Homework 3: Numeric Predictions

Abraao Lourenco
aln@WPI.EDU
BS-MS

October 1st, 2004

Contents

Chapter 3

Homework 3: Numeric Predictions

3.1 Linear Regression

3.1.1 Regression Equation

The linear regression would express mpg as a linear combination of the attributes, with predetermined weights: $mpg = w_0 + w_1 \cdot cylinders + w_2 \cdot horsepower + w_3 \cdot weight + w_4 \cdot acceleration + w_5 \cdot model - year + w_6 \cdot car - name...$. The weights w are calculated from the training data.

3.1.2 Linear Regression Description and Weka Output

Linear Regression Description

The following is from pages 112 and 113 of the book. Linear regression is a prediction technique used when the class and all attributes are numeric. In linear regression the class is expressed as a liner combination of the attributes, each of which has a specific weight:

$$class = w_0 + w_1 \cdot a_1 + ... + w_n \cdot a_n$$

The weights are calculated from the training data as follows. Suppose the first instance has a class $x^{(1)}$ where the superscript (1) indicates that it is the first instance. The predicted class value for the first instance is expressed as the following formula:

$$class = w_0 \cdot a_0^{(1)} + w_1 \cdot a_1^{(1)} + ... + w_k \cdot a_k^{(1)} = \sum_{j=0}^{k} w_j \cdot a_j^{(1)}$$

The goal in linear regression is to choose the weights that will minimize the sum of the squares of the difference between the predicted class value above and the actual class values in the dataset. Given n training instances, with the ith instance denoted by the superscript (i), the sum of squares of the differences is as follows:

$$\sum_{i=0}^{n} \left(x^{(i)} - \sum_{j=0}^{k} w_j \cdot a_j^{(i)} \right)^2$$

Weka Output

Below is the linear regression equation output by Weka running after running linear regression over this dataset:

```
Linear Regression Model

mpg =

      2.4728 * cylinders +
      0.0896 * horsepower +
     -0.0101 * weight +
      0.9169 * acceleration +
      1.1738 * model-year +
      4.5119 * car-name=chevrolet,toyota,volkswagen +
      3.2888 * car-name=toyota,volkswagen +
    -76.9352
```

3.2 Regression Trees and Model Trees

3.2.1 Translating car-name

First we take the average of the class values associated with each of the vendor values: chevrolet, toyota, volkswagen, ford. This process is shown in table 3.1.

Table 3.1: Translation car-name into numeric attributes

Value	Average
chevrolet	$\frac{(18+27+34)}{3} = 26.333$
toyota	$\frac{(24+28+32)}{3} = 28$
volkswagen	$\frac{(26+29+44)}{3} = 33$
ford	$\frac{(10+28+16)}{3} = 18$

Table 3.2: Sorted attributes

Value	Average
volkswagen	33
toyota	28
chevrolet	26.333
ford	18

After calculating the average of class values we sort them in decresing order by average. This is shown in table 3.2.

In order to transform the nominal attribute car-name into numeric attributes we create new boolean attributes, one for each possible split of these four nominal values in the order listed. The new attributes and the correspondence between car-name and the values for the new attributes is show in table 3.3.

3.2.2 Defining Split Points

For each attribute in the new dataset we sort the values of each attribute in increasing order and define a "split point" of an attribute as the midpoint

Table 3.3: Creation of boolean attributes

	volkwagen	toyota	chevrolet	ford
volkwagen	1	0	0	0
volkwagen_or_toyota	1	1	0	0
volkwagen_or_toyota_or_chevrolet	1	1	1	0

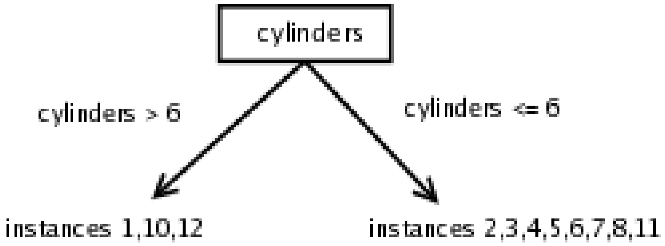

Figure 3.1: Cylinders is the root node of the tree

between two subsequent values of the attribute. This is shown in table set 1.

