User profile-based movie recommender

Aaron Stearns

6/11/2018

This recommender engine uses the Movielens dataset and generates recommendations for users based on their movie genre preference histories.

The 100k row version of the Movielens dataset used for this project is based on anonymous user reviews and viewing history collected in the late 1990s and is available .

First I'll import the data and do some initial cleaning and formatting:

```
library(Hmisc)
library(dplyr)

movies <- read.csv('http://files.grouplens.org/datasets/movielens/ml-100k/u.data',
                   sep = "\t",
                   header=F)

colnames(movies) <- c("user_id", "movie_id", "rating", "timestamp")

items <- read.csv('http://files.grouplens.org/datasets/movielens/ml-100k/u.item',
                  sep = "|",
                  header=F)

# renaming columns
names <- tolower(Cs(movie_id, title, release_date, video_release_date,
                    imdb, unknown, Action, Adventure, Animation,
                    Childrens, Comedy, Crime, Documentary, Drama,
                    Fantasy, Noir, Horror, Musical, Mystery, Romance,
                    SciFi, Thriller, War, Western))

colnames(items) <- names

df <- left_join(movies, items, by = "movie_id")

dfIndiv <- df[, c(1, 9:27)]

userTotals <- dfIndiv %>%
                group_by(user_id) %>%
                summarise_all(funs(sum))

# take the transpose of a subset of columns from the userTotals dataframe
userTranspose <- t(userTotals[, 2:19])
colnames(userTranspose) <- 1:943
```

Main recommender function. Takes in a single param, "x", which is the index in the data frame "userTranspose" of a target user you want to generate movie recommendations for.

```
movieRecommender <- function(x) {

  # similarUsers helper function finds top n users
  # with most closely correlated genre history
```

```r
similarUsers <- function(x) {

  user <- userTranspose[, x]
  others <- userTranspose[, -x]

  # correlate target user profile with other user profiles
  # to find most similar users. Take the transpose of this,
  # and create a dataframe
  comparisons <- data.frame(t(cor(user, others)))

  comparisons$user <- colnames(others)

  colnames(comparisons) <- c("correlation", "user")

  comparisons <- comparisons %>%
                 group_by(user) %>%
                 arrange(desc(correlation))

  tenMostSimilar <- comparisons[1:10, 2]
  # return 10 most highly correlated user profiles
  return(tenMostSimilar)
}

# ten most similar users to target user are returned
ten <- similarUsers(x)

ten <- as.numeric(unlist(ten))

# list all movies target user has watched
userWatched <- movies %>%
               filter(user_id == x) %>%
               select(movie_id, rating)

# list all movies the 10 most similar users have watched
otherUserMovies <- movies %>%
                   filter(user_id %in% ten) %>%
                   select(movie_id, rating)

# anti-join to get movies not yet seen by target user
unwatchedMovies <- anti_join(otherUserMovies,
                             userWatched,
                             by = "movie_id")

# find the highest mean ratings of unwatched movies
highestRated <- unwatchedMovies %>%
                group_by(movie_id) %>%
                summarise(averages = mean(rating)) %>%
                arrange(desc(averages))

# find the most frequently watched movies in unwatchedMovies
mostFrequent <- unwatchedMovies %>%
                group_by(movie_id) %>%
                summarise(watched = n()) %>%
```

```
                    arrange(desc(watched))

  ratedAndWatched <- inner_join(highestRated, mostFrequent, by = "movie_id")

  # find the highest rated, most frequently watched movies from similar users
  ratedAndWatched <- left_join(ratedAndWatched, items, by = "movie_id")

  userFavGenres <- userTranspose[, x]

  # find the top 5 favorite genres of the target user
  userFavGenres <- userFavGenres %>%
                    sort() %>%
                    tail(5) %>%
                    names()

  genreNames <- unlist(strsplit(userFavGenres, "_tot"))


  ratedAndWatched$keep <- ifelse(ratedAndWatched[, genreNames[1]] == 1 |
                                 ratedAndWatched[, genreNames[2]] == 1 |
                                 ratedAndWatched[, genreNames[3]] == 1 |
                                 ratedAndWatched[, genreNames[4]] == 1 |
                                 ratedAndWatched[, genreNames[5]] == 1, 1, 0)

  ratedAndWatched <- ratedAndWatched %>%
                      filter(keep == 1)

  # return the 100 most watched highest rated films not seen by target user
  highestAvgs <- ratedAndWatched %>%
                  arrange(desc(watched))%>%
                  head(100)

  highestAvgs <- highestAvgs %>%
                  arrange(desc(averages)) %>%
                  head(10)

  # return top 10 suggestions
  movieSuggestions <- head(highestAvgs$title, 10)

  return(movieSuggestions)
}
```

Testing the recommender engine:

```
# genre preference history for user #6
t(userTotals[6, ])

##             [,1]
## user_id        6
## unknown        0
## action        25
## adventure     22
## animation     10
## childrens     20
```

```
## comedy        66
## crime         14
## documentary    1
## drama        104
## fantasy        3
## noir           6
## horror         4
## musical       13
## mystery       12
## romance       41
## scifi         13
## thriller      24
## war           21
## western        5
```

It looks like user six's top five genres are dramas, comedies, romance, action, and thrillers. Let's see the top ten movies the function recommends:

```
# user_id 6
movieRecommender(6)
```

```
##  [1] Chinatown (1974)
##  [2] Some Folks Call It a Sling Blade (1993)
##  [3] It Happened One Night (1934)
##  [4] Being There (1979)
##  [5] Singin' in the Rain (1952)
##  [6] Strictly Ballroom (1992)
##  [7] Raise the Red Lantern (1991)
##  [8] Rear Window (1954)
##  [9] Bullets Over Broadway (1994)
## [10] Ridicule (1996)
## 1664 Levels: 'Til There Was You (1997) ...
```