TIRESIA: report

Abstract

We tried to develop a method to build a social network graph starting from the data we gathered in the SmartBiblio system, developed inside the Infostud Lab app. We achieved a good result, but we need to develop our ideas further to make it a tool for scientific research. We faced a big problem in the evaluation of our results; indeed, we do not have a solid ground truth to use and the analysis methods we explored did not give us a good insight of the truthfulness of our clusters, used as basis to build the social network.

Introduction

We are three students from the bachelor's degree course of Computer Science in Sapienza. We had a common experience in SapienzaApps laboratory: our internship involved app development and we have worked as developers in InfoStud and InfoProf mobile applications.

One of the functionalities of InfoStud is SmartBiblio: it gives the possibility to the students to reserve, confirm or cancel a seat reservation into Sapienza's libraries.

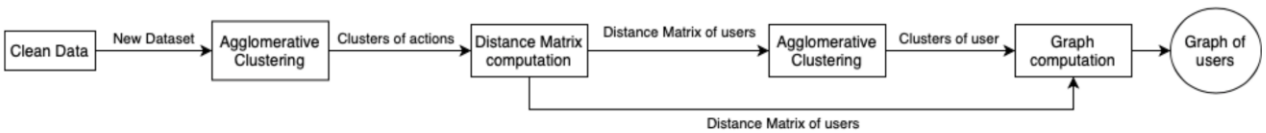
The dataset is composed by users' actions in the system, which can be *reserve*, *confirm* or *cancel* and are made by the *system* or the *user*. Of course, there is also the timestamp when the action is performed and the identification code corresponding to the student.

The idea arises from the desire to exploit the data gathered in the past year, to generate new information, hoping to create a useful tool for data analysis.

The task is clustering group of users' actions, inferring links between people, which can be either friends or colleagues.

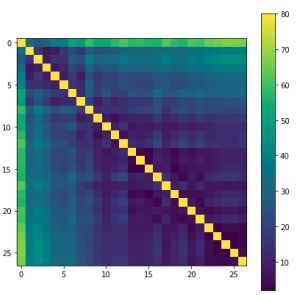
Proposed methods

After some research on the multiple existing clustering methods, we chose the Agglomerative Clustering one, because we do not know anything about the number of clusters we actually have and it allows us to have a hierarchical view on the clusters. We tried all the linkage methods with different types of affinity, like *single*, *complete*, *ward*, *centroid*, *average* and the relative *Euclidian*, *Manhattan* and other distances (for the sake of synthesis we will report only the most interesting ones).



The first step is dataset cleaning and features selection. Those are then passed to the dendrogram generator, which is useful to individuate the optimal number of clusters (or equivalently the correct distance). The distance is an hyperparameter of the agglomerative clustering, hence it is important

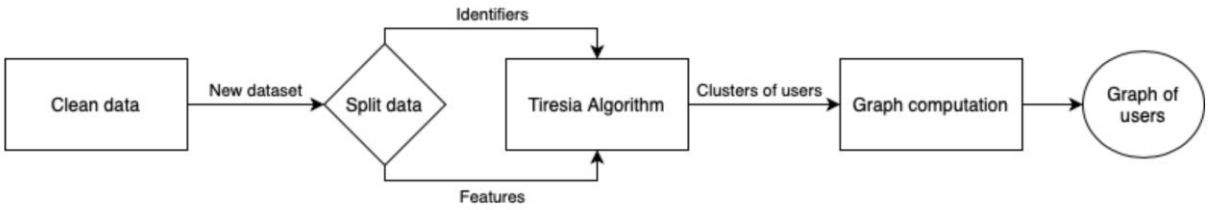
to know exactly what threshold we should pick. After running the clustering algorithm, the result is a corresponding number of clusters of actions.



After this step we build the distance matrix: users are coupled with an array of length equal to the total number of clusters. Thus, we count how many actions each user have in each cluster, so at the *i*-th array element there is the user's number of actions in the *i*-th cluster. The final matrix is computed by the Euclidian distance between the arrays.

At this stage we then proceed either passing the distance matrix again to the Agglomerative Clustering algorithm, which this time computes *user clusters* and finally computing the user graph; or we compute it directly from the distance matrix. Each node of the graph represents a user and there is an edge between the nodes *i* and *j* if they belong to the same *user cluster* or if their distance is smaller than a certain threshold.

Despite the results might seem good, there is no actual correspondence in the real scenario. For this reason we tried to implement our method to cluster user's actions, the TIRESIA algorithm.



The pipeline beginning is identical to the previous one, we need to clean the dataset.

In our means same actions by different users stay in the same cluster. To achieve this there is the necessity to split samples in two data. First the identifiers, which are attributes to group on and second the number of features: actions in our case. The algorithm computes the Euclidean distance between samples and search the appropriate cluster to add the respective identifier. With this implementation the number of clusters must not be pre-defined, indeed the algorithm looks like kMeans with a non-definite number of *k*.

The graph computation is another algorithm, the input is the results of TIRESIA, and the output is the network.

Dataset

The dataset is composed by 65.000 rows which represent the total users' actions in two different libraries subscribed in the system.

