

# Discrete Hidden Markov Chain Inference Test

Group Four

April / 27 / 2018

## I. Package Introduction

This is a R markdown document for our package mydHMM. We will give a quick intro to our major functions and give a example in the later part II, and the result are showing in the part III.

*dhmm.model*: initialize a random hidden markov model

*dhmm.sim*: simulate states and observations from a given Hidden Markov Model

*dhmm.estimate*: estimate all parameters of hmm given observations

*dhmm.viterbi*: find the max possible hidden state sequence given a HMM *dhmm.decoding*: return estimate model and max possible hidden state sequence given observations

The following code automatically generated samples from a given hidden markov model

```
size <- 400
model <- dhmm.model(4, 'count')
model[[3]] = c(0, 3, 5, 7)
model[[2]] <- matrix(c(0.3, 0.2, 0.3, 0.2,
                      0.2, 0.2, 0.5, 0.1,
                      0.3, 0.3, 0.1, 0.3,
                      0.2, 0.2, 0.3, 0.3), 4, 4, byrow = T)

sim_samples <- dhmm.sim(model, size)
obser <- sim_samples[[2]]
show(obser[1:50])

## [1] 0.2784794 8.1381281 5.0023971 -2.2770674 5.0128678 7.0451306
## [7] 0.2240398 5.0102969 7.9263463 6.2745437 5.0098113 3.0123775
## [13] 0.9938383 0.3067492 0.2456500 0.2588475 0.6521741 -1.0228817
## [19] 3.3701953 4.9868149 -0.2013181 5.0087611 4.9921081 -1.1215034
## [25] 4.9943783 3.1719298 5.0202903 6.5660756 5.0185580 6.1987383
## [31] 5.0497751 0.3697323 7.5222593 4.9911386 2.5155890 0.4570862
## [37] 7.2485099 6.2526517 3.0017892 5.0046357 6.1279126 2.2701657
## [43] -0.2672933 2.9280228 -0.8592775 0.7745651 5.0076587 4.9925600
## [49] 2.8365535 5.0421469
```

## II. Examples

We can use the generated samples in the first part to test our code, the result will return a list contain:

**best\_estimate model**: a parameter set of hmm, include transition matrix and mean and sd vector

**decoding sequence**: the estimated sequence of hidden state of hmm

**likelihood list**: each element in the list represent a sequence of log-likelihood value in a full iteration

```
epsilon <- 0.00001
max_iter <- 1000
range <- c(4, 5)
result <- dhmm.decoding(obser, range, epsilon, max_iter)
# show estimated hidden markov model
result[[1]]
```

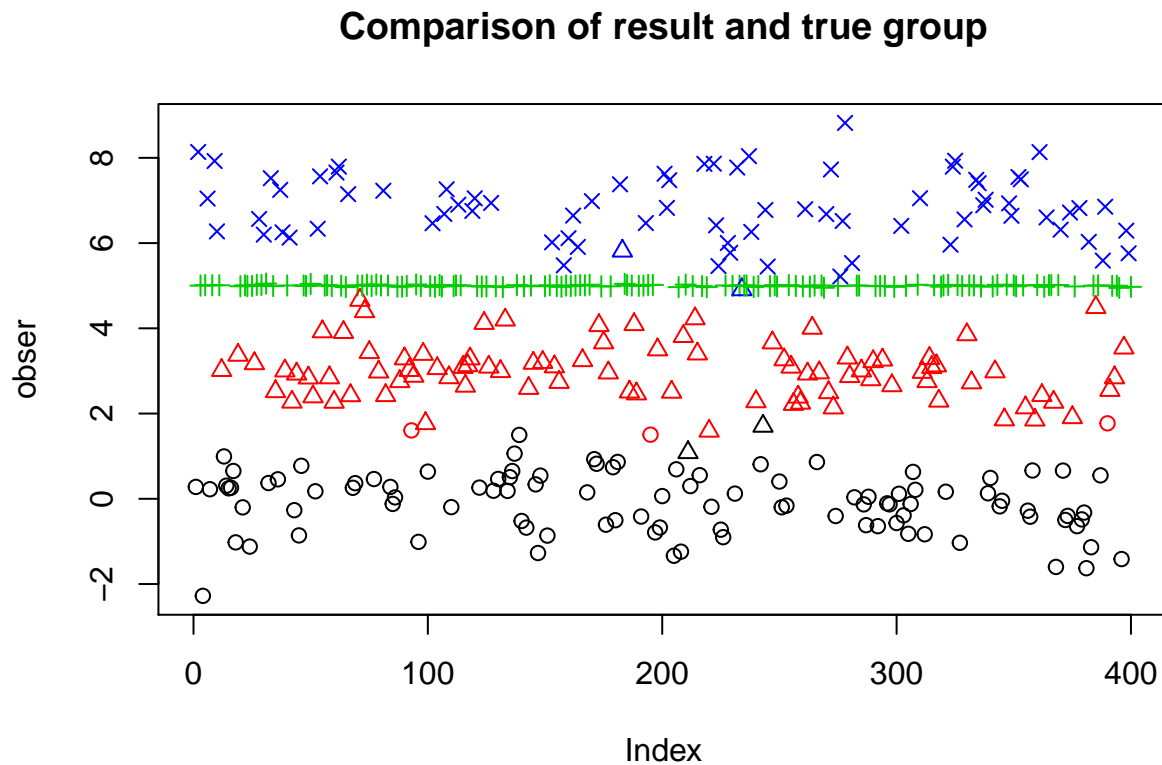
```
## $states
## [1] 1 2 3 4
##
## $`transition matrix`
##      [,1]      [,2]      [,3]      [,4]
## [1,] 0.3755787 0.1708999 0.3225764 0.13094496
## [2,] 0.2326075 0.1564978 0.5414754 0.06941207
## [3,] 0.2574255 0.3518085 0.1075582 0.27520211
## [4,] 0.1320090 0.2154713 0.3412508 0.31120091
##
## $mean
## [1] -0.04361348  2.96857645  4.99926502  6.77360834
##
## $sd
## [1] 0.7386590 0.6785593 0.0170941 0.8131395
```