3.2.3 Building the Tree

Algorithm

We consider the set of split points of all attributes and select as the condition for the root node on your tree the split point that maximizes the value of the following formula:

$$SDR = sd(i_1) - \left(\left(\frac{k_1}{n} \right) * sd(i_b) + \left(\frac{k_2}{n} \right) * sd(i_a) \right) \tag{3.1}$$

where sd stands for standard deviation, k_1 is the number of instances with attribute value below split point, k_2 is the number of instances with attribute value above split point, n is the number of instances, i_1 is mpg over all instances, i_b is the mpg of instances with attribute value below split point, and i_a is the mpg of instances with attribute value above split point.

Determining the Root Node

Table set 1 shows the calculations of split points of all attributes and their respective SDR values. According to the calculations the root node of the tree is cylinders (see figure 3.1).

L1: cylinders $>$ 6 Instances 1, 10, and 12 lie in this child. This is less than four data instances so we stop splitting.

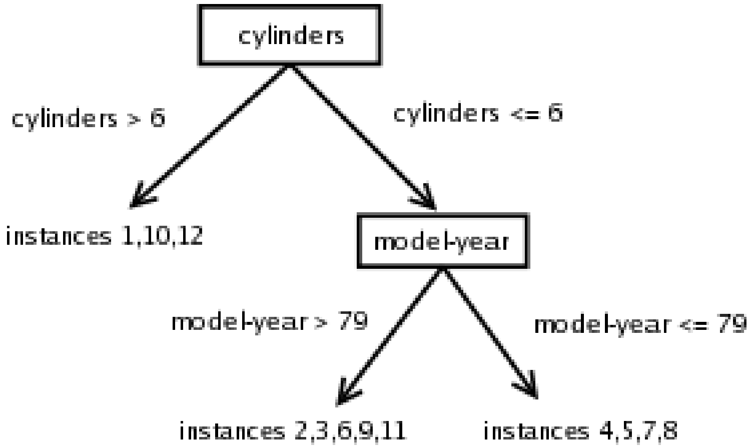

Figure 3.2: model-year is the next split point

L1: cylinders $<$ 6 Instances 2 thourgh 9 and 11 lie in this child. The standard deviation of the class value of the node's instances is $sd(27, 34, 24, 28, 32, 26, 29, 44, 28) = 5.97 > 0.05 \cdot sda$ where $0.05 \cdot 8.87 = 0.4435$; therefore, we continue splitting.

Level Two Split: cylinders $>$ 6

We remove the instances for which $cylinders < 6$ and continue procedure with the instances for which $cylinders > 6$. Table set 2 shows the split points of all attributes for which $cylinders > 6$ and their respective SDR values. According to the calculations the next split ppoint is model-year (see figure 3.2.

L2: model-year $<$ 79 Instances 4, 5, 7, and 8 lie in this child. The standard deviation of the class value of the node's instances is $sd(24, 28, 26, 29) = 2.22 > 0.05 \cdot sda$ where $0.05 \cdot 8.87 = 0.4435$; therefore, we continue splitting.

L2: model-year $>$ 79 Instances 2, 3, 6, 9, 11 lie in this child. The standard deviation of the class value of the node's instances is $sd(27, 34, 32, 44, 28) = 6.78 > 0.05 \cdot sda$ where $0.05 \cdot sda = 0.4435$; therefore, we continue splitting.

