CPH 636 — Spring 2009 — Dr. Charnigo

Lecture 8

Regression trees

Scenario and motivation for alternative methodologies. We have a continuous response variable Y and several explanatory variables $X_1, \ldots, X_k$. The explanatory variables may be any mix of continuous and dichotomous.

A common methodological choice in this scenario, for both conventional analysis and data mining, is linear regression modeling. Recall (from Lecture 4) that a linear regression model describes the expected response as a linear combination of explanatory variable values,

$$E[Y|x_1, \ldots, x_k] = \alpha + \beta_1 x_1 + \cdots + \beta_k x_k, \tag{1}$$

and prescribes that the differences between actual responses and expected responses must be independent normal random variables with unknown but common variance σ^2,

$$Y_i = E[Y|x_{1,i}, \ldots, x_{k,i}] + \epsilon_i = \alpha + \beta_1 x_{1,i} + \cdots + \beta_k x_{k,i} + \epsilon_i. \tag{2}$$

When fitting a linear regression model, many people use diagnostic plots to check whether the ϵ_i in (2) are normally distributed with unknown but common variance σ^2. Yet, many of these same people take for granted that (1) is reasonable. Unfortunately, there are situations in which (1) is not reasonable and in which corrective action is difficult.

A potential problem with (1) is that we may really have

$$E[Y|x_1, \ldots, x_k] = \alpha + g_1(x_1) + \cdots + g_k(x_k), \tag{3}$$

where $g_1(x_1)$ through $g_k(x_k)$ are smooth but not linear functions. Only if the approximations $g_1(x_1) \approx \beta_1 x_1, \ldots, g_k(x_k) \approx \beta_k x_k$ are good will (1) be reasonable. We can consider more sophisticated approximations, such as $g_1(x_1) \approx \beta_1 x_1 + \gamma_1 x_1^2, \ldots, g_k(x_k) \approx \beta_k x_k + \gamma_k x_k^2$, which will yield a version of (1) with quadratic terms,

$$E[Y|x_1, \ldots, x_k] = \alpha + \beta_1 x_1 + \cdots + \beta_k x_k + \gamma_1 x_1^2 + \cdots + \gamma_k x_k^2. \qquad (4)$$

However, even (4) may be unreasonable. [[The approximation $g_1(x_1) \approx \beta_1 x_1 + \gamma_1 x_1^2$ may be good if $g_1(x_1)$ is either convex or concave; otherwise, the approximation will be poor.]]

Another potential problem with (1) is that it does not allow for interactions among the explanatory variables, in the sense that the effect of changing x_1 does not depend on the (fixed) values $x_2, \ldots, x_k$. In particular, increasing x_1 by one unit increases the expected response by β_1 units regardless of $x_2, \ldots, x_k$. [[Actually, (3) does not allow for interactions among the explanatory variables either, as the effect of changing x_1 still does not depend on $x_2, \ldots, x_k$. In particular, increasing x_1 from c to $c+1$ increases the expected response by roughly $\frac{d}{dx_1} g_1(x_1)|_{x_1=c}$ units regardless of $x_2, \ldots, x_k$.]]

If there are interactions, we can try to incorporate them by adding product terms to (1). For instance, an interaction involving the first two explanatory variables can be captured by a term of the form $\beta_{1,2} x_1 x_2$. Yet, we face the difficulty that with k explanatory variables there are $k(k-1)/2$ possible interaction terms. [[Here I am considering only two-way interactions. The situation becomes worse if I allow three-way interactions.]]

Discussion question. Suppose that I am willing to fit a model that includes all $k(k-1)/2$ possible interaction terms, with the intention of removing any interaction terms that have non-significant coefficient estimates. What are the potential problems with this approach?

Alternative methodologies. In Lectures 8, 9, and 10 we will discuss alternative methodologies for the present scenario when (1) is unreasonable and not easily repaired or (1) is possibly reasonable but we want to look at the data in a second way to gain a complementary perspective. The alternative methodology to be discussed in Lecture 8 is called a regression tree. The alternative methodologies to be discussed in Lectures 9 and 10 are called a neural network and a nearest neighbors analysis, respectively.

A regression tree. The main idea of a regression tree is that we play, as it were, a statistician's version of the "20 questions" parlor game. I can most easily elaborate by way of an example, so let us refer to {FEVEx.mdi}. This file displays a regression tree that has been fit to the training subset of the forced expiratory volume (FEV) data from Lecture 4.

Suppose that we want to make a prediction for the FEV of a 10 year old girl who is 5 feet 2 inches tall and does not smoke.

