

Main

Arnav Bhutani

12/6/2020

A non-commented version of this file is available here

Initial Library Setup

variable initialization.

```
hidden = matrix(1/2, 2, 2)
visible = matrix(1:6, 2, 3)
visible = visible/rowSums(visible)
initial_distribution = c(1/2, 1/2)
```

Get the actual hidden and visible probabilities, as defined by the dataset.

```
output = BaumWelch(data$Visible, hidden, visible, initial_distribution, n.iter = 100)
```

```
output$switching
```

```
##           [,1]      [,2]
## [1,] 0.5381634 0.4618366
## [2,] 0.4866444 0.5133556
```

```
output$visible
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.1627751 0.2625807 0.5746441
## [2,] 0.2514996 0.2778097 0.4706907
```

```
write_rds(output, here("/data/hidden_visible.rds"))
```

Get the HMM library's output

```
hmm = initHMM(c("A", "B"), c(1, 2, 3),
              startProbs = initial_distribution,
              transProbs = hidden,
              emissionProbs = visible)
```

```
hmm.out <- baumWelch(hmm, data$Visible, 100)
```

```
hmm.out$hmm$transProbs
```

```
##      to
## from      A      B
##      A 0.5381634 0.4618366
##      B 0.4866444 0.5133556
```

```
hmm.out$hmm$emissionProbs
```

```
##      symbols
## states      1      2      3
##      A 0.1627751 0.2625807 0.5746441
##      B 0.2514996 0.2778097 0.4706907
```

```
write_rds(hmm.out$hmm, here("/data/hmm_hidden_visible.rds"))
```

Create a 100 tests for each viterbi algorithm to find the most probable paths using the HMM package.

```
test <- (map(seq(100, 10000, 100), ~ simHMM(hmm.out$hmm, .x)))
test[[1]]
```

```
## $states
## [1] "A" "B" "A" "A" "A" "A" "B" "B" "B" "A" "B" "A" "B" "B" "A" "B" "B" "B"
## [19] "B" "B" "B" "B" "A" "A" "B" "B" "A" "B" "A" "A" "A" "B" "B" "B" "B" "A"
## [37] "A" "B" "B" "B" "A" "A" "B" "B" "B" "B" "B" "B" "B" "A" "A" "B" "A"
## [55] "B" "B" "A" "B" "B" "A" "A" "A" "A" "B" "B" "B" "A" "B" "B" "B" "B" "B"
## [73] "A" "B" "A" "A" "B" "A" "B" "B" "A" "A" "A" "B" "B" "A" "B" "A" "A" "B"
## [91] "B" "B" "A" "A" "A" "B" "A" "A" "B" "A"
##
## $observation
## [1] 1 2 3 3 3 3 3 3 3 1 3 3 3 2 2 2 3 2 3 3 1 3 1 2 3 2 1 1 2 3 3 3 1 2 3
## [38] 1 2 2 3 2 3 3 3 3 2 1 3 3 3 3 2 2 3 3 3 3 3 2 3 2 3 3 3 2 3 3 3 1 3 1 3 3
## [75] 3 3 2 3 3 3 1 1 3 3 2 3 3 2 3 3 3 2 3 3 3 3 2 3 3 3
```

Accuracy check for my model

```
my_accuracy <- map(1:length(test), ~
  sum(Viterbi(test[[]]$observation, output$switching, output$visible, initial_distribution)
)
my_accuracy[[1]]
```

```
## [1] 0.52
```

```
write_rds(my_accuracy, here("/data/my_accuracy.rds"))
```