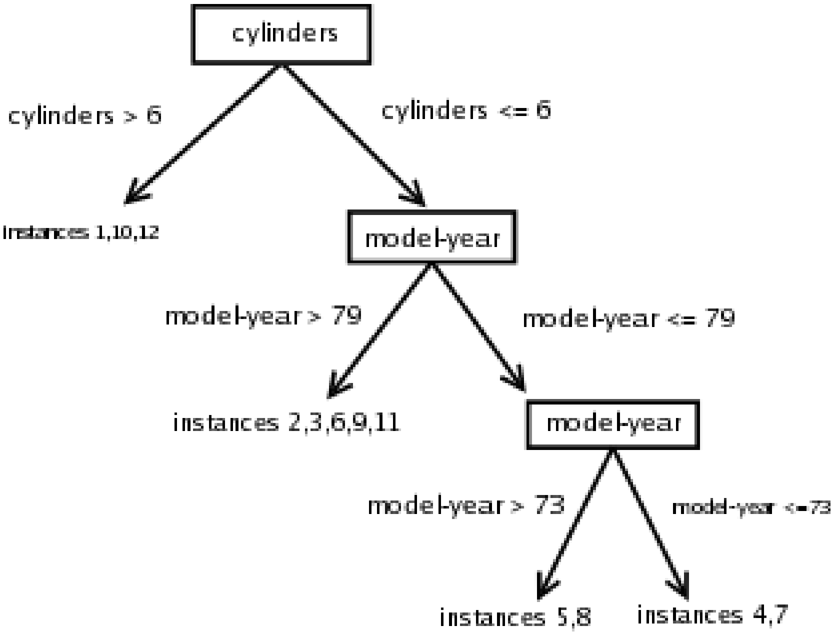

Figure 3.3: model-year is the next split point

Level Three Split: model-year $<$ 79

We remove the instances for which *model-year* $>$ 79 and continue procedure with the instances for which *model-year* $<$ 79. Table set 3b shows the split points of all attributes for which *model-year* $<$ 79 and their respective SDR values. According to the calculations the next split ppoint is model-year (see figure 3.3.

L3: model-year $<$ 73 Instances 4 and 7 lie in this child. This is less than four instances so we stop splitting.

L3: model-year $>$ 73 Instances 5 and 8 lie in this child. This is less than four instances so we stop splitting.

Level Three Split: model-year $>$ 79

We remove the instances for which *model-year* $<$ 79 and continue procedure with the instances for which *model-year* $>$ 79. Table set 3b shows the split

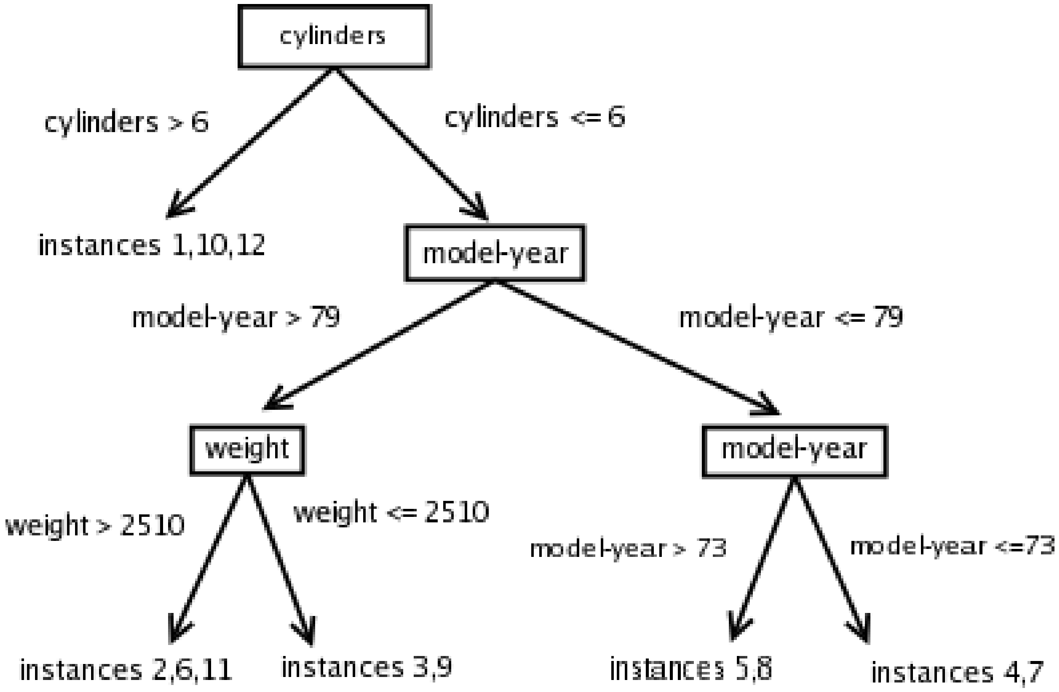

Figure 3.4: weight is the next split point

points of all attributes for which *model-year* $>$ 79 and their respective SDR values. According to the calculations the next split ppoint is weight (see figure 3.4.

L3: weight $<$ 2510 Instances 3 and 9 lie in this child. This is less than four instances so we stop splitting.

L3: weight $>$ 2510 Instances 2, 6, and 11 lie in this child. This is less than four instances so we stop splitting.