Question #1: "Is the person's height less than 61.75 inches?" The answer is "No", so we proceed from the top node of the tree (marked "Average: 2.61") along the right branch to the node marked "Average: 3.31".

Question #2: "Is the person's height less than 66.25 inches?" The answer is "Yes", so we proceed from the node marked "Average: 3.31" along the left branch to the node marked "Average: 2.94".

Question #3: "Is the person's age less than 10.5 years?" The answer is "Yes", so we proceed from the node marked "Average: 2.94" along the left branch to the node marked "Average: 2.77".

We now make a prediction of 2.77 for the FEV, as asking more questions would not help us to make a better prediction.

Discussion question. What do we predict for the FEV of a 16 year old boy who is 5 feet 9 inches tall and does not smoke?

How a regression tree is constructed. Let $\tilde{y}_{i,j}$ denote the "prediction" that we would make for subject i in the training data set after asking j "Yes"/"No" questions.

When $j = 0$ (i.e., before we have asked any "Yes"/"No" questions), the most reasonable course of action is to set $\tilde{y}_{i,0} := \bar{y}$ for every subject in the training data set since there is not yet any basis for distinguishing any subject from the others. In our example above, $\bar{y} = 2.61$.

When $j = 1$ (i.e., after we have asked just one "Yes"/"No" question), the most reasonable course of action is to set $\tilde{y}_{i,1} := \bar{y}_{YES}$ if subject i would answer "Yes" and $\tilde{y}_{i,1} := \bar{y}_{NO}$ if subject i would answer "No", where $\bar{y}_{YES}$ denotes the average response in the training data set among people who would answer "Yes" and $\bar{y}_{NO}$ denotes the average response in the training data set among people who would answer "No". In our example above, $\bar{y}_{YES} = 2.03$ and $\bar{y}_{NO} = 3.31$.

Why do we ask our first question about height rather than (say) age, and why do we use a cutoff of 61.75 instead of (say) 72? The reason is that asking the first question about height and using a cutoff of 61.75 results in the smallest possible value of

$$n^{-1} \sum_{i=1}^{n} (y_i - \tilde{y}_{i,1})^2,$$

the average squared error on the training data after one question.

When $j = 2$ (i.e., after we have asked two "Yes"/"No" questions), the most reasonable course of action is to set $\tilde{y}_{i,2}$ to one of $\bar{y}_{YES,YES}$, $\bar{y}_{YES,NO}$, $\bar{y}_{NO,YES}$, or $\bar{y}_{NO,NO}$ based on whether subject i would answer "Yes, Yes", "Yes, No", "No, Yes", or "No, No" to the two questions. Here $\bar{y}_{YES,YES}$

denotes the average response in the training data set among people who would answer "Yes, Yes", and the other averages are defined analogously. In our example above, $\bar{y}_{YES,YES} = 1.75$, $\bar{y}_{YES,NO} = 2.50$, $\bar{y}_{NO,YES} = 2.94$, and $\bar{y}_{NO,NO} = 3.92$.

What determines the second question? The second question is chosen so that we can have the smallest possible value of

$$n^{-1} \sum_{i=1}^{n} (y_i - \tilde{y}_{i,2})^2,$$

the average squared error on the training data after two questions. Also, note that what the second question is depends on the answer to the first question.

This process continues until we deplete the training data set, in the sense that node-specific averages like 1.35 and 2.55 are based on fewer than (say) ten subjects, or until we have reached a pre-specified quota for the number of questions that we are willing to ask. Incidentally, the software default is actually six questions, not 20.

Discussion question. Suppose that one subject has an extreme outying value on X_1. Do you think that this will be more of an issue for a linear regression model or for a regression tree?

Pruning a regression tree. We made a prediction of 2.77 for the FEV of a 10 year old girl who is 5 feet 2 inches tall and does not smoke, stating that no more questions would be asked since the answers that we would obtain would not help us to make a better prediction. Yet, the $\bar{y}_{NO,YES,YES}$ of 2.77 was based on 36 subjects, so we certainly could have asked another question or two without the training data set being depleted. Why didn't we?

Just as we don't want to have too many explanatory variables in a linear regression model, we don't want to have too many nodes or "leaves" in a regression tree. Part of the reason for this is that a simpler regression tree is easier to interpret. But the main issue is that an overly complex regression tree, like an overly complex linear regression model, will adapt to the idiosyncrasies of the training data. A regression tree that has adapted to the idiosyncrasies of the training data will have difficulty making predictions for people in other data sets.

Thus, if we have a validation data set, we "prune" the regression tree to whatever intermediate version of the tree yields the smallest average squared error on the validation data. The pruned tree is the one that is actually shown in {FEVEx.mdi}.

