are two potential splits on x: split between $x = 0$ and $x = 1$, or split between $x = 1$ and $x = 2$.

    Compute the the misclassification error and the Gini index for the two splits. Which criterion produces a pure region?

    Solution. The misclassification error is .25 for either split. The Gini index is 3/8 for the split between 0 and 1 and 1/3 for the split between 1 and 2. The misclassification error doesn't discriminate between the two splits, and the Gini index tells us to split betwen 1 and 2, giving us one region that is completely pure.

36. Suppose you are fitting a regression tree and have a categorical variable with K classes. When considering splits on this variable, do you have to consider each of the $2^K - 2$ possible splits? Explain.

    Solution. No, you can compute the average response for each category in the region being considered, order the categories by their average response, and treat the categorical variable as an ordinal variable.

37. Describe the procedure for computing the out of bag error from Bagging a regression tree.

    Solution. For each sample x_i, consider all the Bootstrap samples that do not contain x_i and average the predictions for the response y_i made by the corresponding trees; call the average $\hat{y}_i^{oob}$. Then, the out of bag error is given by:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{oob})^2.$$

38. How does Bagging a regression tree differ from a Random Forest? Why might Random Forests be preferable?

    Solution. In each case, we average the result of decision trees fit to many Bootstrap samples. However, in a Random Forest we restrict the number of variables to consider in each split. This produces a greater diversity of decision trees or *weak learners*, which reduces the variance of the averaged prediction.

39. We build a classification tree using the predictors $X_1, X_2, \ldots, X_p$. We build a second tree using the predictors $f(X_1), f(X_2), \ldots, f(X_p)$, where f is monotone:

$$f(x) \geq f(y) \text{ if and only if } x \geq y.$$

    Prove that the two trees produce the same partition of the training data.

    Solution. Every time we make a split in a decision tree, all we need to know is how the training samples are ordered in each predictor. These orderings are unaltered by the monotone transformation.

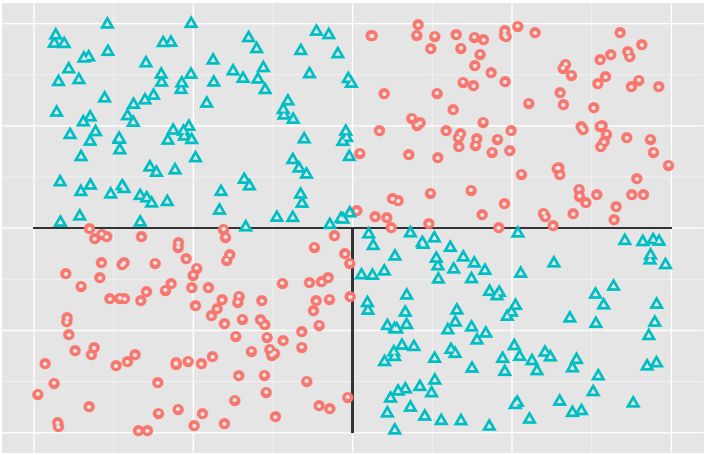
40. The standard method for fitting a decision tree involves:

    - Growing the tree split by split. We maximize the reduction of the training error at each step until there are at most 5 samples per region.
    - Pruning the tree to obtain a sequence of trees of decreasing size.
    - Selecting the optimal size by cross-validation.

    Consider the following alternative approach. Grow the tree split by split until the reduction in the training error produced by the next split is smaller than some threshold. This approach may lead to bad results because it is possible to make a split which does not decrease the error by much, and then make a second split which reduces the error significantly.

    Draw an example dataset where this happens with 2 predictors X_1 and X_2, and a binary categorical response.

    Solution.



    The figure shows the partition produced by a tree with 2 splits. The first split (horizontal line) barely reduces the classification error. However, the second split (vertical line) decreases the error significantly.

41. Describe what the following code does. The `shrinkage` parameter in `gbm` controls what was called λ or the learning rate in class, `n.trees` controls the number of iterations to perform (called B in class), and `interaction.depth` controls the depth of the trees used (called d in class).

```
library(gbm)
library(MASS)
data(Boston)
lambda.grid = seq(.001, .2, length.out = 100)
test.error = rep(0, length(lambda.grid))
train.idx = sample(nrow(Boston), round(nrow(Boston) * .7))
for(i in 1:length(lambda.grid)) {
   lambda = lambda.grid[i]
```

```
    boost = gbm(medv ~ ., data = Boston[train.idx,],
                distribution = "gaussian", n.trees = 5000,
                interaction.depth = 1, shrinkage = lambda)
    yhat.boost = predict(boost, newdata = Boston[-train.idx,],
                         n.trees = 5000)
    test.error[i] = mean((yhat.boost - Boston$medv[-train.idx])^2)
}
```

Solution. This is using a test set to find the test error for values of λ between .001 and .2.

