Cognition, Language & Communication

Computerlab Model Selection

(25-9-2014)

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

1 Introduction: Model selection

In empirical research, we almost always want to assess how strongly the data supports one hypothesis compared to another. The many statistical tests that you learn about in a statistics courses are designed to help us quantify the strength of evidence. Typically, such tests ask us to define a null hypothesis and one or a handful of alternative hypotheses.

However, for some questions – which includes some asked in artificial language learning studies – it is difficult to restrict the number of hypotheses to just a handful. If there are very many (or even infinitely many) reasonable hypotheses, we can sometimes resort to *model selection*, where we formalize the hypotheses as (classes of) models and compute for a each a measure of how well the model fits the data.

There are many ways of doing model selection, and many ways to quantify the goodness of fit, but a key ingredient in many approaches (at least those that use probabilistic models) is to calculate the likelihood of the data under a specific model. In today's computer lab we will look at how likelihood can be used to select the best HMM for observed animal songs or natural language data.

2 A simple example

We will again use the `HMM` library in `R`. You might have to install it again (with `install.packages("HMM")`), but will in any case have to load it again:

```
library(HMM)
```

Now define three slight different transmission matrices:

```
tm1 <- t(matrix(c(0.5,0.5,0,0.4,0.5,0.1,0.1,0.4,0.5),3,3))
tm2 <- t(matrix(c(0.4,0.6,0,0.4,0.5,0.1,0.1,0.4,0.5),3,3))
tm3 <- t(matrix(c(0.6,0.4,0,0.4,0.5,0.1,0.1,0.4,0.5),3,3))
```

And use them to define three HMMs:

```
hmm1 <- initHMM(c("A","B","C"), c("a","b","c"), startProbs=c(1,0,0),
   transProbs=tm1, emissionProbs=diag(3))
hmm2 <- initHMM(c("A","B","C"), c("a","b","c"), startProbs=c(1,0,0),
   transProbs=tm2, emissionProbs=diag(3))
hmm3 <- initHMM(c("A","B","C"), c("a","b","c"), startProbs=c(1,0,0),
   transProbs=tm3, emissionProbs=diag(3))
```

Question 1 *What is the difference between these HMMs and how will that difference show up in the strings it generates?*

Run the HMMs (e.g., with `simHMM(hmm1,10)`) to check your answer.

As we saw last week, with `forward()` we can compute the likelihood of a sequence of observations (The forward function really computes at each point along the sequence, and for each possible state, the probability of the observations up to that point × the probability of being in that state; the likelihood of the whole sequence is therefore the sum of probabilities in the final time step).

Using `simHMM()` to generate data with one of our three models, and `forward()` to compute the likelihood of that data, we can check whether the real model indeed gives the highest likelihood to the data (note that we need `exp()` to turn log probabilities into normal probabilities).

```
obs = simHMM(hmm1,100)$observation
sum(exp(forward(hmm1,obs)[,100]))
sum(exp(forward(hmm2,obs)[,100]))
sum(exp(forward(hmm3,obs)[,100]))
```

Question 2 *What are the likelihoods under each of the three models, and which one is the highest? Is that the correct model?*

3 Homework

Question 3 *Define five HMMs with the same 5 observable symbols each, that have different emission probability matrices. Generate data from one of them, and report on the likelihoods that each of the 5 HMMs assigns to the data. Which one gives the highest likelihood? Which one is the runner up? Do the same experiment several times with very little data - how often does the likelihood criterion lead you to the wrong answer?*

Question 4 *Give an HMM that can generate the set of strings that always starts with an arbitrary number of a's, then continues with an arbitrary number of b's and then ends (include START and END symbols). This is the formal language $a^n b^m$, with $n \geq 1$ and $m \geq 1$.*

Question 5 *Give a context-free grammar that can generate $a^n b^n$ (where the number of a's must be equal to the number of b's). Hint: you need a* recursive *rule, where the symbol from the left-hand side also appears on the right-hand side.*

Question 6 *Some researchers have criticized Chomsky's theories for being unfalsifiable*[1]*. Do you agree? Give arguments for your position (maximum 1/2 page).*

[1] Note: falsifiability is a concept from philosophy of science, and generally seen as a requirement for scientific theories; a theory is *falsifiable* if it is possible, in principle, to disprove the theory. That something is falsifiable does not mean it is false; rather, that if it is false, then this can be shown by observation or experiment.