Global Baseline Estimate

Jawaid Hakim

2022-09-17

Contents

1 Global Baseline Estimate

Most recommender systems use personalized algorithms like "content management" and "item-item collaborative filtering." Sometimes non-personalized recommenders are also useful or necessary. One of the best non-personalized recommender system algorithms is the "Global Baseline Estimate," which is demonstrated in the attached Excel spreadsheet, MovieRatings.xlsx. We will also walk through this spreadsheet in our 14-Sep-2022 meetup. Your job here is to use the survey data you collected in your previous assignment, and write the R code that lets you make an item (e.g. movie) recommendation using the Global Baseline Estimate algorithm.

There needs to be at least one respondent with at least two unrated items. You may modify your survey data if needed. Your task is to use the global baseline estimate algorithm to determine ratings for unseen items, then recommend an item to at least one of your survey respondents.

2 Data

Let's find out names of sheets in the XLSX file. The sheet we want to load is *MovieRatings*.

```
readxl::excel_sheets("MovieRatings.xlsx")
```

```
## [1] "MovieRatings"             "Problem Statement"
## [3] "MeanCenteredMovieRatings" "Global Baseline"
```

Let's load the *MovieRatings* sheet from XLSC file.

```
ds <- readxl::read_excel("MovieRatings.xlsx", sheet = "MovieRatings")
str(ds)
```

```
## tibble [16 x 7] (S3: tbl_df/tbl/data.frame)
##  $ Critic        : chr [1:16] "Burton" "Charley" "Dan" "Dieudonne" ...
##  $ CaptainAmerica: num [1:16] NA 4 NA 5 4 4 4 NA 4 4 ...
##  $ Deadpool      : num [1:16] NA 5 5 4 NA NA 4 NA 4 3 ...
##  $ Frozen        : num [1:16] NA 4 NA NA 2 3 4 NA 1 5 ...
##  $ JungleBook    : num [1:16] 4 3 NA NA NA 3 2 NA NA 5 ...
##  $ PitchPerfect2 : num [1:16] NA 2 NA NA 2 4 2 NA NA 2 ...
##  $ StarWarsForce : num [1:16] 4 3 5 5 5 NA 4 4 5 3 ...
```

Let's remove the *Critic* column from the data frame. Doing so has the advantage that all remaining columns are numeric and it's easier to perform computations on a data frame with only numeric data.

```
num_ds <- ds %>%
    select(-Critic)
num_ds
```

```
## # A tibble: 16 x 6
##    CaptainAmerica Deadpool Frozen JungleBook PitchPerfect2 StarWarsForce
##             <dbl>    <dbl>  <dbl>      <dbl>         <dbl>         <dbl>
##  1             NA       NA     NA          4            NA             4
##  2              4        5      4          3             2             3
##  3             NA        5     NA         NA            NA             5
##  4              5        4     NA         NA            NA             5
##  5              4       NA      2         NA             2             5
##  6              4       NA      3          3             4            NA
##  7              4        4      4          2             2             4
##  8             NA       NA     NA         NA            NA             4
##  9              4        4      1         NA            NA             5
## 10              4        3      5          5             2             3
## 11              5        5      5          5            NA             4
## 12             NA       NA      4          5            NA             3
## 13              5        5      5          4             4             5
## 14              4       NA     NA         NA            NA             4
## 15              4        5      3          3             3            NA
## 16             NA       NA      5          5            NA            NA
```

2.1 Mean Movie Rating

Now let's compute the mean movie rating. Sum of all ratings is 240 and count of valid ratings (excluding NA) is 61. Mean of all movie ratings is 3.93.

```
all_rat <- unlist(num_ds, use.names = FALSE)    # extract all movie ratings

sum_rating <- sum(all_rat, na.rm = TRUE)        # sum all ratings
sum_rating
```

```
## [1] 240
```

```
count_rating <- length(na.exclude(all_rat))     # count number of ratings, excluding NAs
count_rating
```

```
## [1] 61
```

```
mean_movie <- round(sum_rating / count_rating, 2) # compute mean movie ratings
mean_movie
```

```
## [1] 3.93
```

