6.034 Design Assignment 2

April 5, 2005

Weka Script Due: Friday April 8, in recitation
Paper Due: Wednesday April 13, in class
Oral reports: Friday April 15, by appointment

The goal of this assignment is for you to gain some practice in the application of machine learning algorithms to real data. We give you two data sets and a framework that will allow you to experiment with different learning algorithms on that data.

1 Data Sets

We ask you to build classifiers for two data sets:

1. **credit-g-500.arff**: This is a two-class data set related to credit rating in Germany. Information on the data set can be found at the top of the file.
2. **digits-2-4-5-9.arff**: This is a collection of 14x14 binary images of hand-written digits (2,4,5,9); see figure. There are 250 samples of each digit. Each image is converted into a feature vector by listing the content of the array in row-major order.

2 Experiments

We would like you to find an effective learning algorithm for each of these data sets. In order to do so, you should think about the general strengths and weaknesses of the different learning algorithms, as well as to experiment with them on the data.

Using the algorithm you choose, generate a classifier, and a prediction of how well it will perform on new data. We will run that classifier on some additional data and compare its performance to your predicted performance.

3 Write-up

In your write-up, describe process by which you developed the classifiers you did. You should answer the following questions, in detail, including supporting data in graphs or tables.

- What algorithms are generally expected to be appropriate for these data sets?

- How did you choose among the different algorithms? Report your chosen algorithm, as well as at least three others that you tried.
- How did you choose parameter settings for each algorithm? Report the parameters that gave you the best results.
- How did you come up with a prediction for how well the classifiers you delivered would perform on previously unseen data? Report your prediction.
- Compare the best performance you got on each data set with the performance if you had picked the class (a) at random (unbiased coin flip) or (b) by always predicting the most prevalent class in the training data.
- What classifier would you use in the credit data if it were twice as expensive to say that a person with bad credit was going to have good credit, as to say that a person with good credit would have bad credit?
- What two attributes seem to be most relevant in each data set? Or is it the case that they're all just about equally significant? Explain how you determined this, and why you think you obtained the answer you did.
- In the multiple-class digits problem, which two digits are most frequently confused by your classifier. Does that make sense to you?

4 Grading

There will be a late penalty of 20% per day assessed, with no credit given for assignments turned in after the oral report.

Grading will be broken down as follows:

30: Good plan for choosing and validating algorithm, parameters, and classifiers
15: How effective are the classifiers on new data
10: How good is the supplied performance prediction
5: Completing the Weka script given at the end of this handout
20: Clarity and organization of written report
20: Clarity and understanding in oral report

5 Software

We ask you to use the **Weka** environment for machine learning. You can download the software from:

`http://www.cs.waikato.ac.nz/~ml/weka/`

The software is written in Java and should run under Windows, Linux and Mac. A word of warning: Weka will often run out of memory and need to be re-started, so save results as you go.

Within this system, you can find the major algorithms that we've studied:

- K Nearest Neighbor (called IBk in Weka)
- Decision Trees (called J48 in Weka)
- Naive Bayes (called Naive Bayes in Weka)
- SVM (called SMO in Weka)

There are many other algorithms that you can experiment with if you'd like to, as well, but we expect you to consider these methods. Note that the SMO implementation is relatively slow compared to the others. You might want to use cross-validation sparingly with SMO since if you find that it takes a long time to run.

You should try at least one additional interesting processing step on at least one of the data sets. You might, for example:

- Run a feature selection or dimensionality reduction algorithm on the digits data
- Try normalizing the attributes before applying nearest neighbor

Don't just try something at random, though. Think about what is likely to help classification performance, and justify your choice in your write-up.

6 Using Weka

To make sure that you don't (immediately) run out of memory when running the program. Under Linux and Mac OS X, you should start Weka by connecting to the Weka directory (weka-3-4-4) and calling java with the following arguments:

```
java -mx100000000 -oss100000000 -jar weka.jar
```

If you are running Windows, Weka will install under

```
c:\Program Files\Weka-3-4
```

In that folder, you should see a file called `RunWeka.bat`, edit that file to add the `-mx` and `-oss` arguments to the `java` call. When you start Weka from the Start menu, you should see a console window with the appropriate call to java.

There are also a couple of documentation PDFs in the Weka directory: Tutorial.pdf and ExplorerGuide.pdf. This page also has some useful information and links:

```
http://www.comp.leeds.ac.uk/andyr/teaching/db32/weka-db32.pdf
```

Here's a script to follow that will expose you to the basics of operating Weka. Report the performance values that we ask for below. Hand them in at recitation on April 8. This will be worth 5 points on your grade for the assignment. If you have any questions, please bring them up at recitation.

```
java -mx100000000 -oss100000000 -jar weka.jar
Go to Weka GUI Chooser window
Click Explorer
Choose Preprocess Tab at the top of the new window
Open File
<pick breast-cancer.arff>
```

