Traditional Markov Chain

Ricky Trujillo

2024-05-09

```
weather_summary = read.csv("weather.csv")
weather_summary$weather_state = as.factor(weather_summary$weather_state)

states <- c("Chilly", "Cloudy", "Partly Cloudy", "Rainy", "Sunny")
weather_summary$prediction<- rep(NA,nrow(weather_summary))
weather_summary$prediction = factor(weather_summary$prediction, levels= states)


weather24_pred = weather_summary %>%
  filter(Date>= as.Date("2024-01-01", "%Y-%m-%d"))%>%
  select(Date, Month, weather_state, prediction)

state_mat = one_hot(as.data.table(weather_summary$weather_state))
colnames(state_mat)<-  states
```

```
transition_mat = createSequenceMatrix(weather_summary %>%
                                        filter(Date < as.Date("2024-01-01","%Y-%m-%d")) %>%
                                        select(weather_state), toRowProbs = TRUE)
transition_mat = as.matrix(transition_mat, nrow=5, ncol=5)
t(round(transition_mat,2))
```

```
##               Chilly Cloudy Partly Cloudy Rainy Sunny
## Chilly          0.69   0.24          0.02  0.43  0.07
## Cloudy          0.16   0.50          0.35  0.26  0.10
## Partly Cloudy   0.00   0.08          0.19  0.09  0.09
## Rainy           0.03   0.06          0.00  0.22  0.01
## Sunny           0.12   0.12          0.44  0.00  0.74
```

```
# rowSums(transition_mat)
```

```
# Function that computes the following state given the one your are in
initial_state <- c(1,0,0,0,0)
days = 1000
final_state = initial_state %*% (transition_mat%^%days)
final_state
```

```
##        Chilly    Cloudy Partly Cloudy      Rainy    Sunny
## [1,] 0.310162 0.2339417    0.06605848 0.03162474 0.358213
```

```
traditional<- function(weather24_pred){
  prob = list()
  for(i in 1:nrow(weather24_pred)){
    if(i==1){ #special case for first day in 2024
      prob[[i]] = t(transition_mat) %*% t(state_mat[730,]) # state_mat[730,] = one hot encoded represen
```

```
      choice = sample(1:5,1,prob=prob[[i]]) # samples 1:5 from "x(t)" probabilities
      weather24_pred$prediction[i] = states[choice] #references name of 1:5 with states list and stores
    }else{#all other days until April
      #one hot encodes the previous step (i.e. when i=2, it will one hot encode i=1 prediction)
      state_bin = one_hot(as.data.table(factor(weather24_pred$prediction[i-1], levels = states)))
      prob[[i]] = t(transition_mat) %*% t(state_bin) #use one hot encodeded representation in x(t) = P*
      choice = sample(1:5,1,prob=prob[[i]]) # samples 1:5 from "x(t)" probabilities
      weather24_pred$prediction[i] = states[choice] #references name of 1:5 with states list and stores
    }
  }
  return(weather24_pred)
}
```

```
S = 50# Number of Simulations
someData <- rep(NA, 4*5*S)  #(4 Months, 5 States, S simulations)
avg_state <- array(someData, c(4, 5, S)) #3d matrix storing ratio for each simulation S

for(s in 1:S){
  weather24_pred$prediction<- rep(NA,nrow(weather24_pred))
  temp = traditional(weather24_pred)
  temp_avg = temp %>% group_by(Month) %>% mutate(month_len = length(Month)) %>%
    ungroup() %>%
    group_by(Month, prediction) %>%
    reframe(ratio = length(prediction)/month_len)%>%
    unique()
  for(i in 1:nrow(temp_avg)){
    if(temp_avg$Month[i]==1){
      if(temp_avg$prediction[i]=="Chilly"){
        avg_state[1,1,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Cloudy"){
        avg_state[1,2,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Partly Cloudy"){
        avg_state[1,3,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Rainy"){
        avg_state[1,4,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Sunny"){
        avg_state[1,5,s] = temp_avg$ratio[i]
      }
    }else if(temp_avg$Month[i]==2){
      if(temp_avg$prediction[i]=="Chilly"){
        avg_state[2,1,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Cloudy"){
        avg_state[2,2,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Partly Cloudy"){
        avg_state[2,3,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Rainy"){
        avg_state[2,4,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Sunny"){
        avg_state[2,5,s] = temp_avg$ratio[i]
      }
    }else if(temp_avg$Month[i]==3){
      if(temp_avg$prediction[i]=="Chilly"){
        avg_state[3,1,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Cloudy"){
```

```
        avg_state[3,2,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Partly Cloudy"){
        avg_state[3,3,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Rainy"){
        avg_state[3,4,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Sunny"){
        avg_state[3,5,s] = temp_avg$ratio[i]
      }
    }else if(temp_avg$Month[i]==4){
      if(temp_avg$prediction[i]=="Chilly"){
        avg_state[4,1,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Cloudy"){
        avg_state[4,2,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Partly Cloudy"){
        avg_state[4,3,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Rainy"){
        avg_state[4,4,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Sunny"){
        avg_state[4,5,s] = temp_avg$ratio[i]
      }
    }
  }
  avg_state[is.na(avg_state)] <- 0
  }
```

```
someData <- rep(0, 4*5)
sum_mat = array(someData, c(4, 5))
for(s in 1:S){
  sum_mat = sum_mat + avg_state[,,s]
}

sum_mat = sum_mat/S
round(sum_mat,2)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 0.32 0.26 0.06 0.04 0.32
## [2,] 0.29 0.22 0.08 0.03 0.37
## [3,] 0.33 0.24 0.07 0.03 0.33
## [4,] 0.30 0.24 0.06 0.03 0.36
```

```
mse<-function(x_hat,x) rowMeans((x_hat-x)^2)

weather_ratio = matrix(c(0.58064516, 0.25806452, 0.06451613, 0.06451613,0.03225806,
                         0.41379310, 0.34482759, 0, 0.24137931,0,
                         0.54838710, 0.29032258, 0, 0.12903226, 0.03225806,
                         0.6, 0.33333333, 0, 0.03333333, 0.03333333), nrow = 4, ncol = 5, byrow = TRUE)
round(weather_ratio,2)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 0.58 0.26 0.06 0.06 0.03
## [2,] 0.41 0.34 0.00 0.24 0.00
## [3,] 0.55 0.29 0.00 0.13 0.03
## [4,] 0.60 0.33 0.00 0.03 0.03
```

```r
sum(mse(sum_mat, weather_ratio))
```

```
## [1] 0.1454314
```