Exploring Hidden Markov Models for U.S. Presidents

Jaucelyn Canfield

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#library(tidyverse)
library(HMM)
```

In my project, I looked at United States GDP, crime, and presidential data from 1960 to present day to create two different Hidden Markov Models. In each model, I treated the presidential party (Republican or Democrat) of the president as the hidden states. In the first model, the GDP growth at the end of the given year was the observed state, and in the other model, the crime rate was the observed state.

```
#read in presidential data
presidents <- read.csv("/Users/jaucelyncanfield/Downloads/presidents.csv")
presidents_edit <- presidents[175:235,] #selecting from 1960 on
head(presidents_edit)
```

```
##     Years..after.inauguration.         President    Party
## 175                       1961   John F. Kennedy Democrat
## 176                       1962   John F. Kennedy Democrat
## 177                       1963 Lyndon B. Johnson Democrat
## 178                       1964 Lyndon B. Johnson Democrat
## 179                       1965 Lyndon B. Johnson Democrat
## 180                       1966 Lyndon B. Johnson Democrat
```

```
#read in GDP data
gdp <- read.csv("/Users/jaucelyncanfield/Documents/gdpdata.csv")
gdp_clean <- gdp[18:78,]
colnames(gdp_clean) <- c("Date", "GDP", "per capita", "growth")
gdp_clean$growth <- as.numeric(gdp_clean$growth)
head(gdp_clean)
```

```
##          Date   GDP per capita growth
## 18 1961-12-31 563.3  3066.5629    2.3
## 19 1962-12-31 605.1  3243.8431    6.1
## 20 1963-12-31 638.6  3374.5152    4.4
## 21 1964-12-31 685.8  3573.9412    5.8
## 22 1965-12-31 743.7  3827.5271    6.4
## 23 1966-12-31   815  4146.3166    6.5
```

```
mean(gdp_clean$growth)
```

```
## [1] 2.970995
```

I had to make the GDP growth a categorical variable in order to use it as an observed state. I decided to have three different categories: decrease, small increase, and big increase. Since the average growth was around 3, I decided to make that the cutoff between a small and big increase.

```
#create categorical variable for growth
growth_categ <- c()
for(i in 1:nrow(gdp_clean)){
  if(gdp_clean[i, 4] < 0){ #growth is 4th column
    growth_categ[i] <- "decrease"
  }else if(gdp_clean[i, 4] < 3 & gdp_clean[i,4] > 0){
    growth_categ[i] <- "small increase"
  }else{
    growth_categ[i] <- "big increase"
  }
}
```

```
#combine to one dataset
data <-cbind(presidents_edit,growth_categ)
```

```
#build emission probabilities

#filter by party
democrat <- data %>% filter(Party == "Democrat")
republican <- data %>% filter(Party == "Republican")

#calculate probability of decrease, small/big increase given democrat president
dem_decrease <- sum(democrat$growth_categ == "decrease") /nrow(democrat)
dem_smallincrease <- sum(democrat$growth_categ == "small increase") /nrow(democrat)
dem_bigincrease <-sum(democrat$growth_categ == "big increase") /nrow(democrat)

#calculate probability of decrease, small/big increase given republican president
rep_decrease <- sum(republican$growth_categ == "decrease") /nrow(republican)
rep_smallincrease <- sum(republican$growth_categ == "small increase") /nrow(republican)
rep_bigincrease <- sum(republican$growth_categ == "big increase") /nrow(republican)
```

```
#create emission probability matrix
emit_p <- matrix(c(dem_decrease,dem_smallincrease, dem_bigincrease,
                   rep_decrease,rep_smallincrease,rep_bigincrease),nrow=2,ncol=3,byrow=TRUE)
colnames(emit_p) <- c("Decrease", "Small Increase", "Big Increase")
rownames(emit_p) <- c("Democrat", "Republican")

emit_p
```

```
##              Decrease Small Increase Big Increase
## Democrat   0.06896552      0.3793103    0.5517241
## Republican 0.18750000      0.3437500    0.4687500
```

```
#check rows sum to 1
rowSums(emit_p)
```

```
##   Democrat Republican
##          1          1
```

```
#build transition probabilities
demtorep <- 0
demtodem <- 0
reptodem <- 0
reptorep <- 0

#track each instance party switches or stays the same
for(i in 1:(nrow(data)-1)){
  if(data[i,3] == "Democrat"){
    if(data[i+1,3] == "Republican"){
      demtorep <- demtorep + 1
    }else if(data[i+1,3] == "Democrat"){
      demtodem <- demtodem + 1
    }
  }else if(data[i,3] == "Republican"){
    if(data[i+1,3] == "Democrat"){
      reptodem <- reptodem + 1
    }else if(data[i+1,3] == "Republican"){
      reptorep <- reptorep + 1
    }
  }
}

