Homework 10&11

Due date: Dec 7, 2005

The selected database information

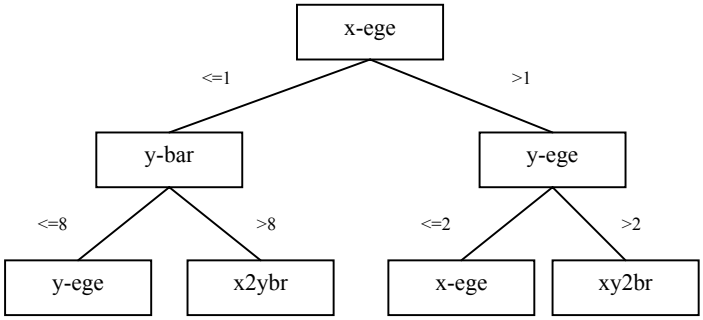
Database Name: Letter Recognition Database
From: David Slate
Based on: various fonts
Instances: 20,000 instances (712565 bytes) (.Z available)
Attributes: 17 attributes: 1 class (letter category) and 16 numeric (integer)
Additional Information: No missing attribute values
Objective: The objective is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15. We typically train on the first 16000 items and then use the resulting model to predict the letter category for the remaining 4000. See the article cited above for more details.
Attribute Information:

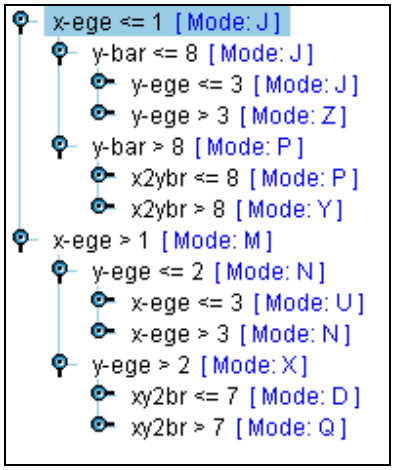
1.	lettr	capital letter	(26 values from A to Z)
2.	x-box	horizontal position of box	(integer)
3.	y-box	vertical position of box	(integer)
4.	width	width of box	(integer)
5.	high	height of box	(integer)
6.	onpix	total # on pixels	(integer)
7.	x-bar	mean x of on pixels in box	(integer)
8.	y-bar	mean y of on pixels in box	(integer)
9.	x2bar	mean x variance	(integer)
10.	y2bar	mean y variance	(integer)
11.	xybar	mean x y correlation	(integer)
12.	x2ybr	mean of x * x * y	(integer)
13.	xy2br	mean of x * y * y	(integer)
14.	x-ege	mean edge count left to right	(integer)
15.	xegvy	correlation of x-ege with y	(integer)
16.	y-ege	mean edge count bottom to top	(integer)
17.	yegvx	correlation of y-ege with x	(integer)

Data Analysis

The data in this database allows interested researchers to reveal understanding about the English letter recognition from the given attributes. However, it needs some kinds of algorithms to figure out what the relationships between each letter and the given attributes are. As a consequence, choosing appropriate models in Clementine to figure that out is a good start for the data analysis. The C5.0 node and neural net model node were being used to find out the facts hidden in this database. Also, the characteristics of each letter will be revealed by those two models as well.

For the first attempt, the C5.0 node retrieves the data samples randomly (25%) as inputs. After it ran completely, a decision tree was created. The first three attributes used to recognize the letter of the decision tree are shown in the chart below (because the tree is very big, it is hard to display the entire tree. So, I pick only an important part to display).