3.2.4 Regression Tree

For each leaf node in the tree we compute the value that would be predicted by that leaf in the case of a Regression Tree. We compute this value by taking the average mpg of the instances that belong to each leaf. Since this

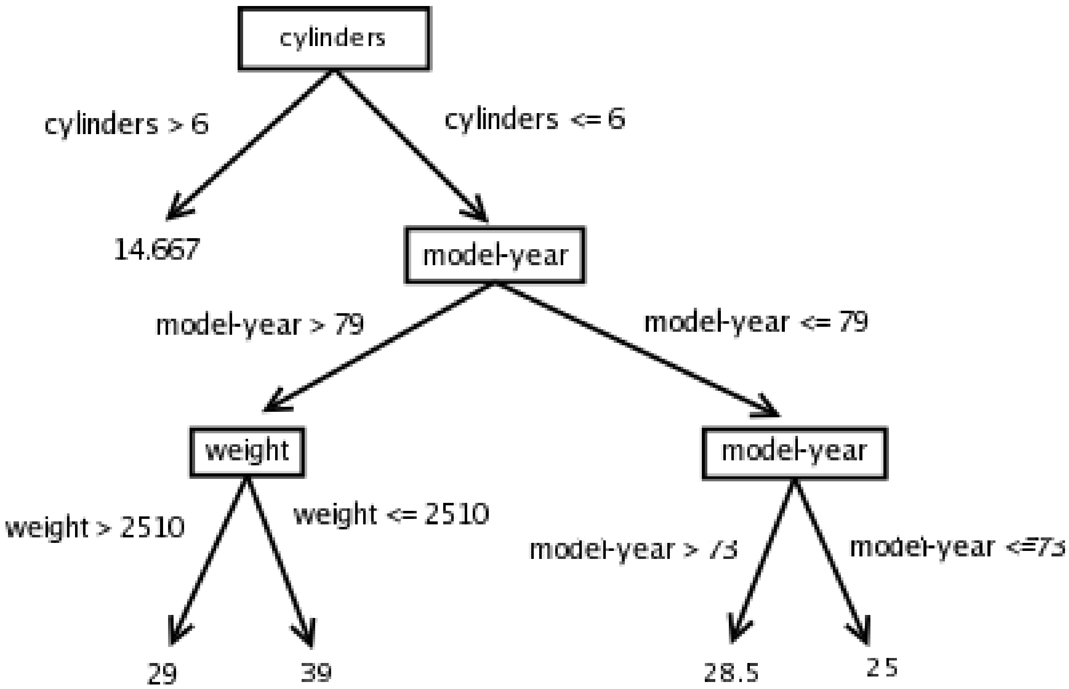

Figure 3.5: The regression tree

is a fairly simple calculation, the steps are not shown here. The final tree is shown in figure 3.5.

3.2.5 Linear Regression

For each leaf node in the tree we compute the linear regression formula that would be used by that leaf to predict the class value in the case of a Model Tree. The structure of the model tree is identical to that of the regression tree constructed above. The only difference is the predictive function in each of the nodes. They are linear regressions that predict the target attribute (mpg) in terms of the nodes in the tree.

The linear regressions L1, L2, L3, L4, and L5 are usually calculated as follows:

$$LX = w_0 + w_1 \cdot cylinders + w_2 \cdot model\text{-}year + w_3 \cdot weight + w_4 \cdot model\text{-}year = classvalue$$

In order to find the coefficients of the linear regression formula, we run the linear regression method implemented in the Weka system for the appropriate