If we do not have a validation data set, then using an AIC-type criterion to prune the regression tree may be reasonable. Unfortunately, such an option is not provided in the Enterprise Miner software, which is the software that we will use to fit regression (and classification) trees since Fernandez has not written a macro for that purpose.

Further output from Enterprise Miner. The file {em_report.html} under the FEV heading yields further information about the regression tree.

The "Model assessment plot" shows the average squared error on the training data and the average squared error on the validation data for all intermediate versions of the tree, which are indexed by the number of leaves. The average squared error on the training data continually declines, in the same way that the residual sum of squares continually declines as we add explanatory variables to a linear regression model. Yet, most of the reduction in average squared error attained at 35 leaves (the most complex regression tree considered) has already been attained at 11 leaves. More importantly, the average squared error on the validation data is minimized at 21 leaves. Thus, the pruned tree shown in {FEVEx.mdi} has 21 leaves.

Immediately below the “Model assessment plot”, we see the average squared error on the training data, the average squared error on the validation data, and the average squared error on the test data for the pruned tree with 21 leaves. The 0.115 figure quoted for the training data must be regarded as optimistic in view of the 0.181 and 0.175 figures quoted for the validation and test data. To interpret the 0.175, for instance, we can take its square root. The square root is 0.418, which indicates the typical magnitude of the errors in prediction made by the pruned tree with 21 leaves.

Proceeding further, we find a link to “English rules”. Clicking the link, we see a collection of prediction rules expressed logically. The first rule is to predict an FEV of 2.29 for anyone less than 8.5 years old with height between 58.75 and 61.75 inches. The 2.29 was the sample mean in the training data set among the 14 subjects less than 8.5 years old with height between 58.75 and 61.75 inches. The sample standard deviation among these 14 subjects was 0.27.

Continuing, we find a link to “Datastep score code”. Clicking the link, we see dozens of lines of SAS code that could be placed between statements like “DATA RCLIB.FEVTEST; SET RCLIB.FEVTEST;” and “RUN;” to add a new variable “P_FEV” to any data set with the same explanatory variables as the training data set. The new variable P_FEV would contain predicted FEV values for all of the subjects in the data set. Note that these lines of SAS code contain several instances of “IF NOT MISSING(HGT)” and “IF NOT MISSING(AGE)”, suggesting that predictions will be made even for people who have missing values on height and age.

Handling missing values. Suppose that we want to make a prediction for someone who has a missing value on one of the explanatory variables. This is generally not a problem with a regression tree.

First, the explanatory variable on which the person has a missing value may not be encountered as one traverses the branches of the tree determined by the person's answers to the "Yes" / "No" questions. In our example above, we never used gender or smoking status to obtain the predicted FEV of 2.77 for a 10 year old girl who is 5 feet 2 inches tall and does not smoke. So, if smoking status had been missing, there would have been no difficulty.

Second, even if the explanatory variable on which the person has a missing value comes into play, we can work around the missingness using a "surrogate split". A surrogate split is a backup question that we have ready in case the original question cannot be answered. For instance, suppose that we want to make a prediction for someone of known age but unknown height. The original question "Is the person's height less than 61.75 inches?" can be replaced by a backup question that does not require knowledge of height, such as "Is the person's age less than 9.5 years?" Of course, the backup question should be such that most people who answer "Yes" to the backup question would also answer "Yes" to the original question.

The surrogate split mechanism is rather specific to regression (and classification) trees. In other settings, such as linear (and logistic) regression modeling, one can handle missing values on explanatory variables using some form of "imputation". This entails making an intelligent guess as to what each missing value should have been, possibly based on the non-missing values for other explanatory variables with which the affected explanatory variable is correlated. [[See Lecture 5 of CPH 931 from Fall 2008.]]

Classification trees

Scenario and motivation for alternative methodologies. We have a dichotomous response variable Y and several explanatory variables $X_1, \ldots, X_k$. The explanatory variables may be any mix of continuous and dichotomous.

A common methodological choice in this scenario, for both conventional analysis and data mining, is logistic regression modeling. Recall (from Lecture 6) that a logistic regression model describes the logit risk as a linear combination of explanatory variable values,

$$\log\left(\frac{p_{x_1,\ldots,x_k}}{1 - p_{x_1,\ldots,x_k}}\right) = \alpha + \beta_1 x_1 + \cdots + \beta_k x_k. \qquad (5)$$

However, we may be reluctant to assume (5) if we think that the relationships between explanatory variables and the logit risk may not be linear or that there may be interactions among the explanatory variables.