#calculate hidden state transition probabilities
prob_demtodem <- demtodem/(demtodem + demtorep)
prob_demtorep <- demtorep/(demtodem + demtorep)
prob_reptodem <- reptodem/(reptodem + reptorep)
prob_reptorep <- reptorep/(reptodem + reptorep)
trans_p <- matrix(c(prob_demtodem,prob_demtorep,
                    prob_reptodem,prob_reptorep),nrow=2,ncol=2,byrow=TRUE)
colnames(trans_p) <- c("Democrat", "Republican")
rownames(trans_p) <- c("Democrat", "Republican")
trans_p
```

```
##             Democrat Republican
## Democrat   0.8571429  0.1428571
## Republican 0.1250000  0.8750000
```

```
rowSums(trans_p)
```

```
##   Democrat Republican
##          1          1
```

At this point I realized something might get thrown off by the fact that you will always observe the same party for at least 4 years in a row. Stay tuned for what ends up happening.

```
#build hidden markov model
States <- c("Democrat","Republican")
Symbols <- c("decrease","small increase", "big increase")


start_p <- c(nrow(democrat)/nrow(data),nrow(republican)/nrow(data))
#start_p <- c(0.5,0.5)
#start_p <- c(1,0)
#start_p <- c(0.7,0.3)

hmm = initHMM(States, Symbols, start_p, trans_p, emit_p)
viterbi(hmm, growth_categ)
```

```
##  [1] "Republican" "Republican" "Republican" "Republican" "Republican"
##  [6] "Republican" "Republican" "Republican" "Republican" "Republican"
## [11] "Republican" "Republican" "Republican" "Republican" "Republican"
## [16] "Republican" "Republican" "Republican" "Republican" "Republican"
## [21] "Republican" "Republican" "Republican" "Republican" "Republican"
## [26] "Republican" "Republican" "Republican" "Republican" "Republican"
## [31] "Republican" "Republican" "Republican" "Republican" "Republican"
## [36] "Republican" "Republican" "Republican" "Republican" "Republican"
## [41] "Republican" "Republican" "Republican" "Republican" "Republican"
## [46] "Republican" "Republican" "Republican" "Republican" "Republican"
## [51] "Republican" "Republican" "Republican" "Republican" "Republican"
## [56] "Republican" "Republican" "Republican" "Republican" "Republican"
## [61] "Republican"
```

Basically the viterbi algorithm predicts that the most likely path of hidden states is republican in each year. I messed around with a few different starting probability splits to see if I could get this to change. An equal split gives the same outcome. So I tried making the starting probability for being a democrat the highest it could possibly be and I got the following output:

```
start_p <- c(1,0)
hmm = initHMM(States, Symbols, start_p, trans_p, emit_p)
viterbi(hmm, growth_categ)
```

```
##  [1] "Democrat"   "Democrat"   "Democrat"   "Democrat"   "Democrat"
##  [6] "Democrat"   "Democrat"   "Democrat"   "Democrat"   "Republican"
## [11] "Republican" "Republican" "Republican" "Republican" "Republican"
## [16] "Republican" "Republican" "Republican" "Republican" "Republican"
## [21] "Republican" "Republican" "Republican" "Republican" "Republican"
## [26] "Republican" "Republican" "Republican" "Republican" "Republican"
## [31] "Republican" "Republican" "Republican" "Republican" "Republican"
## [36] "Republican" "Republican" "Republican" "Republican" "Republican"
## [41] "Republican" "Republican" "Republican" "Republican" "Republican"
## [46] "Republican" "Republican" "Republican" "Republican" "Republican"
## [51] "Republican" "Republican" "Republican" "Republican" "Republican"
## [56] "Republican" "Republican" "Republican" "Republican" "Republican"
## [61] "Republican"
```

It stays Democrat for 9 steps before becoming Republican for the rest. It's almost as if being Republican is an absorbing state, but I am not familiar with how this works with Hidden Markov Models. This is the same output if you say the starting probability for being a Democrat is 0.7 or higher; anything lower results in all Republican hidden states.

I decided to see if I could fix this by addressing the problem I identified earlier. I decided to only consider unique presidents, so I didnt include every year. This cut the observations down significantly, but I just wanted to see what would happen. Since each president would have a few potential values for GDP growth, I decided to pick whichever observation had their maximum growth. Consequently, there was never a decrease so I just had two categorical variables: a small increase and a big increase.

