Recommending Movies Using a Hybrid Recommendation System

Data 612 Project 2

William Outcault

16 June 2020

Contents

Introduction

This project utilizes two recommendation algorithms to predict movies based on the MovieLense dataset. Four different algorithms will be analyzed in order to determine which hybrid model will provide the best recommendations for future users.

The following libraries are required for this project.

```
library(recommenderlab)
library(ggplot2)
library(tidyverse)
library(purrr)
```

Loading MovieLense Ratings

```
data(MovieLense)
ratings_movies <- MovieLense[rowCounts(MovieLense) > 25, colCounts(MovieLense) > 50]
ratings_movies
```

```
## 799 x 591 rating matrix of class 'realRatingMatrix' with 80045 ratings.
```

Creating Training/Testing Sets

Using recommenderlab's `evaluationScheme` function, 80-20 training/testing datasets are created using a 5-fold cross-validation scheme.

evaluationScheme

```
esCross <- evaluationScheme(ratings_movies,
                            method="cross-validation",
                            train=0.8,
                            k=5,
                            given=-1,
                            goodRating=3)
```

Evaluate Models

Below we begin by listing the algorithms we will be testing for our dataset.

List of Models

```
algorithms <- list(
  "random items" = list(name = "RANDOM", param = NULL),
  "popular items" = list(name = "POPULAR", param = NULL),
```

```
  "item-based CF" = list(name = "IBCF", param = list(k = 5)),
  "user-based CF" = list(name = "UBCF", param = list(method = "Cosine", nn = 500))
  )
```

Each algorithm will be ran five times using the 5-fold cross-validation evaluation scheme. Each algorithm's performance is analyzed using ROC curves and Precision-Recall curves. This allows for an analysis of each model's tradeoff between true positive rates, false positive rates, and overall success of predictions.

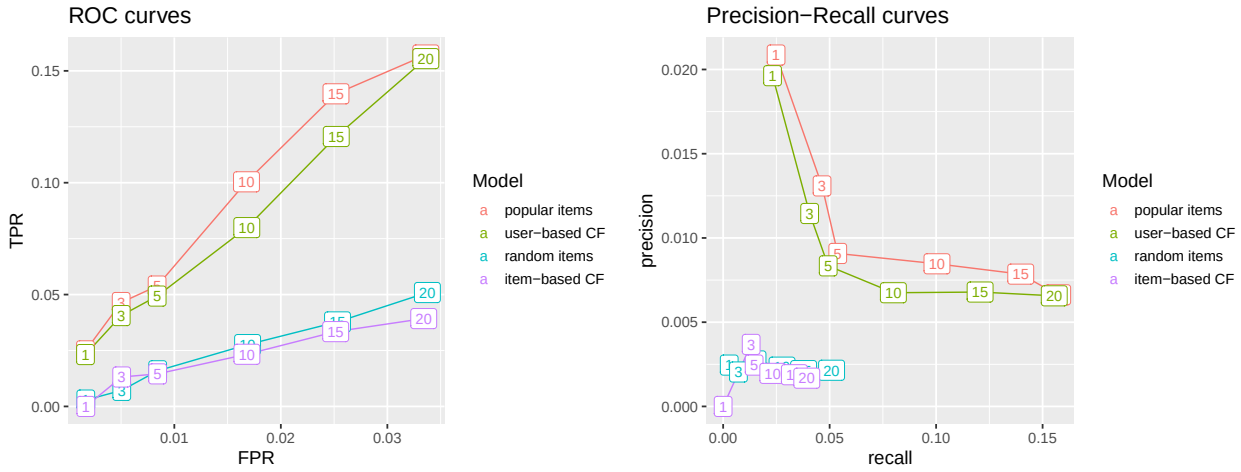
Model Performance

```
results <- evaluate(esCross, algorithms, type = "topNList", n = c(1, 3, 5, 10, 15, 20))
```

```
avg_conf_matr <- function(results) {
  tmp <- results %>%
    getConfusionMatrix() %>%
    as.list()
    as.data.frame(Reduce("+",tmp) / length(tmp)) %>%
    mutate(n = c(1, 3, 5, 10, 15, 20)) %>%
    select('n', 'precision', 'recall', 'TPR', 'FPR')
}

results_tbl <- results %>% map(avg_conf_matr) %>% enframe()
results_tbl <- unnest(results_tbl, colnames(results_tbl))
results_tbl
```

```
## # A tibble: 24 x 6
##    name              n precision  recall     TPR     FPR
##    <chr>         <dbl>     <dbl>   <dbl>   <dbl>   <dbl>
##  1 random items      1   0.00245 0.00285 0.00285 0.00169
##  2 random items      3   0.00204 0.00718 0.00718 0.00506
##  3 random items      5   0.00270 0.0159  0.0159  0.00844
##  4 random items     10   0.00233 0.0276  0.0276  0.0169
##  5 random items     15   0.00213 0.0378  0.0378  0.0253
##  6 random items     20   0.00215 0.0509  0.0509  0.0338
##  7 popular items     1   0.0209  0.0249  0.0249  0.00166
##  8 popular items     3   0.0131  0.0464  0.0464  0.00501
##  9 popular items     5   0.00908 0.0538  0.0538  0.00838
## 10 popular items    10   0.00847 0.100   0.100   0.0168
## # ... with 14 more rows
```

```
par(mfrow=c(2,1))

results_tbl %>%
  ggplot(aes(FPR, TPR,
             colour = fct_reorder2(as.factor(name),
                                   FPR, TPR))) +
  geom_line() + geom_label(aes(label = n)) +
  labs(title = "ROC curves", colour = "Model") +
  theme_grey(base_size = 14)

results_tbl %>%
  ggplot(aes(recall, precision,
             colour = fct_reorder2(as.factor(name),
                                   precision, recall))) +
  geom_line() +
  geom_label(aes(label = n)) +
  labs(title = "Precision-Recall curves", colour = "Model") +
  theme_grey(base_size = 14)
```



The UBCF and popular items are the best performing algorithms out of those tested.

Hybrid Model Recommendations

Since both UBCF and Popular Items algorithms performed relatively the same we will be adding equal weights. The individual ratings are combined using the weighted sum which is how we will determine our recommendations.

```
hybrid_recom <- HybridRecommender(Recommender(getData(esCross, "known"), method = "UBCF"),
                                  Recommender(getData(esCross, "known"), method = "POPULAR"),
                                  weights = c(.5, .5)
)
```

First Recommendation Set

```
as(predict(hybrid_recom, getData(esCross, "unknown")[1]), "list")
```

```
## $`6`
##  [1] "Close Shave, A (1995)"
##  [2] "Wrong Trousers, The (1993)"
##  [3] "Wallace & Gromit: The Best of Aardman Animation (1996)"
##  [4] "Schindler's List (1993)"
##  [5] "Shawshank Redemption, The (1994)"
##  [6] "Mr. Smith Goes to Washington (1939)"
##  [7] "Good Will Hunting (1997)"
##  [8] "Usual Suspects, The (1995)"
##  [9] "Casablanca (1942)"
## [10] "Rebecca (1940)"
```