Cognition, Language & Communication

Computerlab Markov Models

(18-9-2014)

Hand-in a brief report with (staccato) answers on the questions from this tutorial on Monday.

1 R

In today's computer lab we will work with R, an environment for data analysis and computer simulation. If you have never worked with R, you should start by (quickly) going through the first 3 pages of Bart de Boer's *Hints for Using R*.

2 Transitional Probabilities

Consider the following songs from two hypothetical song birds:

Bird 1:

abcabbbc
abbbbbcccccc
abcabcabbcc

Bird 2:

abbbcabbbcabbbc
abbbcabbbcabbb
abbbca
abbbcabbbcabbbc
abbbcabbbc

Question 1 *What are some of the patterns that you observe in these repertoires?*

Transitional probabilities are the term we use for the probabilities of any element in a sequence to be followed by any other element. We often introduce a special symbol (e.g., #) to mark the beginning of a song, and another symbol (e.g., $) to mark the end of a song, so that we can treat beginning and end the same way as transitions in the middle of a song (i.e., P(#,a) would be the probability of a song starting with an a). Common practice (given sufficient data) is to estimate the probability of transitioning from x to y as the relative frequency of the pair xy (relative to the frequency of x on its own):

$$P(x, y) \approx \frac{\text{count}(xy)}{\text{count}(x)}$$

Question 2 *Draw for each bird a 5×5 matrix with transitional probabilities. Which of the patterns from the previous question are captured in that matrix?*

In R, it easy to do calculations with such matrices. You can enter a vector of numbers with the function `c()`, and turn it into a matrix with the function `matrix(vec,width,height)`. To take the transpose of a matrix (i.e., flipping around horizontal and vertical dimensions) we use `t()`. E.g., we can create a 4×4 transitional probability matrix like this:

```
tm <- t(matrix(c(0.8,0.2,0.0,0.0, 0.1,0.8,0.1,0.0, 0.0,0.0,0.0,1.0, 1.0,0.0,0.0,0.0),4,4))
```

And give the rows and columns names with the following commands:

```
colnames(tm) <- c("a","b","#","$")
rownames(tm) <- c("a","b","#","$")
```

Question 3 *Enter both of your matrices in R (with different names). Have a look at the result by typing the name of the variable you used (e.g.* `tm1`*).*

You can now use the matrix to *generate* songs and check whether the songs generated look somewhat like the data you observed:

```
symbol="$"; symbols=NULL; while (symbol!="#") {
   symbol=sample(colnames(tm),1,prob=tm[symbol,]);
   symbols=append(symbols,symbol) }
paste(symbols)
```

Question 4 *Briefly describe what these commands do. Hint: You can check the descriptions of R functions by typing there name preceded by a question mark (e.g.,* `?paste`*).*

Question 5 *Generate 6 songs and briefly describe the structural commonalities and differences with the songs above.*

3 Ngrams

Transitional probabilities describe a dependency between *two* items (the probability of the *next* observable element given the *current* observable element) and are therefore commonly called *bi*grams. We can generalize from bigrams to dependencies between the next element and any number of previously generated elements. This gives us *Ngrams*, with $N = 2$ for bigrams, $N = 3$ ("trigrams") if we consider the last 2 elements for computing the probability of the next, etcetera.

Question 6 *What N would you need to model the pattern from bird 2 above properly?*

You can do various calculations with ngrams models (and the more expressive Hidden Markov Models) by using the package `HMM`:

```
install.packages("HMM")
library(HMM)
```

To study a bigram model with this package, you have to define states (corresponding to the elements) and transitional probabilities. (We will initially to use the HMM's distinction between observable and unobservable states, and use the same names for both them, an set the emission probabilities to the identy matrix).

```
tm <- matrix(c(0.6,0.1,0,0.4,0.5,0,0,0.4,1),3,3)
hmm <- initHMM(c("A","B","#"), c("a","b","#"), startProbs=c(1,0,0),
   transProbs=tm, emissionProbs=diag(3))
```

Having defined an HMM this way, we can use it to generate example songs, but also to compute the likelihood of an observed song (with the function `forward()`:

```
simHMM(hmm,10)

obs <- simHMM(hmm,10)$observation
forward(hmm,obs)
```

Question 7 *What likelihood does your transitional probability model assign to some of the songs above?*

4 Natural language

Ultimately, we would like to test how well ngram models (with various n's) and richer models (such as HMM's) model natural language. Today, we won't yet be able to make that analysis, but we can already count bigrams in texts by using the package `tau`:

```
install.packages("tau")
library(tau)
```

Download a text from the internet, e.g. the novel "Tom Sawyer" from the course website. Put the file in your project folder, change your working directory (top left menu) to that project folder and run:

```
myfile <- readLines("ts-text-dutch.txt")
```

You can now count the occurrences of words, and plot the frequency distribution with:

```
monograms = textcnt(myfile, n = 1, method = "string")
monograms = monograms[order(monograms, decreasing = TRUE)]
plot(monograms,log="xy")
```

Question 8 *What kind of distribution is this? Note that both axes are using a logarithmic scale.*

Bigrams frequencies (not probabilities) can be extracted in the same way:

```
bigrams = textcnt(myfile, n = 2, method = "string")
bigrams = bigrams[order(bigrams, decreasing = TRUE)]
plot(bigrams,log="xy")
```

Now you can compute the probability of a specific line from the text corpus (say line 23) under the bigram model, by extracting all bigranms from that line and retrieving the relevant monogram and bigram frequencies:

```
senbg <- textcnt(myfile[23], n = 2, method = "string")
bigrams[names(senbg)]

senmg <- sapply( strsplit(names(senbg)," "), function(m) m[1] )
monograms[senmg]

#sentence loglikelihood (ignoring start and end)
sum(log(senbg/monograms[senmg]))
```

Question 9 *What is the total log likelihood of the sentence? This analysis ignores the start and end of sentence probabilities. How would they change the log likelihood?*

Question 10 *Think of an interesting question you can answer (without too much trouble) with the tools we looked at. Show results, or discuss where it failed.*