AC / Machine Learning 2021/2022
Rita P. Ribeiro DCC - FCUP
Nuno Moniz DEI - FEUP

2 Hands On: Web Usage Mining

Read the file `log.csv`, containing information on the web pages visited by a set of users, into a data frame in R.

2.1 Simple Recommendation Strategies

Most Visited Pages

1. Recommend the 3 most visited pages. For that purpose:

(a) inspect how many times each page was visited;

(b) sort the pages by decreasing number of visits;

(c) obtain the top 3 pages for recommendation.

Using Clustering Results

2. Suppose we want to form two clusters of users, according to the pages they have visited. For that purpose:

(a) start by transforming the log access data into a matrix that has on each row a user and for each user the information on his visits to each page; this can be obtained with the `table()` function;

(b) use the function `dist()` to obtain a distance matrix with the Euclidean distance between the users;

(c) check for alternatives in the help page of `dist()`;

(d) use the function `hclust()` with the distance matrix to obtain an agglomerative clustering model of this data;

(e) visualize the obtained dendogram with function `plot()`;

(f) visualize again the dendogram, but now with option `hang=-0.1`.

(g) use the function `cutree()` to "cut"the hierarchical clustering in just two clusters; inspect the cluster membership of each user;

(h) use the function `rect.hclust()` to draw the previous solution in the dendogram.

3. Recommend the top 2 pages for users of cluster 1. For that purpose:

(a) inspect what were the pages visited by users in cluster 1;

(b) inspect how many times each of these pages were visited;

(c) sort the pages by decreasing order of visits;

(d) obtain the top 2 pages for recommendation.

4. Recommend the top 2 pages for users of cluster 2.

5. Using the same clustering results, recommend the top 3 pages for user u2.
From that top pages you should remove the pages that the user has already visited.

2.2 Recommendation using Association Rules

Load the package `recommenderlab` and read the `log1.csv` file.

6. Obtain a recommendation model using association rules with the first 6 users. For that purpose:

(a) start by coercing the data frame with user-page access information from the `log1.csv` file to a `binaryRatingMatrix` (`brm`);

(b) select the information on the first 6 users to be used as training offline data and save it to a new variable (e.g `brm_offline`);

(c) inspect the content of `brm_offline`; use the function `getRatingMatrix` and `getData.frame`;

(d) apply the functions `rowCounts` and `colCounts` to `brm_offline`; what information does it give you?

(e) apply the function `image` to `brm_offline`;

(f) obtain the recommender model based on association rules with the instruction

```
modelAR <- Recommender(brm_offline,"AR")
```

(g) apply the function `getModel` to the obtained model and then inspect the association rules that compose the model.

7. Suppose that u7 enters the system and becomes an active user. Deploy the recommendation model for him/her.
For that purpose:

(a) apply the `predict` function with the model and the rating matrix of the user, such that only the top 2 recommendations are given as output;

(b) apply the function `getList` to the obtained predictions to inspect the actual recommendations; which are they?

(c) to comprove the obtained recommendations, filter the rules which have been triggered for this active user.

8. Now suppose that u8 enters the system and becomes an active user. Deploy the recommendation model for him/her. Be critical regarding the results.

9. Explore the types of recommendation models available for binary rating matrices.

```
recommenderRegistry$get_entries(dataType ="binaryRatingMatrix")
```

10. Make the top 2 recommendations to u7 and u8 using the popularity of the pages, instead of association rules. Try to understand the obtained recommendations.

2.3 Recommendation using Collaborative Filtering

Binary Rating Data

Considering the same binary rating matrix of the previous exercise `brm_offline`, build a recommendation model based on collaborative filtering.

11. Start by using the function `similarity` to build the similarity cosine matrix for:

(a) an user-based approach;

(b) an item-based approach.

12. Obtain the top 2 recommendations with user-based CF and item-based CF methods using the cosine similarity with a neighborhood of size 3, for:

(a) active user `u8`;

(b) active user `u7`.

Non-Binary Rating Data

13. Explore the types of recommendation models available for real rating matrices.

```
recommenderRegistry$get_entries(dataType ="realRatingMatrix")
```

14. Read the file `log1Ratings.csv`, containing information on the ratings given to web pages by a set of users, into a data frame in R. Build an deploy the following colaborative filtering recommendation models using, again, the first 6 users for training:

(a) an user-based CF approach with two neighbours to predict the ratings of users `u7` and `u8`;

(b) an item-based CF approach with two neighbours to predict the ratings of users `u7` and `u8`.

2.4 Recommender Systems: Evaluation

15. Considering the log1 binary data, evaluate different recommendation strategies.

(a) Set the seed to 2021. Use the function `evaluationScheme` to define an evaluation scheme that splits the data into train and test set (80%-20% proportion) and establishes that 2 items of test cases are already known. In case that one or more users do not comply with this setting, you can disregard them.

(b) Check how the data was splitted according to the previous evaluation scheme, using the function `getData` on the evaluation scheme with the arguments `"train"`, `"known"` and `"unknown"`.

(c) Define the list of methods that will be used to obtain the top N recommendations, as follows:

```
methods <- list(
   "popular" = list(name="POPULAR", param = NULL),
   "user-based CF" = list(name="UBCF", param = NULL)
   "item-based CF" = list(name="IBCF", param = NULL)
)
```

(d) Use the function `evaluate` with the previously defined evaluation scheme, methods and considering top 1, 3 and 5 recommendations for each of the models.

(e) Explore the obtained object.

(f) Use the function `getConfusionMatrix` on one of the methods to obtain the corresponding confusion matrices. Be critical regarding the values that are shown.

(g) Plot the ROC curves for each of the methods and different values of N. What can you conclude?

(h) Plot the precision/recall curves for each of the methods and different values of N. What can you conclude?