

Project 4

Shipra Ahuja, Saheli Kar, Nathan Groom

July 11, 2016

Introduction

This recommender system will recommend books to various reader(s) located in different cities having similar taste in books. In other words, the recommender system will take into consideration the city where the reader(s) reside and will look for other reader(s) residing in the same or other nearby cities having the similar choice/taste of books. When the system locates the reader(s) with similar taste, the system will then select books read by other readers in and around the same city and recommend those books to the readers residing in that city.

Datasets

The following datasets have been used -

- 1) Books.csv - contains the list of 55 books with its attributes
- 2) Ratings.csv - contains the ratings by 86 readers for each of the 55 books
- 3) Ratings_with_cities.csv - Contains cities where each of the 86 users are located

Algorithm/Code - Collaborative Filtering

- 1) Upon reading all the datasets - books, ratings and ratings_with_cities datasets into R, the average rating for each book is computed.
- 2) The preference of each user for category of books and the author preference is obtained.
- 3) Hold the books and the cities in a matrix
- 4) Fetch the preferred books for each user from each category
- 5) Weightage is assigned to each book based on where the reader lives
- 6) For each of the 86 readers, get the product of weightage * avg score and populate the products for each reader into a matrix
- 7) Finally recommend the books for each user if the book has not been read by the user

```
# Read the books dataset
books <- read.csv("books.csv",header=TRUE)

# Read the ratings dataset
ratings <- read.csv("Ratings.csv",header=FALSE)

# Read the ratings with city dataset
ratings_with_city <- read.csv("Ratings_with_cities.csv", header=FALSE)

# Get number of rows of the books dataset
books_num<-nrow(books)

# Make column names of the dataframe as User,Cities,ISBN of the books
```

```

names(ratings)<-c("User", as.character(books$ISBN))
names(ratings_with_city)<-c("User", "Cities", as.character(books$ISBN))

# Get the average rating of each book
books=as.data.frame(cbind(books, avg_score=unname(mapply(mean, ratings[,2:(books_num+1)]))))

# Get the names of all readers
readers=ratings[,1]

# Get the names of all categories
categories<-unique(books$Category1)

# Matrix to hold the category preference of users
categories_ratings_matrix<-matrix(0,nrow=length(readers), ncol=length(categories),dimnames=list(readers, categories))

# Get the names of all authors
Authors<- unique(books$Author)

# Matrix to hold the Author preference of users
Authors_ratings_matrix<-matrix(0,nrow=length(readers), ncol=length(Authors),
                               dimnames=list(readers,Authors))

# Get the names of all cities
cities<- unique(ratings_with_city[,2])

# Obtain the category preference and author preference of each reader and populate the corresponding matrix
for(rownum in (1:nrow(ratings))) {
  for (colNum in (2:(books_num+1))) {
    readerName<-as.character(ratings[rownum,1])
    if(ratings[rownum,colNum]!=0){
      Category1<- books[books$ISBN==(names(ratings)[colNum]),]$Category1
      Category2<- books[books$ISBN==(names(ratings)[colNum]),]$Category2
      Author<-     books[books$ISBN==(names(ratings)[colNum]),]$Author

      categories_ratings_matrix[readerName, Category1] = categories_ratings_matrix[readerName, Category1]+ratings[rownum,colNum]
      categories_ratings_matrix[readerName, Category2] = categories_ratings_matrix[readerName, Category2]+ratings[rownum,colNum]
      Authors_ratings_matrix[readerName, Author] = Authors_ratings_matrix[readerName, Author]+ratings[rownum,colNum]
    }
  }
}

# Create a matrix to hold books with city names
Books_cities_profile<- matrix(0, nrow=length(cities), ncol=length(books$ISBN),
                              dimnames = list(cities, books$ISBN))

# Obtain a matrix of books and cities
for(i in 1:length(cities)){
  city <- cities[i]

