Matrix Algebra Exercises

Aaron Stearns

3/17/2019

Exercise 1:

Given a customer's shopping list and two different grocery shops with identical items but different prices, which supermarket would be cheaper for the customer to go to to? (Problem taken from the Fast.ai Computational Linear Algebra course)

```
# create the matrices:
customers <- matrix(c(6, 5, 3, 1, 3, 6, 2, 2, 3, 4, 3, 1),
                    nrow = 3,
                    ncol = 4,
                    byrow = TRUE)
colnames(customers) <- c("roll", "bun", "cake", "bread")
rownames(customers) <- c("Customer1", "Customer2", "Customer3")

shops <- matrix(c(1.5, 1, 2, 2.5, 5, 4.5, 16, 17),
                nrow = 4,
                ncol = 2,
                byrow = TRUE)
colnames(shops) <- c("Shop1", "Shop2")
rownames(shops) <- c("roll", "bun", "cake", "bread")

customers
```

```
##           roll bun cake bread
## Customer1    6   5    3     1
## Customer2    3   6    2     2
## Customer3    3   4    3     1
```

```
shops
```

```
##       Shop1 Shop2
## roll    1.5   1.0
## bun     2.0   2.5
## cake    5.0   4.5
## bread  16.0  17.0
```

As a test, I'll first calculate the dot product of a given customer's shopping list and the price list for each shop:

```
dotProd1 <- customers[1,] %*% shops[,1]
dotProd2 <- customers[1,] %*% shops[,2]

print(paste0("The dot product of customer 1 and shop 1 is: ",
             dotProd1))
```

```
## [1] "The dot product of customer 1 and shop 1 is: 50"
```

```
print(paste0("The dot product of customer 1 and shop 2 is: ",
             dotProd2))
```

```
## [1] "The dot product of customer 1 and shop 2 is: 49"
```

The below function calculates the matrix product of 'customers' and 'shops' and returns the name of the shop where each customer would have the lowest overall bill:

```
shopSelector <- function() {
  matrixProduct <- customers %*% shops

  for (eachrow in 1:nrow(matrixProduct)) {

      if (matrixProduct[eachrow, 1] < matrixProduct[eachrow, 2]) {
        print(
          paste0(rownames(customers)[eachrow],
                 " will get a lower price at ",
                 colnames(shops)[1]))
      } else if (matrixProduct[eachrow, 1] > matrixProduct[eachrow, 2]){
        print(
          paste0(rownames(customers)[eachrow],
                 " will get a lower price at ",
                 colnames(shops)[2]))
      } else {
        print("Price is equivalent at both shops.")
      }
   }
}

# Calling the function:
shopSelector()
```

```
## [1] "Customer1 will get a lower price at Shop2"
## [1] "Customer2 will get a lower price at Shop1"
## [1] "Price is equivalent at both shops."
```

Exercise 2:

Given a list of car buyers and their car preferences (speed, style, efficiency) on a scale of 1-5, and given a list of cars with those same characteristics (speed, style, efficiency) on the same scale, return the car that most closely matches each buyer's preferences.

```
# create the matrices:
buyers <- matrix(c(2, 2, 5, 5, 5, 1, 2, 3, 3),
                 nrow = 3,
                 ncol = 3,
                 byrow = TRUE)
colnames(buyers) <- c("speed", "style", "efficiency")
rownames(buyers) <- c("Buyer1", "Buyer2", "Buyer3")

cars <- matrix(c(5, 5, 1, 2, 1, 4, 2, 4, 4),
               nrow = 3,
               ncol = 3,
               byrow = TRUE)
rownames(cars) <- c("BMW M3", "Honda Accord", "Toyota Prius")
colnames(cars) <- c("speed", "style", "efficiency")
```

```r
buyers
```

```
##        speed style efficiency
## Buyer1     2     2          5
## Buyer2     5     5          1
## Buyer3     2     3          3
```

```r
cars
```

```
##              speed style efficiency
## BMW M3           5     5          1
## Honda Accord     2     1          4
## Toyota Prius     2     4          4
```

```r
# Function takes an argument "buyer", which is the
# row index of a given buyer in the "buyers" dataframe
recommendCarPurchase <- function(buyer) {

  # initialize an empty vector
  recommend <- vector()
  # set a return value to be changed in for loop
  returnValue <- 0

  for (car in 1:nrow(cars)) {
    # subtract the car stats from the buyer preference stats
    difference <- unlist(buyers[buyer,]) - unlist(cars[car,])

    # take the absolute value of the difference and sum the values
    recommend <- append(recommend, sum(abs(difference)))

    # find the lowest value (most similar car stats to buyer preference)
    returnValue <- which(recommend == min(recommend), arr.ind=TRUE)
  }

  return(
    paste0("The best car for ",
           row.names(buyers)[buyer],
           " is: ",
           row.names(cars)[returnValue]))

}

# Test the function with a given customer row index:
recommendCarPurchase(3)
```

```
## [1] "The best car for Buyer3 is: Toyota Prius"
```