2.2 Movie Average

Compute (movie) averages of all numeric columns.

```
movie_average <- num_ds %>%
                    summarise_if(is.numeric, mean, na.rm = TRUE)
movie_average
```

```
## # A tibble: 1 x 6
##   CaptainAmerica Deadpool Frozen JungleBook PitchPerfect2 StarWarsForce
##            <dbl>    <dbl>  <dbl>      <dbl>         <dbl>         <dbl>
## 1           4.27     4.44   3.73        3.9          2.71          4.15
```

Add 2 rows at the end of data frame: 1st row contains movie averages, 2nd row contains the *movie average - mean movie*.

```
num_ds <- add_row(num_ds, movie_average)
num_ds <- add_row(num_ds, round(movie_average - mean_movie, 2))
tail(num_ds)
```

```
## # A tibble: 6 x 6
##   CaptainAmerica Deadpool Frozen JungleBook PitchPerfect2 StarWarsForce
##            <dbl>    <dbl>  <dbl>      <dbl>         <dbl>         <dbl>
## 1           5        5      5          4             4             5
## 2           4       NA     NA         NA            NA             4
## 3           4        5      3          3             3            NA
## 4          NA       NA      5          5            NA            NA
## 5           4.27     4.44   3.73       3.9           2.71          4.15
## 6           0.34     0.51  -0.2       -0.03         -1.22          0.22
```

2.3 User Average

Compute and add *user average* as column to data frame.

```
num_ds <- num_ds %>%
            mutate(UserAvg = rowMeans(., na.rm = TRUE)) %>%
            mutate(UserAvg = round(UserAvg, 1))
head(num_ds)
```

```
## # A tibble: 6 x 7
##   CaptainAmerica Deadpool Frozen JungleBook PitchPerfect2 StarWarsForce UserAvg
##            <dbl>    <dbl>  <dbl>      <dbl>         <dbl>         <dbl>   <dbl>
## 1             NA       NA     NA          4            NA             4     4
## 2              4        5      4          3             2             3     3.5
## 3             NA        5     NA         NA            NA             5     5
## 4              5        4     NA         NA            NA             5     4.7
## 5              4       NA      2         NA             2             5     3.2
## 6              4       NA      3          3             4            NA     3.5
```

Compute and add *user average - mean movie* as column to data frame.

```
num_ds <- num_ds %>%
            mutate("UserAvgMinusMeanMovie" = round(UserAvg - mean_movie, 2))
head(num_ds)
```

```
## # A tibble: 6 x 8
##   CaptainAmerica Deadpool Frozen JungleBook PitchPerfe~1 StarW~2 UserAvg UserA~3
##            <dbl>    <dbl>  <dbl>      <dbl>        <dbl>   <dbl>   <dbl>   <dbl>
## 1             NA       NA     NA          4           NA       4     4      0.07
## 2              4        5      4          3            2       3     3.5   -0.43
## 3             NA        5     NA         NA           NA       5     5      1.07
## 4              5        4     NA         NA           NA       5     4.7    0.77
## 5              4       NA      2         NA            2       5     3.2   -0.73
## 6              4       NA      3          3            4      NA     3.5   -0.43
## # ... with abbreviated variable names 1: PitchPerfect2, 2: StarWarsForce,
## #   3: UserAvgMinusMeanMovie
```

Extract *critics* from original data source.

```
critics <- ds %>%
            select('Critic')
critics <- unlist(critics, use.names = FALSE)
critics
```

```
##  [1] "Burton"    "Charley"   "Dan"       "Dieudonne" "Matt"      "Mauricio"
##  [7] "Max"       "Nathan"    "Param"     "Parshu"    "Prashanth" "Shipra"
## [13] "Sreejaya"  "Steve"     "Vuthy"     "Xingjia"
```

Add *movie avg* and *movie avg - mean movie* as rows to critics.

```
critics <- append(critics, 'movie avg')

critics <- append(critics, 'movie avg - mean movie')

tail(critics)
```

```
## [1] "Sreejaya"               "Steve"                  "Vuthy"
## [4] "Xingjia"                "movie avg"              "movie avg - mean movie"
```