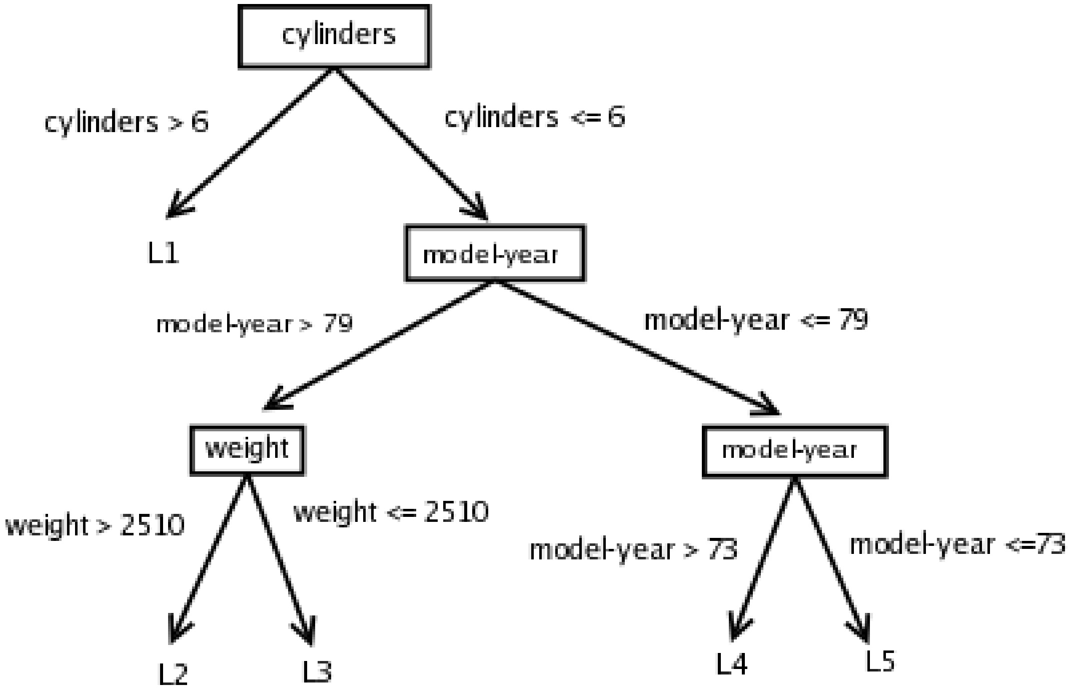

Figure 3.6: The model tree

Table 3.4: Linear regressions for leaf nodes

Linear Regression	Weka Output
L1	mpg = 14.6667
L2	mpg = 29
L3	mpg = 39
L4	mpg = 28.5
L5	mpg = 25

data instances (those that belong to the leaf). Since each leaf has a small number of instances, all Weka produces are linear regressions of the form $mpg = X$ where X is some value, implying a weight of 0 for the remaining attributes. Table 3.4 shows the linear regressions L1, L2, L3, L4, and L5.

3.3 Testing

3.3.1 Predicting the Class Values (mpg)

Table 3.5 shows the class value prediction of each of the three numeric models constructed (linear regression equation, model tree and regression tree) for each of the test instances. Finding the predicted class value is a matter of plugging in number in the appropriate places for the linear regression equation. For the regression tree it is a matter of navigatin the tree to the proper leaf node. For the model tree it is a combination of navigating the tree to the proper leaf node and of plugging the appropiate numbers in the equations.

3.3.2 Error Measures

Table 3.6 shows the root mean-square error and mean absolute error for each of the three numeric models constructed (linear regression equation, model tree, and regression tree). The predicted values p were obtained from table 3.5 and the actual values a were obtained from the test instances. For the sake of brevity only the formulas for root mean-square error and mean absolute error are shown below.

Table 3.5: Predicting mpg

	Linear Regression	Model Tree Prediction	Regression Tree Prediction
13,8,150,4464,12,73,chevrolet	12.4	14.6667	14.667
21,4,72,2401,19.5,73,chevrolet	23.4	25	25
20,6,122,2807,13.5,73,toyota	26.35	25	25
27.5,4,95,2560,14.2,78,toyota	27.99	28.5	28.5
27,4,60,1834,19,71,volkswagen	28.37	25	25
31.5,4,71,1990,14.9,78,volkswagen	32.24	28.5	28.5
21,4,86,2226,16.5,72,ford	17.82	25	25
36.1,4,66,1800,14.4,78,ford	25.45	25	28.5

Table 3.6: Error Measures

	Linear Regression Error	Model Tree Error	Regression Tree Error
root mean-square error (see p. 148)	4.65	4.05	4.05
mean absolute error (see p. 148)	3.22	3.53	3.53

The root mean-square error equation is shown in equation 3.2.

$$\sqrt{\frac{(p_1 - a_1)^2 + ... + (p_n - a_n)^2}{n}} \tag{3.2}$$

The mean absolute error equation is shown in equation 3.3.

$$\frac{|p_1 - a_1| + ... + |p_n - a_n|}{n} \tag{3.3}$$