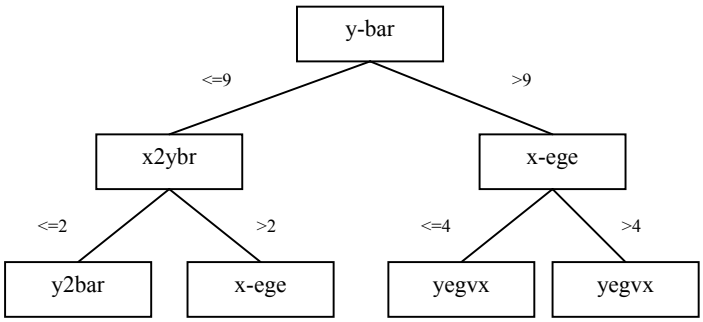


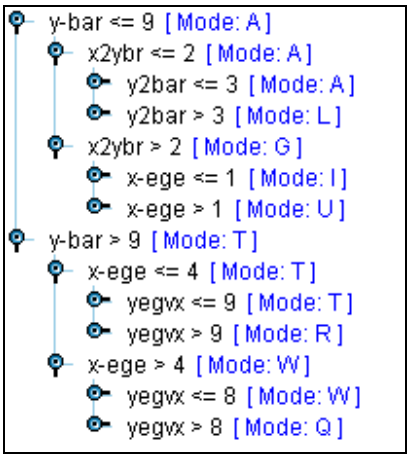
After that, the accuracy of the decision tree had been measured by comparing between attribute "lettr" and the new attribute ($CC-lettr) obtained from the C5.0 node. Using the analysis node, the result came out as follow:

Correct	**4165**	**82.67%**
Wrong	**873**	**17.33%**
Total	**5038**	

At this point, the accuracy of this decision tree is high enough to be used for letter characterization according the given attributes. For that reason, it convinces us that each of the letters tends to have unique characteristics (the pattern of the attribute values). For example, letter "J" is likely to have the value of attribute "x-ege" <= 1.

To improve the accuracy of the decision tree, another attempt has been made. By resizing the sample size to be 50% random, the first three attributes of the decision tree are shown as follow:

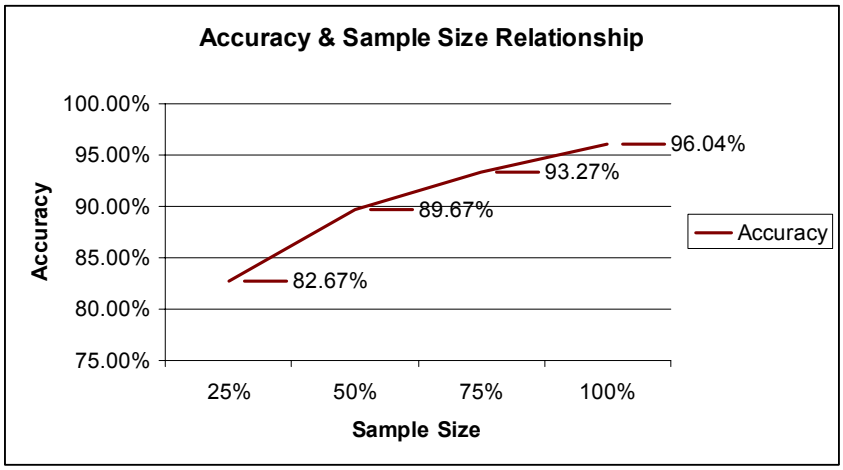




And, the analysis result came out as shown below.

Correct	**8945**	**89.67%**
Wrong	**1030**	**10.33%**
Total	**9975**	

It can be conclude that “the bigger the sample size is, the more the accuracy of the decision tree increases”. Both of the results convince us that the number of the data records might affect the overall accuracy of the decision tree. Having tried different size of samples with the C5.0 node, the relationship between the decision tree accuracy and the sample size is displayed below.



Now it is the neural net turn. Because the neural network is another powerful alternative that can be used to classify data and predict a group the data should belong to. It is a good candidate for this problem, the goal of which is letter recognition by learning from a given data set. As a result, the neural net node has been chosen as another model for mining this database.

After let the neural net node learn from a dataset reserved as a training set (¾ of the entire data has been reserved for the training set), the accuracy of the neural net node with 10 hidden nodes in the first layer is shown in the left column while the weight of the input nodes is in the right column.

