

# CG168 Homework 5

Mark Johnson

due 11th November 2008

This homework requires you to construct a  $k$ -means clustering algorithm for the samples produced by a mixture of multinomials. You'll be working with three different data sets, all available in `/course/cog168/asgn/data`:

- `dice.test`: Each line contains the results of rolling a die a number of times. `dice.dev` is a smaller version of `dice.test` for you to practice on. The first few lines of `dice.dev` are:

```
2 2 1 2 0 5 1 1 1 1
4 0 4 3 4 0 1 3 4 4
4 4 3 0 1 0 4 3 4 1
```

Each number corresponds to a roll of a possibly biased die; the same die is used for each roll on the same line. A small number of different dice (i.e., each die has different probabilities) are used to generate the different lines within each file. Your goal is to figure out which die was used to generate which line in these files.

The file `dice.dev.gold` tells you which die was used to generate which line of `dice.dev`. It begins as follows:

```
0
1
1
```

This says that die 0 was used to generate the first line of `dice.dev`, while die 1 was used to generate the next two lines. Your goal here is to produce a file like `dice.dev.gold`, but for `dice.test`.

- `motherese.txt`: This file contains sentences produced by mothers talking to their children. The first few lines are:

```
big drum ?
horse .
who is that ?
```

Your goal is to cluster these by topic, assuming that the words in each sentence are generated from an unknown categorical distribution for the topic (i.e., like a gigantic die).

- **brown.txt**: This file contains 500 samples from the Brown corpus, with all words transformed to lowercase and with the 50 highest frequency words deleted. Again, your goal here is to cluster these by topic.

You should assume that the clusters are characterized by categorical distributions; i.e., cluster  $k$  is characterized by a vector  $\boldsymbol{\theta}_k = (\theta_{k,1}, \dots, \theta_{k,\ell})$ , where for each  $k$ ,  $\sum_{j=1}^{\ell} \theta_{k,j} = 1$ .

Suppose  $\mathbf{x}$  is a sequence of samples (e.g., rolls of a die), and  $x_j$  is the number of times outcome  $j$  is observed in  $\mathbf{x}$ . Then the “distance” between  $\mathbf{x}$  and cluster  $k$  is:

$$d(\mathbf{x}, \boldsymbol{\theta}_k) = - \sum_{j=1}^{\ell} x_j \log_2 \theta_{k,j}$$

### Homework:

1. Write an incremental  $k$ -means clustering algorithm and apply it for  $m = 2, \dots, 20$  clusters to each of these data sets 20 times with different random initializations. For each data set and each  $m$ , report the average and the lowest value of the sum of the distances between the clusters and the data elements. How many clusters do you think are actually present in each of the data sets?
2. For your “best” clustering with 10 clusters for **motherese.txt** and **brown.txt**, report the 10 most salient words in each cluster.
3. (200-level) Implement a split-merge version of  $k$ -means, and compare the performance of your split-merge clusterer with the incremental  $k$ -means clusterer. Does it find lower intra-cluster divergences?

You might find the following programs in `/course/cog168/asgn/python` helpful.

- **eval-clusters.py** reads from two files containing cluster ids and prints out the accuracy, the 1-to-1 accuracy and the VI statistic.
- **prominent-members.py** reads a data file and a cluster id file and prints out the most prominent members of each cluster.