Similar accuracy check for HMM model

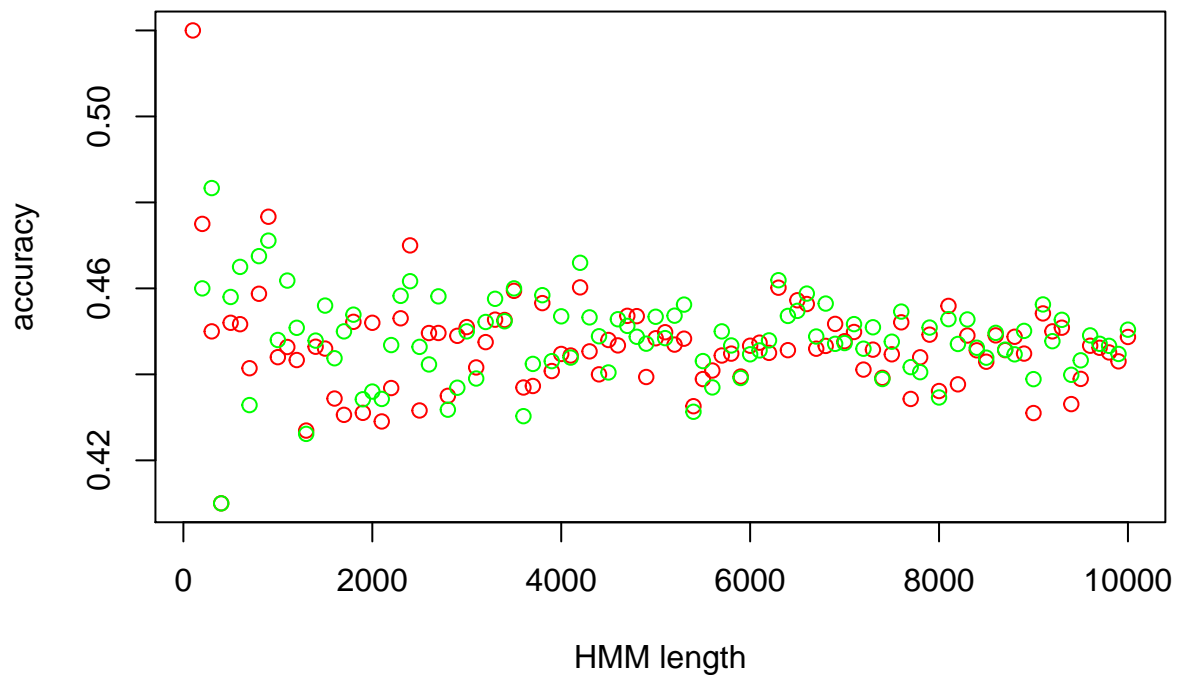
```
hmm_accuracy <- map(1:length(test), ~
  sum(viterbi(hmm, test[[]]$observation) != test[[]]$states)/length(test[[]]$sta
hmm_accuracy[[1]]
```

```
## [1] 0.54
```

```
write_rds(hmm_accuracy, here("/data/hmm_accuracy.rds"))
```

Create a plot of the two accuracies, and save it in the results folder.

```
plot(seq(100, 10000, 100), my_accuracy, xlab = "HMM length", ylab = "accuracy", col = "red")
points(seq(100, 10000, 100), hmm_accuracy, col = "green")
```



```
dev.copy(png, here('/results/accuracy.png'))
```

```
## png
## 3
```

```
dev.off()
```

```
## pdf
## 2
```

Time the various functions:

```
timings <- bench::mark(
  {
    Viterbi(data$Visible, output$switching, output$visible, initial_distribution)
  },
```

```
{
  Viterbi_c(data$Visible,output$switching,output$visible,initial_distribution)
},
{
  viterbi(hmm.out$hmm, data$Visible)
}
)
```

```
## Warning: No function found for Rcpp::export attribute at file30b81e1244e.cpp:5
```

```
timings
```