```

```

for(j in 1:length(books$ISBN)){
  ISBN <- books$ISBN[j]
  Books_cities_profile[i,j] <- mean(ratings_with_city[ratings_with_city$Cities==city,(j+2)])
}
}

categories_ratings_matrix=categories_ratings_matrix/rowSums(categories_ratings_matrix)

#Fetch the preferred books for user for a particular category

getPreferredBooks<-function(User, Category, booksToFetch){
  #fetch all the books from the category
  booksInCategory<-books[books$Category1==Category | books$Category2==Category,c(1:3,6)]

  #Sort the books with average score
  booksInCategory<-booksInCategory[ order(booksInCategory[, "avg_score"], decreasing = TRUE), ]

  #Initialize vector for the books to recommend
  booksToRecommend<-c()
  count=as.numeric(booksToFetch)

  for(i in 1:length(booksInCategory)){
    bookISBN = booksInCategory[i, "ISBN"]

    #Add the book into the recommendation list if the user haven't read it
    if(ratings[ratings$User==User,bookISBN]==0){
      recommendBook<- paste0("Title::", booksInCategory[i,"Title"], "; Author::", booksInCategory[i,"Au
        "; ISBN::", booksInCategory[i,"ISBN"])
      booksToRecommend<-c(booksToRecommend, recommendBook)

      #Decrease the count by 1 as one book is recommended
      count=(count-1)
    }

    #If no more books to fetch then return the list
    if(count==0){
      return(booksToRecommend)
    }
  }
  return(booksToRecommend)
}

#Initialize the empty recommendation matrix

reco_matrix<- matrix(NA, nrow = length(readers), ncol = 5 )
rownames(reco_matrix)<-as.character(readers)

#Compute recommendation for all the users

collaborative_reco_matrix<-matrix(NA, nrow=86, ncol=55)

for(i in 1:86){
  city<-ratings_with_city[i,2]

```

```

for(j in 1:55){
  weightage<- Books_cities_profile[city,j]
  collaborative_reco_matrix[i,j]<-books[i,]$avg_score * weightage
}
}

collaborative_reco_matrix[is.na(collaborative_reco_matrix)] <- 0

RMSE = sqrt(mean((ratings[, 2:56] - t(collaborative_reco_matrix))^2))
RMSE

## [1] 1.867098

```

Algorithm/Code - Content-Based Filtering

- 1) Create a books_profile matrix that will hold the the weighted score for a book for each category. This will enable the user know how much a book falls into a particular category.
- 2) Create a user profile that will hold the normalized score of a book by each user.
- 3) Use the above two matrices to predict the scores by the users for a book and recommend books according to the score.
- 4) Createw the books_city profile matrix that will hold the popularity of a book in a city.
- 5) Use the predicted score from Step 3 and book's profile on a city to add a weightage on the previously calclated score. This will give a prediction of a score with our contextaul value - City.

```

# Create a matrix to get the number of books in each category
Category_books_count<- matrix(0, nrow=length(categories), ncol=1)
rownames(Category_books_count)<- categories

# Get the books in both categories and count of books in both categories
for(i in 1:nrow(books)){
  Category1 = books[i,]$Category1
  Category2 = books[i,]$Category2
  Category_books_count[Category1,1] = Category_books_count[Category1,1]+1
  Category_books_count[Category2,1] = Category_books_count[Category2,1]+1
}

# Define a book profile matrix to hold the books with respective categories
books_profile<- matrix(NA, nrow = length(books$ISBN), ncol = length(categories))

# Rows will be ISBNs
rownames(books_profile)<- as.character(books$ISBN)

# Columns will be books categories
colnames(books_profile)<- as.character(categories)

idf<- log(nrow(books)/Category_books_count)

for(i in 1:nrow(books_profile)){
  for(j in 1: ncol(books_profile)){
    category1<-as.character(books[i,"Category1"])

```

```

category2<-as.character(books[i,"Category2"])
idf1=0
idf2=0
if(as.character(categories[j])==category1){
  idf1<-idf[category1,]
}
if(as.character(categories[j])==category2){
  idf1<-idf[category2,]
}
books_profile[i,j]<- books[i,"avg_score"*(idf1+idf2)

}
}

# Get the average rating of the books per user
avg_rating_by_user = mean(unlist(ratings[,2:56]))

# Create the user profile matrix
user_profile<-matrix(0, nrow = nrow(ratings), ncol = nrow(books))
rownames(user_profile)<- ratings[,1]
colnames(user_profile)<- books$ISBN

# Noramlize the ratings
user_profile<- t(as.matrix(ratings[, 2:56]- avg_rating_by_user))
books_profile = rowSums(books_profile)

rec_profile<- user_profile+books_profile
rec_profile[rec_profile>5]=5

rownames(rec_profile) <- books$ISBN
colnames(rec_profile) <- ratings$User

for(i in 1:55){
  for(j in 1:86){
    city <- ratings_with_city[j,2]
    ISBN <- books$ISBN[i]
    rec_profile <- rec_profile * Books_cities_profile[city, ISBN]
  }
}

# Compute the RMSE
RMSE = sqrt(mean((ratings[, 2:56] - t(rec_profile))^2))
RMSE

```

```
## [1] 1.835059
```