### III. Results

Furthermore, we can visualize our results, 1).contrast of result and true classification and 2).the converge curve of log-likelihood

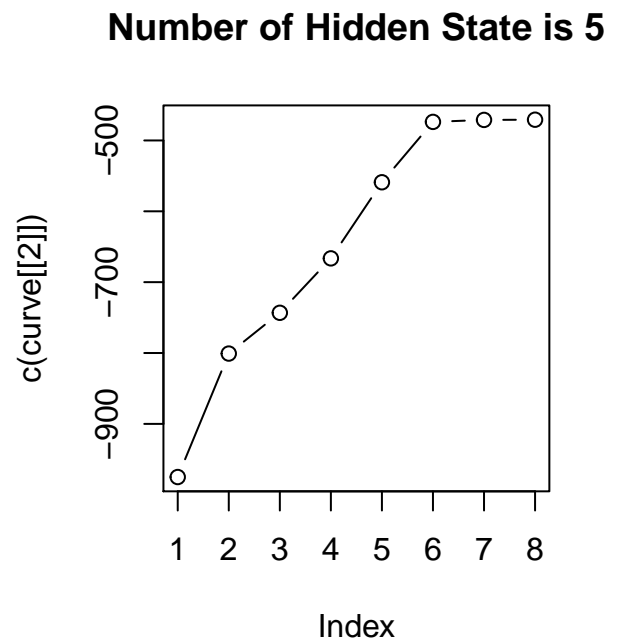
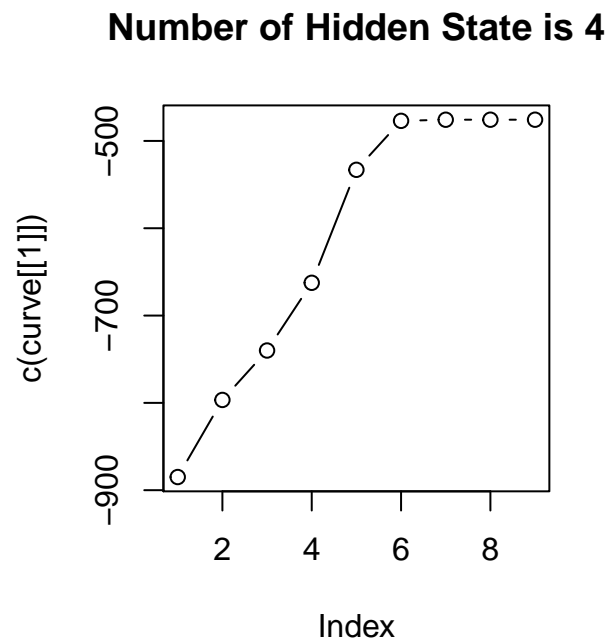
In the first plot, colors represent the predicted groups, while symbols represent the true groups

```
plot(obser, pch=sim_samples[[1]], col=result[[2]])
title("Comparison of result and true group")
```



The converge curve of log-likelihood for different parameters (number of hidden states)

```
# converge curve of log-likelihood
curve <- result[[3]]
par(mfrow=c(1,2), pty="s")
plot(c(curve[[1]]), type="b")
title("Number of Hidden State is 4")
plot(c(curve[[2]]), type="b")
title("Number of Hidden State is 5")
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.