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 and 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 contents 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 do 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?

- Howffidffyouffhooseffamongffheffifferentffalgorithms?ffReportffyourffhosenffalgorithm,ffasffwellffasffatffeastff threeffotherssffhatffyouffried.ff
- Howffidffyouffhooseffparameterffettingsfforfeachffalgorithm?ffReportffheffparametersffhatffavefffyouffheff bestffesults.ff
- Howffidffyouffomefupffwithffaffpredictionfforfhowffwellffheffassifiersffyouffeliveredffwouldffperformffonff previouslyffinseenfflata?ffReportffyourffprediction.ff
- Compareffheffbestffperformanceffyouffgotffonfeachfflataffetffwithffheffperformanceffiffyouffhadffpickedffheff classff(a)ffatffrandomff(unbiasedffoinfflip)fforff(b)ffbyffalwaysffpredictingffheffmostffprevalentffassffinffheff trainingfflata.ff
- Whatffassifierffwouldffyouffseffnffheffreditfflataffffftfwereftwiceffasfexpensiveftofayfthatffapersonfwithff badffreditfwasffgoingfftofhaveffgoodffredit,ffasftofayfthatffapersonfwithffgoodffreditffwouldfhaveffbadff credit?ff
- Whatffwoffattributesffseemfftofbeffmostffrelevantffinfeachfflataffet?ff Orffisffitffheffasefffhatffhey'reffallff justffaboutffequallyffignificant?ffExplainffhowffyouffletermineffhis,ffandffwhyffyouffhinkffyouffbtainedffheff answerffyouffid.ff
- Inffheffmultiple-classffigitsffproblem,ffwhichffwoffigitsfareffmostffrequentlyffonfusedffbyffyourffassifier.ff Doesffhatffmakeffensetoffyou?ff

4- Grading-

Thereffwillffbeffaflateffpenaltyffoff20%ffperfflayffassessed,ffwithffnoffreditffgivenfforfassignmentsffurnedffinffafterff theffralfreport.ff

Grading willffbeffrokenffownffasffollows:ff

30:ff Goodffplanfforfhoosingffandffvalidatingffalgorithm,ffparameters,ffandffassifiersff
15:ff Howffffectiveffareffheffassifiersffonffnewfflataff
10:ff Howffgoodffisffheffsupplied ffperformanceffpredictionff
5:ff CompletingffheffWekaffscriptffgivenffatffheffndffoffhisffandoutff
20:ff Clarityffandffrganizationffoffwrittenffreportff
20:ff Clarityffandffinderstandingffnffralfreportff

5- Software-

Weffaskffyouffoffseffheffwekaffnvironmentfforfachineffearning.ffYouffanfflownloadffheffoftwareffrom:ff

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

Theffoftwareffisffwrittenffnffavaffandffhouldffunfinderffindows,ffinuxffandffac.ffAffwordffoffwarning:ffwekaff willffoftenffunffoutffoffmemoryffandffeedffoffeffe-started,ffoffaveffesultsffasffyouffo.ff

Withinffhisffystem,ffyouffanffindffheffmajorffalgorithmsffhatffe'veffstudied:ff

- KffNearestffNeighborff(calledffBkffnffWeka)ff
- Decisionffreesff(calledff48ffnffWeka)ff
- NaiveffBayesff(calledffNaiveffBayesffnffWeka)ff
- SVMff(calledffMOffnffWeka)ff

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.