42. When we apply Bagging to decision trees, or when we construct Random Forests, each tree fit to a different bootstrap replicate of the data can be grown "deep", i.e. to a level where there are very few training samples per leaf. The pruning step is skipped. Explain why this doesn't lead to overfitting.

    Solution. Growing a tree "deep" increases the variance of the prediction, or the proclivity to overfit. However, averaging predictions produced from many bootstrap replicates of the data reduces the variance. One effect can compensate for the other.

43. We apply Bagging to a 1-nearest neighbor regression problem. To make a prediction at x_0, we average the predictions of 1-nearest neighbor regression fit to M Bootstrap replicates of the training data. Prove that the bagging prediction for M large is equivalent to a weighted average of k-nearest neighbors regression predictions with different values of k.

    Hint: Suppose we take n samples with replacement from the set $\{1, 2, \ldots, n\}$. Let Z be the smallest of the samples. Then,

$$\mathbb{P}(Z = i) = \left(1 - \frac{i-1}{n}\right)^n - \left(1 - \frac{i}{n}\right)^n,$$

    which is monotone decreasing in i.

    Solution. $\mathbb{P}(Z = i)$ is the probability that the nearest neighbor to x_0 in a Bootstrap sample is the ith nearest neighbor in the full training sample. Let $y_1, y_2, \ldots, y_n$ be the observed outputs ordered according to the proximity to the test input to x_0. The Bagging prediction in the limit $M \to \infty$ can be written

$$\hat{y}_0 = \sum_{i=1}^{n} y_i \mathbb{P}(Z = i).$$

    Let

$$\hat{y}_0^{(k)} = \frac{y_1 + y_2 + \cdots + y_k}{k}$$

    be the k-nearest neighbors prediction. Defining $W_n = nP(Z = n)$ and $W_i = i(\mathbb{P}(Z = i) - \mathbb{P}(Z = i + 1))$ for $i < n$, we have

$$\hat{y}_0 = \sum_{i=1}^{n} W_i \hat{y}_0^{(k)}.$$

44. Explain why the maximal margin classifier cannot be applied in the case of non-separable data.

    Solution. If the data are not separable, that means there exists no separating hyperplane. In this case, for any hyperplane, there is no margin between the hyperplane and the data.

45. Consider the maximal margin classifier:

$$\begin{array}{ll} \text{maximize} & M \\ \text{subject to} & \sum_{j=1}^{p} \beta_j^2 = 1 \\ & y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip}) \geq M \ \forall \ i = 1, ..., n \end{array}$$

and the support vector classifier:

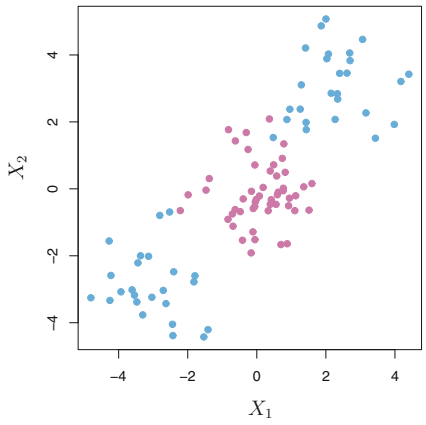
$$\begin{array}{ll} \text{maximize} & M \\ \text{subject to} & \sum_{j=1}^{p} \beta_j^2 = 1 \\ & y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip}) \geq M(1 - \epsilon_i) \ \forall \ i = 1, ..., n \\ & \epsilon_i \geq 0, \ \ \sum_{i=1}^{n} \epsilon_i \leq C \end{array}$$

Note that the support vector classifier introduces a tuning parameter C, and if C is chosen to be zero, then the support vector classifier is equivalent to the maximal margin classifier.

Suppose that a data set is separable, so that the maximal margin classifier is applicable. Describe the relationship between the choice of C and the robustness of the support vector classifier to individual observations.

Solution. The maximal margin classifier (or, equivalently, the support vector classifier with $C = 0$) must fully separate the data: Each individual observation must be on the correct side of the hyperplane. Thus, the maximal margin classifier is not robust to individual observations. If $C > 0$, however, then the classifier is allowed some slack to ignore individual observations for the sake of finding a greater margin between all other observations. The greater C is, the greater the slack is. Hence, as C increases, so does robustness of the support vector classifier to individual observations.

