Coding Assignment 3

Practical Statistical Learning – CS598

Adrian Bandolon

November 13, 2018

Introduction

- For Assignment 3, we were asked to implement the ***Baum-Welch Algorithm*** and the ***Viterbi Algorithm*** for a ***Hidden Markov model (HMM)*** with *two hidden states (A or B)* and whose outcome is a discrete random variable taking *three unique values (1, 2, 3)*.

Baum-Welch Algorithm

- The `forward.prob`, `backward.prob`, `myBW` and `BWonestep` functions were provided. We were asked to write the code that updates `myGamma` in the `BWonestep` function.
  - I used three `for` loops to compute `myGamma`.

```
for (i in 1:mz) {
  for (j in 1:mz) {
    for (t in 1:(T - 1)) {
      myGamma[i, j, t] = alp[t, i] * A[i, j] * B[j, x[t + 1]] * beta[t + 1, j]
      }
    }
  }
```

Chunk 1. Code used for `myGamma` computation inside the `BWonestep` function

- Below is the ***Estimated Transition Matrix (*** $A_{2\times2}$ ***)*** and the ***Estimated Emission Matrix (*** $B_{2\times3}$ ***)*** from the Baum-Welch Algorithm after 100 iterations.

	A	B
A	0.53816345	0.46183655
B	0.48664443	0.51335557

Table 1: Estimated Transition Matrix (A) from the Baum-Welch Algorithm after 100 Iterations

	1	2	3
A	0.16277513	0.26258073	0.57464414
B	0.25149960	0.27780971	0.47069069

Table 2: Estimated Emission Matrix (B) from the Baum-Welch Algorithm after 100 Iterations

Viterbi Algorithm

- The Viterbi Algorithm was implemented in the `myViterbi` function. We were required to write the whole function. The output of `myViterbi` (`Coding3_HMM_Viterbi_Output.txt`) will be compared with the output from the `viterbi` function (`Coding3_HMM_True_Viterbi_Output.txt`) from the `HMM` library. `Coding3_HMM_Viterbi_Output.txt` will be written in the directory where this `*.rmd` file is executed.
- For my implementation fo the Viterbi Algorithm, I modified the `viterbi` function from the `HMM` library to accomodate the data provided and the output from the Baum-Welch Algorithm above as inputs . Source code for the `HMM` library is **here**.
- The inputs for the `myViterbi` function are `(x, A, B, w)`, where `x` is the data, `A` is a 2-by-2 matrix from the Baum-Welch Algorithm as implemented in the previous section, `B` is a 2-by-3 matrix also from the previous section, and `w` is the initial distribution matrix.
- The code below compares `Coding3_HMM_Viterbi_Output.txt` against `Coding3_HMM_True_Viterbi_Output.txt`.

```
## from [Coding3_HMM.html]

library(HMM)

data <- read.csv('Coding3_HMM_Data.csv')

mz = 2; mx = 3
ini.A = matrix(1, mz, mz)
ini.A = ini.A / rowSums(ini.A)
ini.B = matrix(1:6, mz, mx)
ini.B = ini.B / rowSums(ini.B)
ini.w = c(1 / 2, 1 / 2)

hmm0 =initHMM(c("A", "B"), c(1, 2, 3),
              startProbs = ini.w,
              transProbs = ini.A, emissionProbs = ini.B)

true.out = baumWelch(hmm0, data$X, maxIterations=100, pseudoCount=0)

true.viterbi = viterbi(true.out$hmm, data$X)

# myout.Z [1:500] is from the previous section
# it holds the output from the myViterbi function
sum(true.viterbi != myout.Z)
```

```
## [1] 0
```