Algo 2 matching, graphs, clustering : project description

NICOLAS LE HIR
nicolaslehir@gmail.com

1 DESCRIPTION OF THE PROJECT

The goal of the project is too analyze and process a dataset of your choice, using methods studied during the course.

1.1 Dataset constraints

You are free to choose the dataset within the following constraints :

— utf-8 encoded in a **data.csv** file

— several hundreds of lines

— a number n with $5 < n < 10$ (approximately) attributes (or columns, or features), the first being a unique id, separated by commas

— some fields must be quantitative (numbers), others categorical (not numbers).

— some fields could be correlated (for instance there is a correlation between temperature and pressure as seen in class)

It's nice if the dataset comes from a real example but you can also generate it, or even build a hybrid dataset mixing real data and generated data. The goal of the project is to apply some methods seen in class.

Example resources to find datasets :

— Link 1

— Link 2

— Link 2

— Link 4

1.2 Processing

The goal of the processing is to build a distance between datapoints and use it to build a compatibility graph. Then a matching or a clustering should be performed on these data.

Remark : for instance, the kmeans clustering method works with distances and does not need edges to be built. Thus, it works without building a graph with the data.

The processing must be made with **python 3** with **two files** :

build_graph.py Generates of a graph to represent the **compatibilities** between the datapoints (build edges between some of them).

You have to choose how to build the compatibility graph : as we have seen in class, there might be several relevant ways to do it.

However, most probably you will have to :

— 1) quantify the non-quantitatives variables,

— 2) normalize and/or weight the importance of the different fields,

— 3) remove useless variables,

— 4) create a distance or a similarity between datapoints,

— 5) set a threshold and

— 6) build edges between points that are seperated by a distance smaller than the threshold, or between which the similarity is higher than the threshold.

match.py or **cluster.py** Returns either

— a matching in the created graph (extration of the greatest possible number of pairs of compatible elements). For instance a maximal or maximum matching.

— OR, more relevant, a clustering of the data.

In the case of the matching in a bipartite graph, we have seen in the course that there exists a polynomial algorithm to exactly solve the problem.

In the case of a non-bipartite graph, other algorithms exist, such as the blossom algorithm. You may also implement an heuristic of your choice. For instance, some degree-based heuristics exist.

In the case of a clustering, you can use any classic method and any heuristic in order to justify the number of clusters used. We saw some example heuristics during the class but other heuristics are also possible.

In both cases (matching or clustering), you may do some research in other to find a heuristic that you find interesting and relevant for your dataset.

The processing must return a **result.csv** file containing a list of pairs (for a matching) or groups (for a clustering) of ids representing the matching or the clustering.

Important :

The usage of this dataset and its processing should be justified by a question of your choice. Thus, the approach should be explained and justified in a separate pdf file. The pdf file needs to contain explanations about :

— the nature of the dataset

— explanations about its processing, and in particular the construction of the distance or similarity.

— description of the matching or clustering used.

— comments on the results obtained.

If relevant, some comments on the distribution of the data, for instance by studying correlations between columns, or by analyzing the distribution of a given column, will be appreciated.

Do not hesitate to use networx or graphviz or another tool to illustrate the graph created, or parts of the graph if more relevant.

2 ORGANIZATION

The students can form groups with 2 or 3 students (they can work alone too).

The deadline for submitting the project is **December 20th 2020**. You may send a compressed folder or a repo containing :

— the name of the student(s)

— the csv dataset **data.csv**

— the algorithm **build_graph.py**

— the algorithm **match**

Please write "Algo 2 matching session 1" in the subject of your email.

You can reach me by email, I will answer faster if you use the gmail address rather than the epitech address.

3 EXERCISES DONE DURING THE COURSE

The exercises we made during the class are available with correction here : https://github.com/nlehir/ALGO2. The repository contains example functions you can use to create graph images, such as **random_graph.py**. It is not mandatory to use this repo.

4 LIBRARIES

You may use third-party librarie, except for the matching/clustering part, where it is required that you implement the algorithm yourself. However, if you use libraries, for instance for loading the data or visualizing them, it is required that you present them shorty in your document and describe the functions that you use from the library. These comments on the libraries do not need to be long.