The dataset attributes are *userid*, *libraryid*, *seatid*, *created_at* (timestamp), *action* and *actionfrom*.

	userid	libraryid	seatid	created_at	action	actionfrom
0	[redacted]	2	4	2019-11-14 05:28:05	reserve	user

In the cleaning step we want to remove actions performed by the system and in the agglomerative

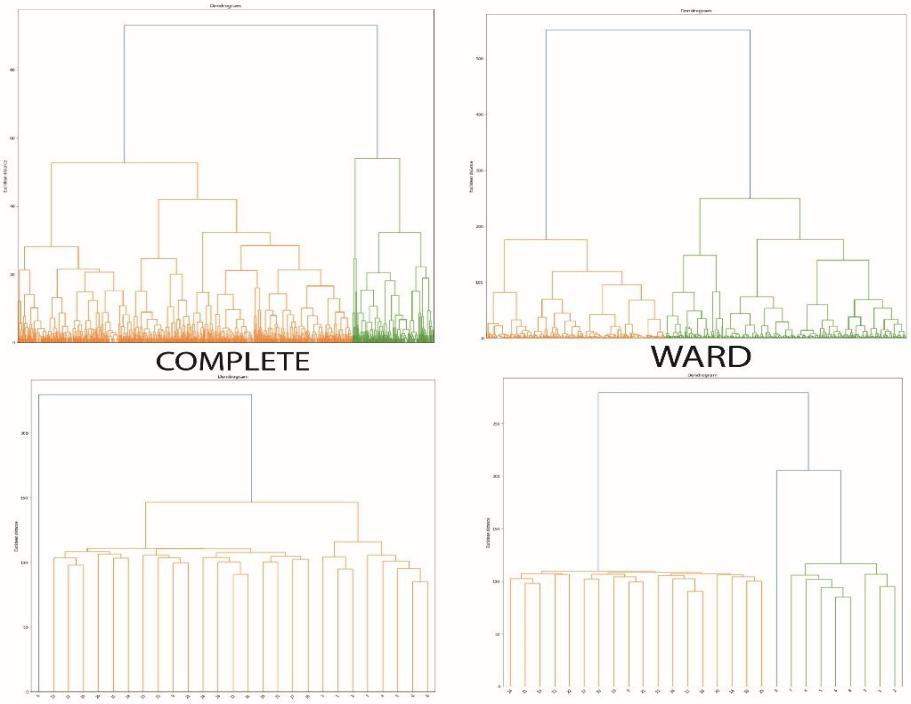
clustering, which we found out to be sensitive to the outliers, we only want the users who made over six actions.

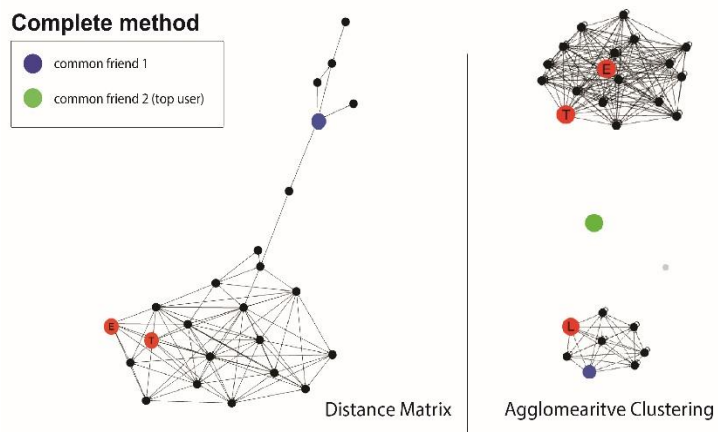
We then divided the timestamp in *day*, *month* and *timeGap* which is a division of the hours in time slots. The seats are dived in group of seats (*groupSeat*) and the *action* type is transformed in an integer value. So, the features are: *day*, *month*, *timeGap*, *groupSeat* and *action.* We did not select the *libraryid* as a feature because we ran our tests on the first library (our laboratory), where we know the connections between the users, and the second library (Boaga) is used to show the results after tuning the models in the test set, since we don't know the actual connections in it.

