hw1

n.hwang

June 11, 2019

Description of the Recommender System

This system recommends jobs to job search candidates based on their previous rankings of job recommendations and other similar applicants.

Dataset

I built my own dataset of dimensions 6 x 4, where the rows are applicants or users, and the columns are the job posting items, for a total of 24 data points as shown below.

```
data <- as.data.frame(matrix(nrow = 6, ncol = 4))
names(data) <- c("A","B","C","D")
data$A <- c(1,2,1,1,1,2)
data$B <- c(2,4,2,3,2,4)
data$C <- c(4,5,4,5,4,5)
data$D <- c(3,4,3,3,3,5)
data
```

```
##   A B C D
## 1 1 2 4 3
## 2 2 4 5 4
## 3 1 2 4 3
## 4 1 3 5 3
## 5 1 2 4 3
## 6 2 4 5 5
```

Train and Test Sets

To create a test set out of the dataset above, I hold out 1 item from the data matrix above representing the following user-item combinations (1,A), (2,B), (3,C), and (4,D), where the numbers represent the users and the capital letters the job description items.\

Then, the train set is the complement of the test set on the data set shown above. As a result, we have 20 train set data points, and 4 test set data points.

```
a <- c(NA,2,1,1,1,2)
b <- c(2,NA,2,3,2,4)
c <- c(4,5,NA,5,4,5)
d <- c(3,4,3,NA,3,5)

train <- data.frame(A = a, B = b, C = c, D = d)
train
```

```
##    A  B  C  D
## 1 NA  2  4  3
## 2  2 NA  5  4
## 3  1  2 NA  3
```

```
## 4  1  3  5 NA
## 5  1  2  4  3
## 6  2  4  5  5
```

The raw average rating for every user-item combination in the train set.

Taking the average of the 20 training set data points as shown below yields 3.05.

```
raw_mean_train <- mean(as.matrix(train), na.rm=TRUE)
raw_mean_train
```

```
## [1] 3.05
```

RMSE for raw average

Using the raw average of all the user-item combinations above, I calculate the RMSE for both the train and test sets below. The RMSE for the train is 1.36, and 1.226 for test.

```
## Train set
Rij = 0
for (i in 1:6){
  for (j in 1:4){
    if (!is.na(train[i,j])){
      Rij = Rij + (train[i,j]-raw_mean_train)^2
    }
  }
}
RMSE_train <- sqrt(Rij/20)
RMSE_train
```

```
## [1] 1.359228
```

```
# Test set
Rij = 0
for (i in 1:6){
  for (j in 1:4){
    if (is.na(train[i,j])){
      Rij = Rij + (data[i,j]-raw_mean_train)^2
    }
  }
}
RMSE_test <- sqrt(Rij/4)
RMSE_test
```

```
## [1] 1.225765
```

The bias for each user and item

Taking the mean across all columns (for items), and rows (for users), and subtracting the raw average of 3.05, I obtain the bias as follows. As expected, the vector for the item bias is comprised of 4 items, while the size of the user bias vector is 6.

```
# Bias
bias_item <- colMeans(train, na.rm = TRUE) - raw_mean_train
```

```
bias_user <- rowMeans(train, na.rm = TRUE) - raw_mean_train
bias_item; bias_user
```

```
##     A     B     C     D
## -1.65 -0.45  1.55  0.55
```

```
## [1] -0.0500000  0.6166667 -1.0500000 -0.0500000 -0.5500000  0.9500000
```

Baseline predictors for every user-item combination

I compute the baseline predictor for each user-item combination using the formula *raw average + user bias + item bias* corresponding to the particular combination.

```
predictors <- data
for (i in 1:6){
  for (j in 1:4){
    predictors[i,j] <- raw_mean_train + bias_user[i] + bias_item[j]
  }
}
predictors
```

```
##          A        B        C        D
## 1 1.350000 2.550000 4.550000 3.550000
## 2 2.016667 3.216667 5.216667 4.216667
## 3 0.350000 1.550000 3.550000 2.550000
## 4 1.350000 2.550000 4.550000 3.550000
## 5 0.850000 2.050000 4.050000 3.050000
## 6 2.350000 3.550000 5.550000 4.550000
```

RMSE for baseline predictors

```
## Train
Rij = 0
for (i in 1:6){
  for (j in 1:4){
    if (!is.na(train[i,j])){
      Rij = Rij + (predictors[i,j]-train[i,j])^2
    }
  }
}
RMSE_train_predictor <- sqrt(Rij/20)

## Test
Rij = 0
for (i in 1:6){
  for (j in 1:4){
    if (is.na(train[i,j])){
      Rij = Rij + (predictors[i,j] - data[i,j])^2
    }
  }
}

RMSE_test_predictor <- sqrt(Rij/4)
```

```
RMSE_train_predictor; RMSE_test_predictor
```

```
## [1] 0.4010403
```

```
## [1] 0.5570258
```

Summary

We see below that the baseline predictions improved over the naive raw average method by over 70% for the train set and 55% for the test set.

```
#summary
# % improvement for train
1 - RMSE_train_predictor/RMSE_train
```

```
## [1] 0.7049499
```

```
# % improvement for test
1 - RMSE_test_predictor/RMSE_test
```

```
## [1] 0.5455688
```