

Lab 1

CS 6243 – Spring 2005
Tom Bylander, Instructor

assigned January 20, 2005
due February 10, 2005

This lab is intended to help you start understanding how Weka algorithms are put together. This lab will modify the IBk classifier.

Currently, the IBk classifier normalizes the values of a numeric attribute by mapping the minimum value to 0, the maximum value to 1, and other values linearly inbetween. The minimum and maximum values of attributes are kept in arrays `m_Min` and `m_Max`, and the `difference` method calls the `norm` method performs the normalization.

Add a new option `-S` to IBk to perform “standardization” of numeric values instead of normalization. For a numeric attribute, this is done by calculating the sample mean u and standard deviation s of the values. A given value x is then standardized by $(x - u)/s$.

Your new `IBk.java` should be able to take the place of the old `IBk.java`. To set things up, you might want a subdirectory with Weka’s source code and classes in it. This can be done by:

```
jar xvf $WEKAHOME/weka-src.jar
jar xvf $WEKAHOME/weka.jar
```

so you can recompile `IBk.java` by:

```
javac weka/classifiers/lazy/IBk.java
```

and run it by (for example):

```
java weka.classifiers.lazy.IBk -S -t $WEKAHOME/data/UCI/iris.arff
```

If you are using JDK 1.5, you will probably want `-source 1.4` in your `javac` command. Be sure that “.” is the first thing in your `$CLASSPATH`.

There a number of details with adding an option to make everything fit nicely with the rest of Weka. Look at what `IBk.java` does with its other options, and yes, you want to update the comments and add a tip text method, too.

Email me your lab with `IBk.java` as one attachment. One other attachment (not four more attachments) should be its performance on the iris and glass datasets (the full datasets, not the simplified ones) using `-K 1` and `-K 3`. Only email me once; points will be deducted for multiple emails.