Results

Agglomerative Clustering

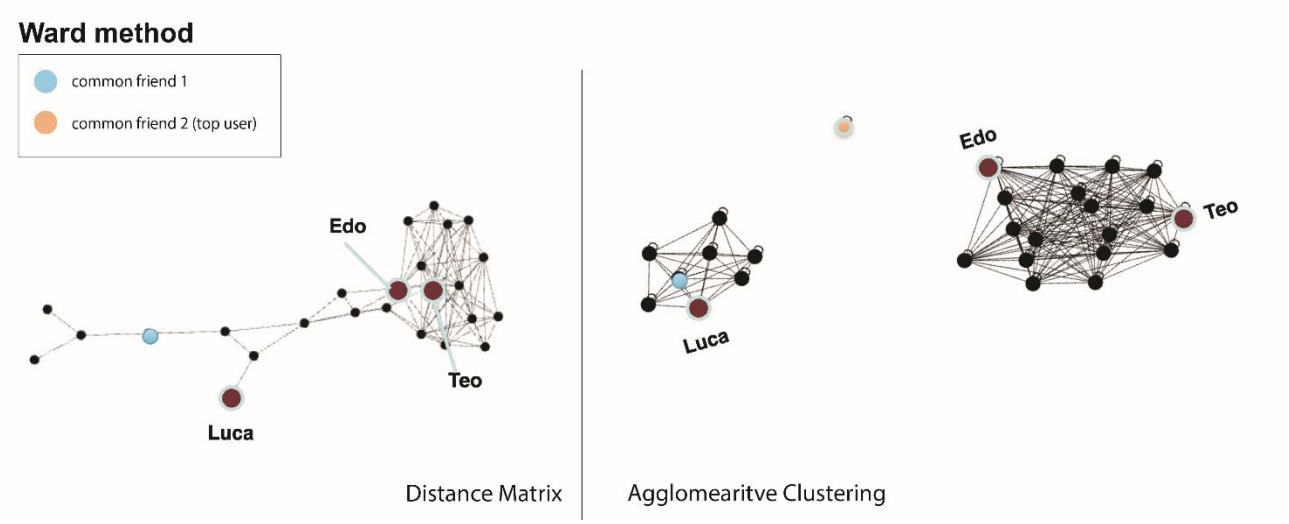
Those are the dendrograms for the agglomerative clustering: in the first row there are the dendrograms generated by the actions and in the second one the dendrograms which cluster the users. In the first column we used the *complete* method and, in the second one, the *ward*. Among those two we prefer *ward* because the distances are greater when it merges the last two cluster.





Since we know the links between developers in the laboratory, we can confirm that the *complete* linkage method is not optimal, indeed both the graphs present some mistakes. In the first one it does not include the user with more actions in the system and in the second one he is not connected with us.

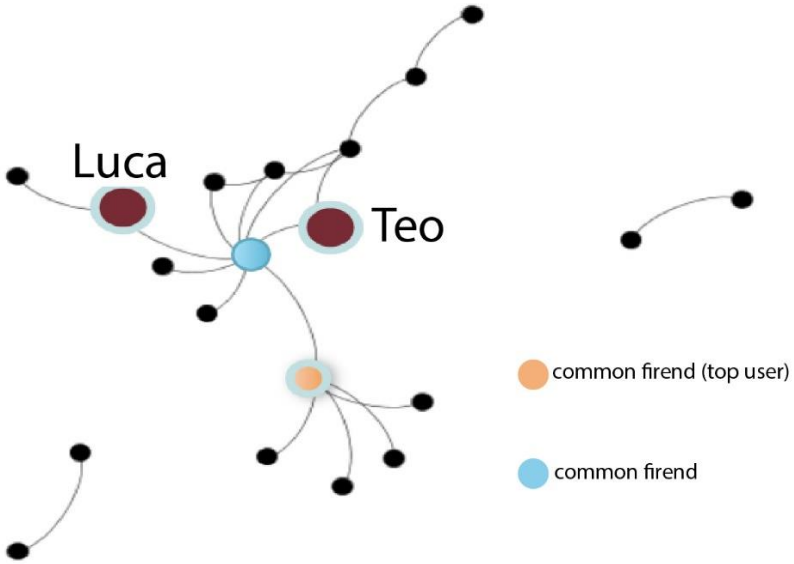
Moreover, Luca is not present in the first graph, while in the second one it is disconnected from us, like our common friend who is unliked to Teo and Edoardo in both of them.



The links between people seem better than the previous linkage method but with a no correspondence with the actual scenario, indeed we expected the colored nodes to be clustered together. Moreover, this linkage method suffers of the same problems of the above one: in the leftmost graph, the top user is not detected (but Luca, at least, is present this time) and in the rightmost one he is clustered again alone.

The results are not representing enough the reality, so we chose to implement a clustering algorithm from scratch, trying to achieve better results.

TIRESIA algorithm



With the TIRESIA clustering model we obtain this graph. In this figure we can see that Luca and Teo are connected by a common friend (like the real scenario) and the top user is a central node, connected with several users. Instead Edoardo is not been clustered, due to the lack of data about him we suppose but we need to elaborate on this.

Conclusion

After we tested the TIRESIA approach on GamLab library we can confirm that the result of this implementation, in this specific case, is good and works better than the known clustering model.

The algorithm creates a lot of cluster of users based on their similar actions. For this reason, the result has a good connection and a no-redundancy of edges like the previous networks.

Finally, we can run the algorithm on the big dataset composed by the Boaga library with all its seats bookable and in a period with medium frequency of access.
Of course, we need to deepen our knowledge and prove formally our model. We know it works well in our laboratory, but we do not know if the same applies in the Boaga library or, in general, in all the libraries where SmartBiblio is in function. Core drilling tests were carried out to test it on this result. The correspondences that emerged from the cores were good and further convinced us of the goodness of the model, although we have no way of having scientific proof.
This model, in the COVID-19 situation, can also be a useful way to understand people frequentation and it could be useful for sociologist to understand the social dynamics in the public places. Below you can see the graph resulting from the Boaga dataset.