In Lectures 8, 9, and 10 we will discuss alternative methodologies for the present scenario when (5) is unreasonable and not easily repaired or (5) is possibly reasonable but we want to look at the data in a second way to gain a complementary perspective. The alternative methodology to be discussed in Lecture 8 is called a classification tree, which is conceptually similar to a regression tree. A classification tree has the further advantage that it can accommodate a categorical response with more than two categories, although we will take Y to be dichotomous for the present example.

A classification tree. Refer to {SAEx.mdi}, which displays a classification tree that has been fit to the training subset of the South African heart disease data from Lecture 6.

Suppose that we want to make a prediction about coronary heart disease for a non-smoker with a family history of coronary heart disease who has an adiposity score of 30 and a low density lipoprotein score of 10.

Question #1: “Does the person consume fewer than 7.38 units of tobacco?” The answer is “Yes”, so we proceed from the top node of the tree (marked “1: 37%”) along the left branch to the node marked “1: 29%”.

Question #2: “Does the person have a family history of heart disease?” The answer is “Yes”, so we proceed from the node marked “1: 29%” along the left branch to the node marked “1: 46%”.

Question #3: “Does the person have an adiposity score less than 34.575?” The answer is “Yes”, so we proceed from the node marked “1: 46%” along the left branch to the node marked “1: 38%”.

Question #4: “Does the person have a low density lipoprotein score less than 8.36?” The answer is “No”, so we proceed from the node marked “1: 38%” along the right branch to the node marked “1: 100%”.

Formally, the classification tree estimates the risk to be 100%. This is because all five people in the training data set who answered these four questions in the same way developed coronary heart disease. Of course, five is a very small number to look at, especially for a dichotomous response, so the risk estimate of 100% is hardly definitive. Be that as it may, since 100% is greater than 50%, we predict that the person will develop coronary heart disease.

How a classification tree is constructed. Just as leaves were added to a regression tree to minimize average squared error, leaves are added to a classification tree to minimize a quantity called the Gini index. We stop adding leaves when the training data set is depleted or when we reach a pre-specified quota for the number of questions that we are willing to ask.

[[With a dichotomous response variable, the Gini index is $\sum_j n_j \widehat{p}_j(1-\widehat{p}_j)$, where the summation is over nodes of the classification tree, n_j is the number of training data subjects in node j, and $\widehat{p}_j$ is the fraction of these who have $Y = 1$. Ignoring the possibility of overfitting the training data, which can be addressed by pruning (see below), can you explain why minimizing the Gini index is desirable?]]

Pruning. If we have a validation data set, we "prune" the classification tree to whatever intermediate version of the tree yields the smallest misclassification rate on the validation data. The pruned tree is the one that is actually shown in {SAEx.mdi}.

Further output from Enterprise Miner. The file {em_report.html} under the SA heading yields further information about the classification tree.

The "Model assessment plot" shows the misclassification rate on the training data and the misclassification rate on the validation data for all intermediate versions of the tree, which are indexed by the number of leaves. The misclassification rate on the training data continually declines, while the misclassification rate on the validation data is minimized at 7 leaves. Thus, the pruned tree shown in {SAEx.mdi} has 7 leaves.

Immediately below the "Model assessment plot", we see the misclassification rate on the training data, the misclassification rate on the validation data, and the misclassification rate on the test data for the pruned tree with 7 leaves. The 0.195 figure quoted for the training data must be regarded as optimistic in view of the 0.261 and 0.336 figures quoted for the validation and test data.

Proceeding further, we find links to "English rules" and "Datastep score code". Clicking the former link reveals a collection of prediction rules expressed logically; clicking the latter link reveals SAS code that could be used to make predictions for individuals in a data set with the same explanatory variables as the training data set.

Continuing, we find a link to “Matrix”. Clicking the link, we see that the classification tree predicts no coronary heart disease for 126 of the 145 subjects in the training data set who did not have coronary heart disease and predicts coronary heart disease for 60 of the 86 subjects who did have coronary heart disease. The classification tree also predicts no coronary heart disease for 69 of the 77 subjects in the validation data set who did not have coronary heart disease while predicting coronary heart disease for 22 of the 38 subjects who did have coronary heart disease. The tree’s predictions are based on a cutoff of 50% for the estimated risk.

See {http://www.richardcharnigo.net/CPH636S09/SASEnterInstr.txt}, item 13, for code that can be used to generate predictions based on other cutoffs for the estimated risk and to generate predictions for people in the test data set or any other data set with the same explanatory variables as the training data set.

Discussion question. Which fractions two paragraphs above represent sensitivities and which represent specificities?