36-350: Data Mining

Lab 3
Date: September 12, 2003 **Due: end of lab**

Interspersed throughout this lab are questions that you will have to answer at check-off.

1. Download the files for this lab from the course web page to the desktop:

   ```
   http://www.stat.cmu.edu/~minka/courses/36-350/lab/
   ```

2. Open a Word or Notepad document to record your work.

Start R

3. `Start -> All Programs -> Class software -> R 1.7.0`

4. Load the special functions for this lab:

   ```
   File -> Source R code...
   ```

   Browse to the desktop and pick `lab3.r` (it may have been renamed to `lab3.r.txt` when you downloaded it). Another window will immediately pop up for you to pick the `mining.zip` file you downloaded.

The dataset

5. The dataset is a larger version of the politics/religion collection in lab 1. The counts are in `doc` and the labels are in `doc.labels`. For this lab, you only want to know if a word is present in a document is not. The following constructs a new matrix containing only presence information:

   ```
   is.present = (doc > 0)
   ```

Information

6. Compute the expected information between `doc.labels` and each column of `is.present`. Sort the words by information and record the top 10:

   ```
   sort(s)
   ```

7. Repeat using the actual information when the word is present. Try with and without smoothing.

8. Find a word which has high expected information, but relatively low actual information (with smoothing). Compute the table of counts between its presence and the label. Record it for the homework.

9. Find a word which has high actual information without smoothing, but low actual information with smoothing. Compute the table of counts between its presence and the label. Record it for the homework.

Interaction

10. The following constructs a vector of row numbers which have the word “god”:

    ```
    i = which(is.present[,"god"])
    ```

    Compute the expected information between `doc.labels` and each column of `is.present`, for this set of rows. *What word has the most information?*

11. Construct a three-way table of counts between `doc.labels`, `is.present[,"god"]`, and the word you picked. Make an information graph.

12. You can now get checked off.

Computing information If `x` is a matrix of outcomes and `y` is a vector, then

```
s = information(x,y)
```

computes the expected information between `y` and each column of `x`. The result is put into a vector `s`. The actual information for a particular value in `x` (for example, “TRUE”) is given by

```
s2 = information(x,y,actual="TRUE")
```

To add 1 to the counts before computing the information, use

```
s3 = information(x,y,actual="TRUE",smooth=1)
```

You can also compute information from the output of `table`:

```
information(table(x[,1],y))
```

This is the same as `s[1]` above.

Tables of counts If `x` is a vector of outcomes and `y` is a similar vector, then

```
table(x,y)
```

returns a table counting each pairing of a value in `x` with a value in `y`. This also works with more than two vectors:

```
table(x,y,z)
```

You can name the dimensions of the table by naming the arguments to `table`, such as

```
table(first=x,second=y)
```

Information graph If `tab` is a two-way or three-way table of counts, then

```
information.graph(tab)
```

plots the expected information and interaction between the variables.