```
#create new data for only unique presidents
data2 <- cbind(data,gdp_clean$growth)
data2 <- data2 %>% group_by(President) %>% mutate(max_gdp = max(`gdp_clean$growth`))
#only keep unique presidents
new_data <- data2 %>% distinct(President,.keep_all = TRUE)
#create new categorical variable
new_data <- new_data %>% mutate(growth_categ = ifelse(max_gdp > mean(new_data$max_gdp),"big", "small"))
```

The rest of the code for creating the HMM is similar to before. It produced the following emission probability matrix and transition probability matrix:

```
## [1] "Emission Probability Matrix"

##                Small       Big
## Democrat   0.3333333 0.6666667
## Republican 0.5000000 0.5000000

## [1] "Transition Probability Matrix"

##             Democrat Republican
## Democrat   0.2000000  0.8000000
## Republican 0.6666667  0.3333333
```

Now, building the HMM and using the Viterbi algorithm:

```
States <- c("Democrat","Republican")
Symbols <- c("big", "small")

start_p <- c(0.5,0.5)
#start_p <- c(nrow(democrat)/nrow(data),nrow(republican)/nrow(data))

hmm2 = initHMM(States, Symbols, start_p, trans_p, emit_p)

print(viterbi2 <- viterbi(hmm2, new_data$growth_categ))
```

```
##  [1] "Democrat"   "Republican" "Democrat"   "Republican" "Democrat"
##  [6] "Republican" "Democrat"   "Republican" "Democrat"   "Republican"
## [11] "Democrat"   "Republican"
```

Now it alternates between being Republican and Democrat. This is probably due to the fact that transition probability matrix has much higher probabilities of switching parties than before. The Viterbi algorithm appears to be quite sensitive to both the transition probability matrix and the intial state distribution.

In the first attempt at making this HMM model, I meant to calculate how often the predicted Viterbi path deviated from the actual path of hidden states. Given the outcome that the algorithm only produced "Republican" for the most part, there didn't seem like a point. But now that it does change frequently,

```
misclassification <- sum(viterbi2 != new_data$Party) / length(viterbi2)
misclassification
```

```
## [1] 0.6666667
```

That's pretty high, but this wasn't the best type of data, nor probably the best variable to use for an observed state clearly.

Then, I made another HMM where the crime rate was the observed state. To make it a categorical variable I considered if there was an increase or a decrease in the crime rate in a given year from the year before.

```
#read in crime data
crime <- read.csv("/Users/jaucelyncanfield/Downloads/crime.csv")
head(crime)
```

```
##   Year  Population     Total   Rate
## 1 1960 179,323,175 3,384,200 0.0189
## 2 1961 182,992,000 3,488,000 0.0191
## 3 1962 185,771,000 3,752,200 0.0202
## 4 1963 188,483,000 4,109,500 0.0218
## 5 1964 191,141,000 4,564,600 0.0239
## 6 1965 193,526,000 4,739,400 0.0245
```

```
#turn crime rate into categorical variable of decrease or increase
change <- c()
for(i in 2:nrow(crime)){
  change[i-1] <- ifelse(crime[i-1,4] > crime[i,4],"decrease","increase")
}
```

```
#crime data is only up to 2019, so I have to make
#a new presidents dataset
presidents2 <- presidents[175:233,]
#combine to one dataset
pres_crime <- cbind(presidents2,change)
```

Originally, I went back to including every year from 1960 on in the calculations (so the same party would be in office at least four years in a row) to see if it would give the same results of it always being a Republican. It does. So, to have unique presidents again, I only included observations at the end of each of their terms.

```
#only include last year president was in office in the data set
pres_crime <- pres_crime %>% group_by(President) %>% mutate(Year = max(Years..after.inauguration.))
pres_crime <- pres_crime %>% filter(Years..after.inauguration. == Year)
```

The code for calculating the rest is the same as with the other HMM. It produced the following emission probability matrix and transition probability matrix:

```
## [1] "Emission Transition Probability Matrix"
```

```
##             Decrease  Increase
## Democrat   0.4000000 0.6000000
## Republican 0.6666667 0.3333333
```

```
## [1] "Transition Probability Matrix"
```

```
##            Democrat Republican
## Democrat        0.2        0.8
## Republican      0.6        0.4
```

And creating the HMM and using the Viterbi algorithm:

```
States <- c("Democrat","Republican")
Symbols <- c("decrease", "increase")

start_p <- c(0.5,0.5)
#start_p <- c(nrow(democrat)/nrow(data),nrow(republican)/nrow(data))

hmm4 = initHMM(States, Symbols, start_p, trans_p, emit_p)

print(viterbi4 <- viterbi(hmm4, pres_crime$change))
```

```
##  [1] "Democrat"   "Republican" "Democrat"   "Republican" "Democrat"
##  [6] "Republican" "Democrat"   "Republican" "Democrat"   "Republican"
## [11] "Republican"
```

```
misclassification <- sum(viterbi4 != pres_crime$Party) / length(viterbi4)
paste("misclassification:",misclassification)
```

```
## [1] "misclassification: 0.545454545454545"
```

This is a little better than the model with GDP growth, but still way too high.
A better variable to treat as the observed state would be interesting. Preferably something that is already categorical, I just couldn't think of one.