46. The figure below, taken from *Introduction to Statistical Learning*, depicts a data set of two classes, represented by the colors blue and purple. Explain why the support vector machine would be much preferable to the support vector classifier in this instance.



Solution. The data are clearly well separated in this case, but not in a linear manner. Support vector machines are the nonlinear extension of the support vector classifier and thus would perform much better in this case.

47. Describe two methods for extending support vector machines from two classes to an arbitrary number (say, K) of classes.

Solution. One method is one-versus-one classification. If there are K classes, then there are $\binom{K}{2}$ pairs of classes, and one could fit a support vector machine for each pair of classes, garnering a $\binom{K}{2}$ classifications for each point. Then that point could be classified according to which class it was most frequently classified in the $\binom{K}{2}$ pairwise classifications. Another method is one-versus-all classification. One could fit K support vector machines, each time comparing one class against all other classes, yielding a classifier of the form

$$\delta_k(x^*) = \beta_{0k} + \beta_{1k}x_1^* + \beta_{2k}x_2^* + ... + \beta_{pk}x_p^* \text{ for } k = 1, ..., K$$

Any data point x could then be classified according to which $\delta_k(x^*)$ was largest.

48. Recall an equivalent formulation of the support vector **classifier**:

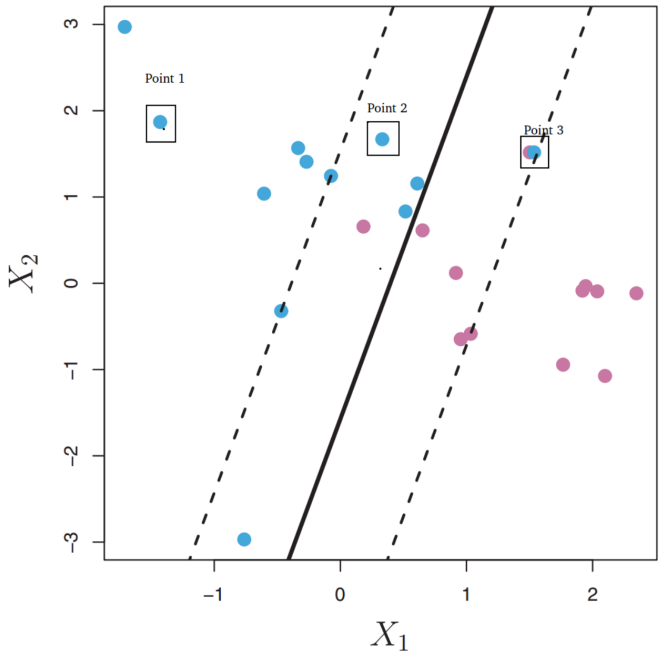
$$\text{minimize} \left\{ \sum_{i=1}^{n} \max[0, 1 - y_i(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip})] + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

where λ is some tuning parameter. Describe the relationship between λ and the tuning parameter C from the definition of the support vector classifier above. What does this relationship imply about the choice of C as it pertains to bias-variance tradeoff?

Solution. Small λ corresponds to small C, and large λ corresponds to large C. Thus C can be thought of as a regularization parameter: Hi-C yields relatively low variance and high bias while low-C yields relatively low bias and high variance.

49. The plot below, from ISLR, shows the separating hyperplane (solid line) and margin (dotted lines) resulting from fitting the SVM optimization

$$\begin{aligned}
&\text{maximize } M && (1)\\
&\text{over } \beta, \epsilon \text{ such that,} && (2)\\
&\beta_0 + \beta_1^2 + \beta_2^2 = 1 && (3)\\
&y_i \left(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}\right) \geq M \left(1 - \epsilon_i\right) && (4)\\
&\epsilon_i \geq 0 && (5)\\
&\sum_{i=1}^{n} \epsilon_i \leq C. && (6)
\end{aligned}$$



This problem is about the interpretation of these parameters.

(a) Suppose we decrease C. What happens to the separating hyperplane? What happens to the margin?