Clicking on the name of the different "Attributes" shows a histogram of the values on the bottom right, colored by the Class variable (or whatever attribute is chosen in the pull down above the graph). If you click on the Class attribute, you'll see how many of each class there are in the dataset.

```
Click Classify Tab
Click Choose in Classifier Pane, under Trees, pick J48 (which is Decision Tree)
Click the Percentage Split button under Test Options
 (this holds out 1/3 of the data for validation; you could instead do cross-validation)
Click Start (always make sure it says Class right above the Start button)
Report the line "Correctly Classified Instances" in the Classifier Output window
Click on the Classifier pane (where it says "J48 -C ...").
In the dialog window, change the value of MinNumObj to 1
Click OK
Click Start
Report the line "Correctly Classified Instances" in the Classifier Output window
Click Choose in the Classifier Pane, under Bayes, pick NaiveBayes
Click Start
Report the line "Correctly Classified Instances" in the Classifier Output window
Click Choose in the Classifier Pane, under functions, pick SMO (which is an SVM)
Click Start
Report the line "Correctly Classified Instances" in the Classifier Output window
Click on the Classifier pane (where it says "SMO -C ...").
In the dialog window, change the value of gamma to 0.1 and useRBF to True
Click OK
Click Start
Report the line "Correctly Classified Instances" in the Classifier Output window
Click Choose in the Classifier Pane, under lazy, pick IBk (which is K Nearest Neighbor)
Click Start
Report the line "Correctly Classified Instances" in the Classifier Output window
Click on the Classifier pane (where it says "IBk -K ...").
In the dialog window, change the value of KNN to 3
Click OK
Click Start
Report the line "Correctly Classified Instances" in the Classifier Output window
```

The procedure above is for a binary (two-class) classification problem. Most of the classifiers in Weka have been extended to handle multiple classes, including Naive Bayes. The SMO (SVM) algorithm is inherently for two classes, if you use it on a data set with more than two classes; Weka will build classifiers for each pair of classes (1-against-1) by default. You can have control of how to build a multi-class classifier from a binary classifier by using the Weka "Meta" classifier MultiClassClassifier (see below).

```
Choose Preprocess Tab
Open File
<pick vehicle.arff>
Click Classify Tab
Click Choose in the Classifier Pane, under meta, pick MultiClassClassifier
Click on the Classifier pane (where it says "MultiClassClassifier -M
...").
In the dialog window, the method pull-down allows you to choose
1-against-all or 1-against-1 (and some other options).  Pick 1-against-all.
In the same dialog window, click Choose in the Classifier entry
Under functions, pick SMO
Click OK
Click on the Classifier pane (where it says "MultiClassClassifier -M ...").
In the dialog window, click the Classifier pane (where it says "SMO...")
Change the value of C to 10.0
```

```
Click OK, Click OK
Click Start (make sure it says Class right above the Start button)
Report the line "Correctly Classified Instances" in the Classifier Output window
```

Note that you can Right click on each of the entries in Results list and choose to save the output to a file.

To select a subset of the features (attributes) for the vehicle.arff dataset:

```
Choose Select attributes Tab
Leave default choices for Attribute Evaluator and Search Method
Click Use full training set button
Click Start
Report the line Selected attributes
Click the Preprocess Tab
On the Attributes, click on the attributes that were NOT selected
Make sure that you DON'T click on the Class attribute.
Click on the Remove button
Repeat the MultiClassClassifier operation we described above.
```

Here's example of comparing multiple algorithms on a dataset:

```
Go to Weka GUI Chooser window
Click Experimenter
Click Setup Tab at the top of the new window
Click New
Pick a name for an ARFF File under Results Destination
Make sure Experiment Type is Cross-validation and Number of Folds is 10
Click Add new under Datasets
<pick breast-cancer.arff>
Click Add new under Algorithms
Click Choose, under bayes, pick NaiveBayes, click OK
Click Choose, under trees, pick J48, click OK
Click Choose, under lazy, pick IBk, set KNN to 3, click OK
Click Choose, under functions, pick SMO, set C to 10.0, gamma to 0.1 and useRBF to True, click OK
Click the Run Tab at the top of the window
Click Start (wait till it says Finished)
Click Analyse Tab
Click Experiment button
Click Perform test button
Click Save output
Report the performance of the methods
```

7 Visualization

Weka has tools for helping you understand the classifiers you're learning. If you right-click on a classifier result, you get a menu of options.

Visualize tree: (available only for classifiers that build trees) will show the tree in a new window. If you resize the window, and then right-click (or option-click) in the window, you can resize the tree to fit in the window.

Visualize classifier errors: correctly classified instances are represented by crosses, errors by squares. You can pick which attributes to use on the X and Y axes.

Visualize margin curve: The margin is the difference between the probability predicted for the actual class and the highest probability predicted for the other classes. (So, for a single class, if it is predicted to

be positive with probability p, the margin is $p - (1 - p) = 2p - 1$.) Unfortunately, although this can be calculated for SVM, it's not available in Weka for SMO. It's also not available for nearest neighbor.

Visualize threshold curve: Generates a plot illustrating the tradeoffs in prediction that are obtained by varying the threshold value between classes. For example, with default threshold value of 0.5, the predicted probability of "positive" must be greater than 0.5 for the instance to be predicted as "positive". (Unfortunately, not applicable for SMO or nearest neighbor).