Add *critics* columns back into data source.

```
num_ds <- add_column(num_ds, critics, .before = 0)
num_ds
```

```
## # A tibble: 18 x 9
##    critics        Capta~1 Deadp~2 Frozen Jungl~3 Pitch~4 StarW~5 UserAvg UserA~6
##    <chr>            <dbl>   <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Burton            NA      NA     NA       4      NA       4       4      0.07
##  2 Charley            4       5      4       3       2       3       3.5   -0.43
##  3 Dan               NA       5     NA      NA      NA       5       5      1.07
##  4 Dieudonne          5       4     NA      NA      NA       5       4.7    0.77
##  5 Matt               4      NA      2      NA       2       5       3.2   -0.73
##  6 Mauricio           4      NA      3       3       4      NA       3.5   -0.43
##  7 Max                4       4      4       2       2       4       3.3   -0.63
##  8 Nathan            NA      NA     NA      NA      NA       4       4      0.07
##  9 Param              4       4      1      NA      NA       5       3.5   -0.43
## 10 Parshu             4       3      5       5       2       3       3.7   -0.23
## 11 Prashanth          5       5      5       5      NA       4       4.8    0.87
## 12 Shipra            NA      NA      4       5      NA       3       4      0.07
## 13 Sreejaya           5       5      5       4       4       5       4.7    0.77
## 14 Steve              4      NA     NA      NA      NA       4       4      0.07
## 15 Vuthy              4       5      3       3       3      NA       3.6   -0.33
## 16 Xingjia           NA      NA      5       5      NA      NA       5      1.07
## 17 movie avg          4.27    4.44   3.73    3.9     2.71    4.15    3.9   -0.03
## 18 movie avg - m~     0.34    0.51  -0.2    -0.03   -1.22    0.22   -0.1   -4.03
## # ... with abbreviated variable names 1: CaptainAmerica, 2: Deadpool,
## #   3: JungleBook, 4: PitchPerfect2, 5: StarWarsForce, 6: UserAvgMinusMeanMovie
```

Define function to predict movie rating for any given movie and critic pair. Handle basic error cases of invalid movie or critic names.

```
me.predictRating <- function(movie, critic) {
                        if (! movie %in% colnames(num_ds)) {
                            print(paste("Error - nvalid movie:", movie))
                            NA
                        }
                        else if (! critic %in% num_ds$critics) {
                            print(paste("Error -  critic:", critic))
                            NA
                        }
                        else {
                            mean_movie +
                            num_ds[num_ds$critics == 'movie avg - mean movie', movie][[1]] +
                            num_ds[num_ds$critics == critic, 'UserAvgMinusMeanMovie'][[1]]
                        }
}
```

2.4 Test

Test predictions.

```
test_data <- c('PitchPerfect2', 'Param', 'Deadpool', 'Parshu', 'JungleBook', 'Vuthy', 'Frozen', 'Dan2',

for (i in seq(1, length(test_data), 2)) {
                        if (! test_data[i] %in% colnames(num_ds)) {
                            print(paste("Error - invalid movie:", test_data[i]))
                        }
                        else if (! test_data[i + 1] %in% num_ds$critics) {
                            print(paste("Error - invalid critic:", test_data[i + 1]))
                        }
                        else {
                            print(
                                paste(
                                "Rating for movie", test_data[i], "by critic", test_data[i + 1], "- Pred
                                me.predictRating(test_data[i], test_data[i + 1]),
                                ", Actual:",
                                num_ds[num_ds$critics == test_data[i + 1], test_data[i]]))
                        }
}
```

```
## [1] "Rating for movie PitchPerfect2 by critic Param - Predicted: 2.28 , Actual: NA"
## [1] "Rating for movie Deadpool by critic Parshu - Predicted: 4.21 , Actual: 3"
## [1] "Rating for movie JungleBook by critic Vuthy - Predicted: 3.57 , Actual: 3"
## [1] "Error - invalid critic: Dan2"
## [1] "Rating for movie Frozen by critic Dan - Predicted: 4.8 , Actual: NA"
## [1] "Rating for movie CaptainAmerica by critic Steve - Predicted: 4.34 , Actual: 4"
## [1] "Rating for movie JungleBook by critic Nathan - Predicted: 3.97 , Actual: NA"
```