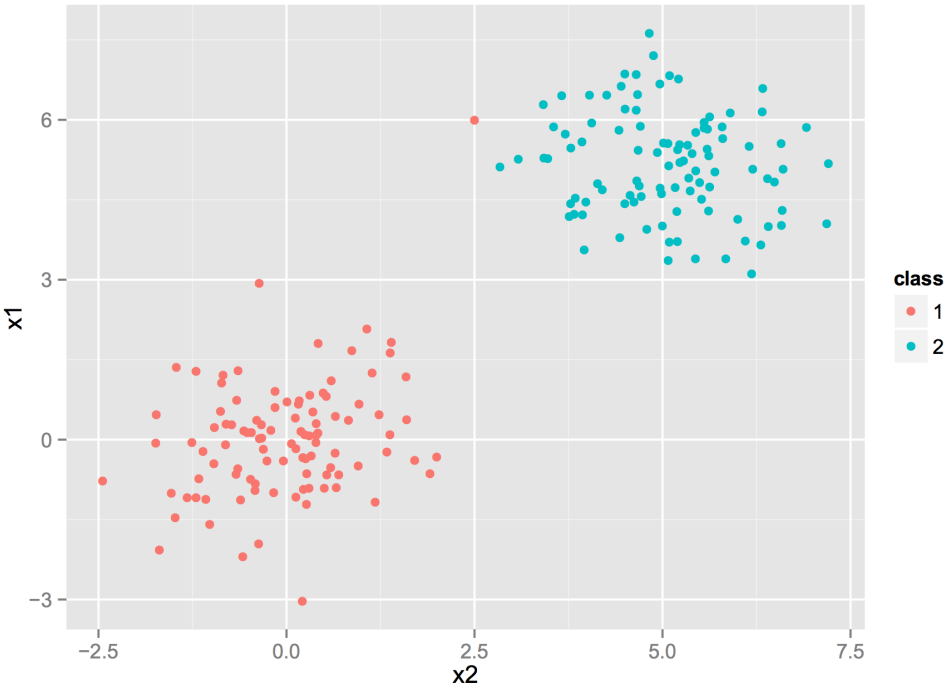
(b) For each of the boxed blue points, specify whether $\epsilon_i = 0, \epsilon_i \in (0, 1]$, or $\epsilon_i > 1$.

Solution.

(a) The separating hyperplane doesn't change very much, since the current SVM already has many support vectors. The margin gets narrower.

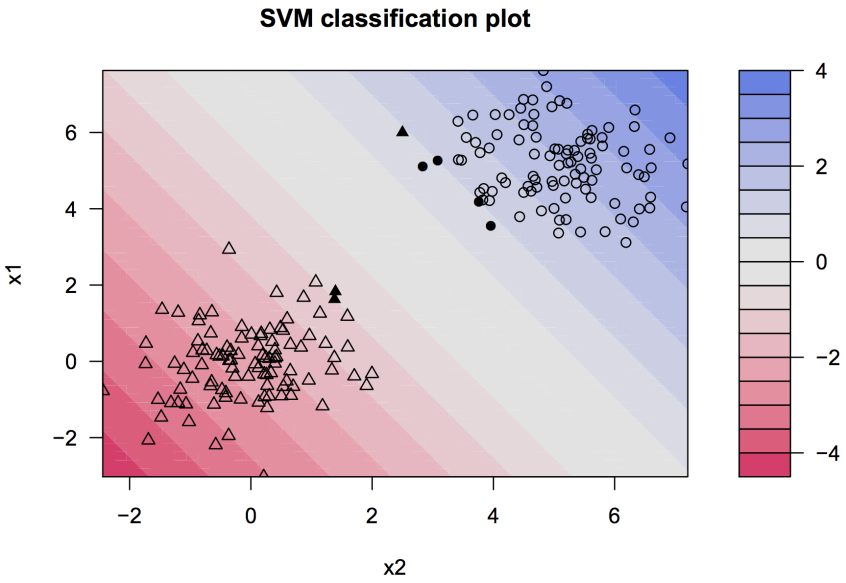
(b) (1) is correctly classified, so $\epsilon_i = 0$. (2) is correctly classified, but within the margin, so $\epsilon_i \in (0, 1]$. (3) is incorrectly classified, so $\epsilon_i > 1$.

50. You have collected a data set with two clearly distinguishable classes, but with one outlier, as plotted below.



You decide to build an linear-kernel SVM to distinguish between these two classes. You tinker with different parameters, and plot the resulting separating hyperplanes. For this problem, consider the following two fits, resulting from model 1,

and model 2,



The gradient indicates the distance from the hyperplane, which is positive in the direction of the circles and negative in the direction of the triangles. The filled points are the support vectors.

(a) You forgot which model has a larger value of C, the constraint on the sum of the slack variables in the usual SVM optimization.

$$\begin{aligned} &\text{maximize } M \\ &\text{over } \beta, \epsilon \text{ such that} \\ \beta_0 + \beta_1^2 + \beta_2^2 &= 1 \\ y_i \left(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}\right) &\geq M \left(1 - \epsilon_i\right) \\ \epsilon_i &\geq 0 \\ \sum_{i=1}^{n} \epsilon_i &\leq C. \end{aligned}$$

Can you deduce this from the given plots?

(b) Which should you use?