Correct	**11341**	**75.61%**
Wrong	**3659**	**24.39%**
Total	**15000**	

xy2br	0.0913522
y2bar	0.090709
xegvy	0.0902532
y-ege	0.0868695
x-ege	0.0847132
x-bar	0.0832533
x2bar	0.0814758
width	0.0801124
yegvx	0.0793132
x2ybr	0.0779908
onpix	0.0768932
y-bar	0.0729374
xybar	0.061941
x-box	0.0556407
high	0.0554005
y-box	0.0239728

Unfortunately, the accuracy of the neural net node with 10 hidden nodes in the first layer for the test set (the other ¼ of the entire data reserved as a test set) seems to be lower than the training set. The result came out as follow:

Correct	**3653**	**73.06%**
Wrong	**1347**	**26.94%**
Total	**5000**	

y2bar	0.114582
y-bar	0.0930819
x-bar	0.0926523
xy2br	0.0924354
x2bar	0.0871913
y-ege	0.0842961
yegvx	0.0836077
x-ege	0.08325
xegvy	0.0815209
x2ybr	0.08003
x-box	0.061671
onpix	0.0616291
high	0.0585488
xybar	0.050713
width	0.0505949
y-box	0.0279642

To increase the percentage of the neural net accuracy, adding not only hidden nodes but also hidden layers seems to be a good idea. As a result, another neural net with 20 hidden nodes in the first layer and 15 hidden nodes in the second layer was created. After let the neural net

node learn from a dataset reserved as a training set (¾ of the entire data has been reserved for the training set), its accuracy and the weight of input nodes are shown in the table below.

Correct	**12609**	**84.06%**
Wrong	**2391**	**15.94%**
Total	**15000**	

x-ege	0.106641
y-ege	0.103948
xy2br	0.0979489
xegvy	0.094446
x2bar	0.0933467
y2bar	0.0926004
yegvx	0.0814213
y-bar	0.0770188
x-box	0.0766543
x2ybr	0.0744524
onpix	0.0716717
x-bar	0.0678165
xybar	0.063357
width	0.0609719
high	0.0566325
y-box	0.0293847

Like the previous case, the accuracy of the neural net node with 20 hidden nodes in the first layer and 15 hidden nodes in the second layer for the test set (the other ¼ of the entire data reserved as a test set) seems to be lower than the training set. The result came out as follow:

Correct	**4106**	**82.12%**
Wrong	**894**	**17.88%**
Total	**5000**	

y-ege	0.0963175
xy2br	0.0961787
x-ege	0.0959508
x2bar	0.0950814
xegvy	0.0872296
y2bar	0.0865713
y-bar	0.0823759
yegvx	0.0817468
x2ybr	0.0743128
xybar	0.0684665
width	0.0679684
x-bar	0.0666145
onpix	0.0586988
high	0.056228
x-box	0.0539023
y-box	0.0214771

To conclude, the accuracy of the neural net nodes are acceptable because it is still over 80% which can be indicated as a high accuracy level.

Conclusion

Although the neural network seems to be a powerful tool to provide a good guess in predicting a letter effectively because of its high accuracy level, the C5.0 come up with a better accuracy level. Although the C5.0 node uses only 5000 records for constructing those rules, it produces a set of rules with **82.67%** accuracy which is more than that of neural net node (the accuracy is **82.12%** even though it has many hidden nodes and layers). Another problem with the neural network is to find an appropriate numbers of hidden nodes as well as hidden layers which is difficult for novice users and sometimes even experts. Furthermore, the C5.0 node could bring better accuracy if it has a bigger set of data. As a result, if we think in term of processing time and memory used, the C5.0 becomes preferred method to construct the rules for this letter recognition because the neural net require much time and memory space to finalize appropriate weight comparing to the C5.0 node.

From this apparent result, it is convincing that the algorithm using in C5.0 node is a good technique for inventing a reasonable solution for recognizing a letter problem which mainly depends on the letter attributes.

Sources used

- The tutorial of "Clementine 9.0"
- The Help topics of "Clementine 9.0"
- Textbook: Data Mining Concepts and Techniques