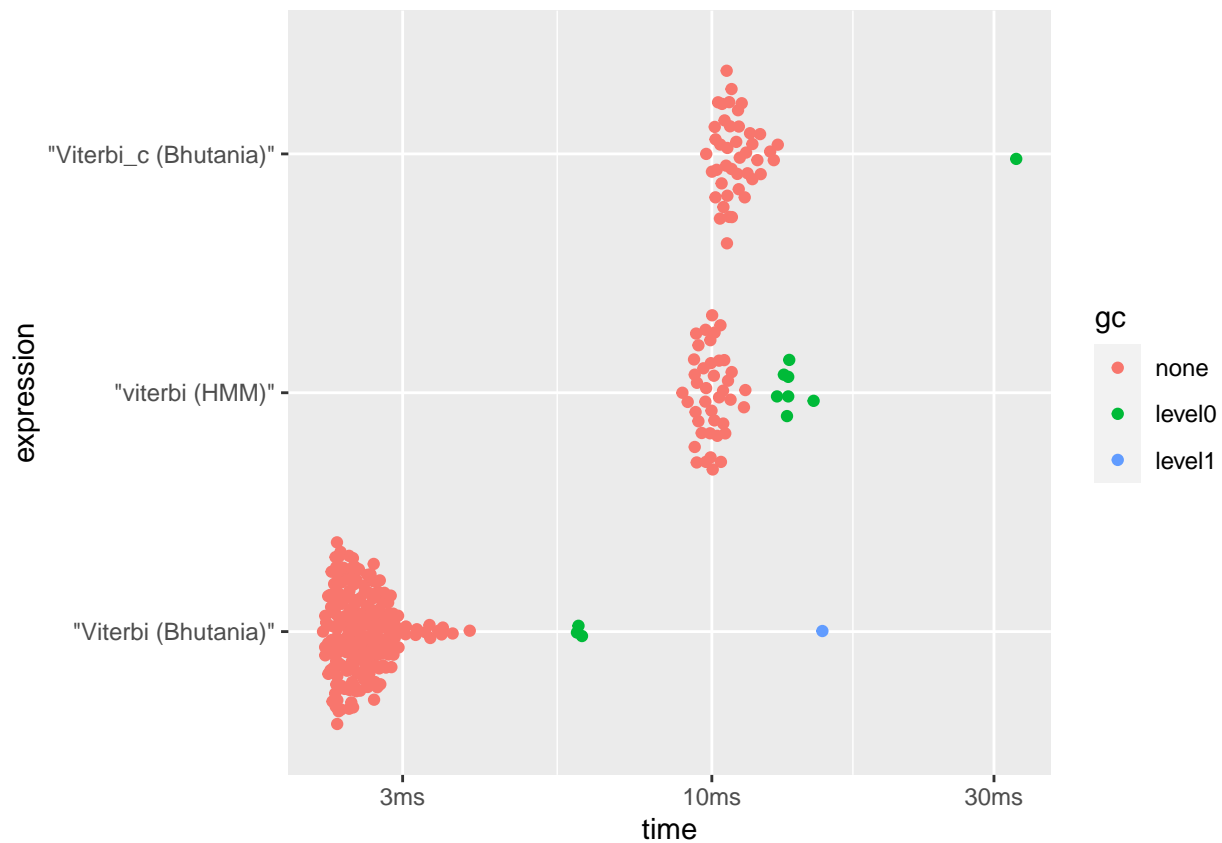
```
## # A tibble: 3 x 6
## # ... with 6 more variables: expression <bch:expr>, min <bch:tm>,
## #   median <bch:tm>, 'itr/sec' <dbl>, mem_alloc <bch:byt>, 'gc/sec' <dbl>
```

```
write_rds(timings, here("/data/timings.rds"))
```

Name the timings and plot:

```
timings$expression[1] <- "Viterbi (Bhutania)"
timings$expression[2] <- "Viterbi_c (Bhutania)"
timings$expression[3] <- "viterbi (HMM)"

plot(timings, type = "beeswarm")
```



```
dev.copy(png, here('/results/timing.png'))
```

```
## png  
## 3
```

```
dev.off()
```

```
## pdf  
## 2
```