(c) In general, state if increasing the parameter C increases or decreases the following quantities: size of margin, number of discovered support vectors, variance of the fitted model.

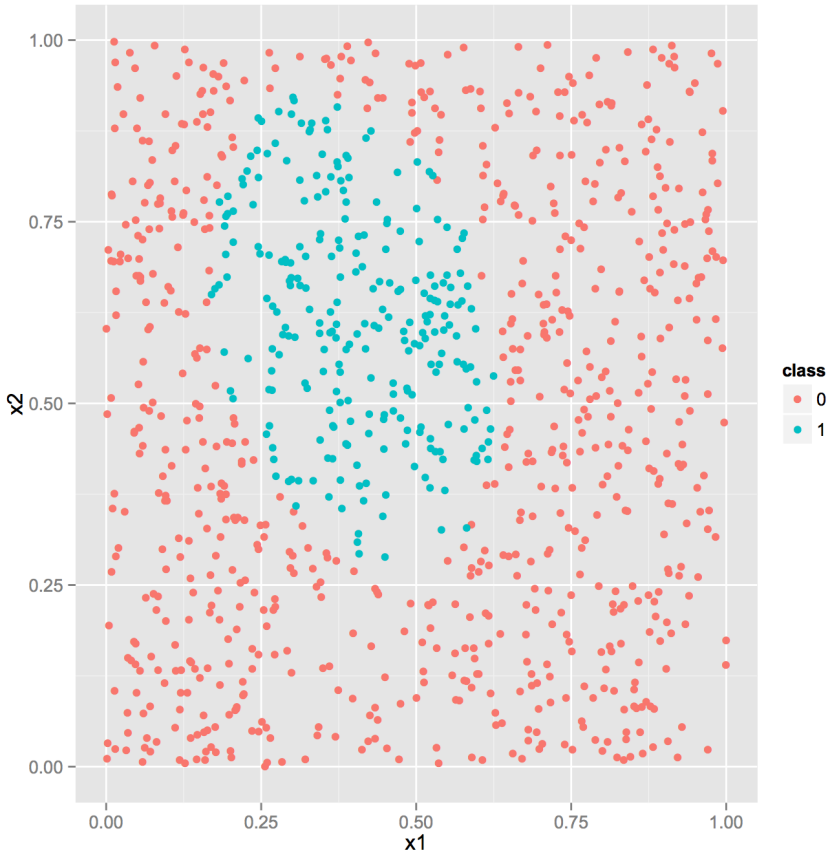
Solution.

(a) We can tell model 1 had a smaller value of C, since it tries to enforce perfect separation even when a much larger margin fit can be obtained by ignoring the one outlier. This model also has fewer support vectors.

(b) Model 2 would have much better generalization error, since it does not overfit to an outlier.

(c) Increasing C increases the margin, increases the number of support vectors, and decreases the variance of the fitted model.

51. You have a data set where one class falls in the interior of an ellipse, and the other falls outside, as in the plot below.



You would like to build an SVM to distinguish between the two classes.

(a) Recall the linear kernel,

$$K\left(x_i, x_{i'}\right) = \sum_{j=1}^{2} x_{ij} x_{i'j}, \tag{7}$$

the $d-$degree polynomial kernel,

$$K\left(x_i, x_{i'}\right) = \left(1 + \sum_{j=1}^{2} x_{ij} x_{i'j}\right)^{d}, \tag{8}$$

and the radial kernel,

$$K\left(x_i, x_{i'}\right) = \exp\left(-\gamma \sum_{j=1}^{2} \left(x_{ij} - x_{i'j}\right)^2\right). \tag{9}$$

Which of these kernels is most appropriate for your problem?

Hint: In two dimensions, an ellipse centered at $x_0 = \begin{pmatrix} x_{01} \\ x_{02} \end{pmatrix}$ with rotation $K = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}$ is the set of vectors $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ satisfying

$$(x_1 - x_{01})^2 K_{11} + (x_2 - x_{01})^2 K_{22} + 2 (x_2 - x_{02}) (x_1 - x_{01}) \leq C, \tag{10}$$

for some C.

(b) For each of the polynomial and radial kernels above, explain the impact of increasing d and γ, respectively, on the bias and variance of the fit.

Solution.

(a) From the hint, we see the decision boundary can be written in terms of a quadratic in x_1, x_2. Hence, the polynomial kernel of order 2 would be most appropriate.

(b) Increasing d reduces bias but increases variances, since we can fit more complicated decision boundaries, at the expense of involving more features (we begin to suffer from the curse of dimensionality). Increasing γ similarly reduces bias and increases variance.