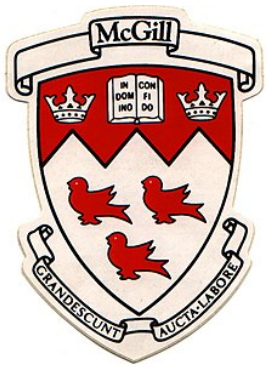
Final project
Hybrid-based Filtering

Building a CC (Content and Collaborative) Based Filtering Recommender System

Contacts etienne.tajeuna@mcgill.ca
dima.alsaleh@mcgill.ca

School of Continuing Studies

McGill University



Final project: Dataset

- In this project you will have to build a recommender system by leveraging both content and collaborative information.
- The data put under investigation is the MovieLens `https://drive.google.com/file/d/1cZ5cKXcwTp1A61MTctWIqDqaPH77ePhM/view?usp=sharing`.
- Main files to take into consideration are: ratings.csv, movies.csv, genome-scores.csv and genome-tags.csv.

- In the previous assessment, you have created the user profiles file: user_profiles.csv;
- Using this file, you are asked to create the graph of user preferences. For this, you have to create the dataframe given as,

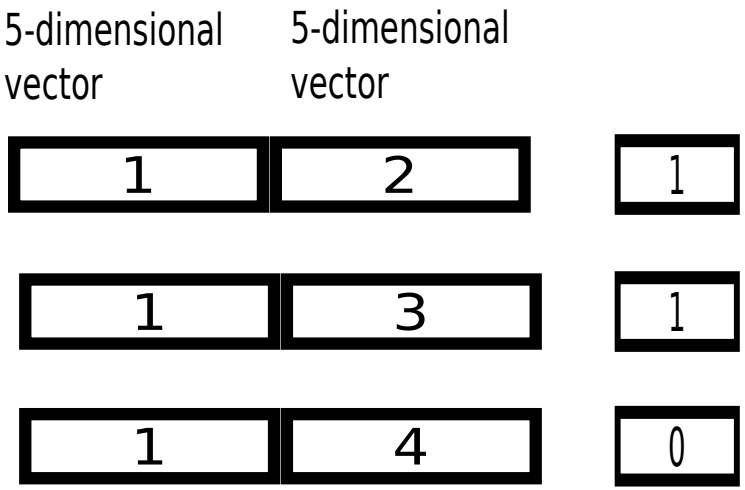
from	to	weight

  where the columns "from" and "to" will have user ids where as column "weight" will have the number of items two users well rated (that is rating > 2) together. For instance in the file "ratings.csv", the userId 1 and 2 both well rated 212 movies. In this case, the dataframe will have the row with values: $(1,\ 2,\ 4)$.
- In this file, two users should be recorded if and only if they both well rated at least 2.
- The dataframe should be returned into a csv file called "graph_representation".

- Based on files user_profiles.csv and graph_representation.csv we want to learn novel user representation for recommendation purpose.
- Using igraph, build the graph of users G. Each node of the graph should be the userId found in user_profiles.csv.
- We call adjacency matrix, the matrix $A = (a_{i,j})$, where $a_{i,j} = 0$ if there is no edge between userId "i" and "j". In case there is an edge $a_{i,j}$ should be the weight of the graph, that is the number of movies they well rated together (this value can be read from the data graph_representation). Determine the adjacency matrix A of the graph.
- We call degree matrix, the matrix $D = (d_{i,j})$, where $d_{i,j} = 0$ if $i \neq j$. In case $i = j$ $d_{i,j}$ should be the degree of the node i. Determine the degree matrix D of the graph.
- Based on the adjacency matrix and the degree matrix, calculate the Laplacian matrix $L = D - A$
- Calculate the matrix $M = L \cdot X$, where X should be the matrix of user profile.

- Using a PCA, you are asked to reduce the dimensionality of matrix M to 5.
- Return the novel user representation into a file called user_profiles2.csv.
- Over the obtained representation, perform a K-means with $K = 3$ to identify 3 clusters of users.
- Using the two first component of your PCA, plot your users into a 2-dimensional plan to visualize the users. Each group should be colored accordingly.
- Make an interpratation of your results.

- We now want to recommend items to users based on their relationships.
- For this we will use the logic of link prediction in a graph. Hence, from the constructed graph, you are asked to intentionally hide some edges (5%) for the online test purpose. 95% of the edges should be considered as known as saved for the offline test purpose.
- From the edges reserved for the offline test, you are asked to perform a 5-fold cross validation test. Here you are supposed to keep 80% of this offline data as train whereas 20% should be kept for the test data. To do this,
  1. Randomly split your offline data (5 times), and get 5 files of training edges and 5 files of testing edges.
  2. From each training file create the matrix $X_{train} \in \mathbb{R}^{e \times 10}$ and $y_{train} \in \mathbb{R}^{e \times 1}$ where e is the total number of combination between the 80% users (the same process should be done for the 20% datasets by creating the X_{test} and y_{test}).
  3. In fact X_{train} (resp. X_{test}) is the concatenation of the obtained user PCA representation. Whereas y_{train} (resp. y_{test}) is the vector telling if two users are related or not.



- An illustration is given in the above drawing.
- Here we have 4 users: 1, 2, and 4. Each user is represented by a 5-dimensional feature vector. Each row relates the concatenation of these features (this why we now have a 10-dimensional feature vector).
- The latest column, illustrate if two users are related or not. In this example, the value 1 in the first row of this column means that users 1 and 2 are related. Whereas the value 0 at the last row of this column means that users 1 and 4 are not related.

- Using an SVM (with RBF kernel), for each pair (X_{train}, y_{train}) you are asked to train your SVM model.
- Using the different X_{test} predict the values $\hat{y}_{test}$
- Give the average accuracy (balanced accuracy from sklearn) of the SVM.
- Give an interpretation of your result.

- We now want to evaluate our model online (that is over the 5% data)
- For this, we need to predict if a given user i is related to another user j and thus recommend her or him (user i) all movies of user j not yet watched.
- Here, we will use the clustering initially performed.
- For each user i in the 5% data, identify its corresponding cluster.
- Build the corresponding matrix X_{test} putting this user in relation with other users in the group.
- Use your SVM to predict the relationships.
- Using the balanced accuracy, evaluate your model.