Algo 2 matching, graphs, clustering : project

NICOLAS LE HIR
nicolaslehir@gmail.com

1 DESCRIPTION OF THE PROJECT

The goal of the project is too generate a dataset and to process it.

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
- around 10 attributes (columns), the first being a unique `id`, separated by commas
- some fields must be quantitative (a number), others not
- some fields must 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 randomly generate it.

1.2 Processing

The processing must be made with python with two files :

build_graph.py should generate of a graph to describe 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 probablty you will have to :

- 1) quantify the non-quantitatives variables,
- 2) normalize and/or weight the importance of the different fields,
- 3) remove the 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, of between which the similarity is higher than the threshold.

match.py or **cluster.py** should return a maximum matching in the created graph (extration of the greatest possible number of pairs of compatible elements) OR if it is more relevant, it should return groups of datapoints containing a number $\geqslant 3$ of elements (ie a cluster).

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

In the case of the clustering, the problem might require a harder algorithm in terms of complexity. You can thus use a heuristic to solve it, and/or a classical method.

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.

The usage of this dataset and its processing should be justified by a question of your choice. The approach should be explained and justified. The more interesting and original the dataset is, and the more relevant and justified the construction and matching in the graph, the higher the score.

Don't hesitate to use graphviz or another tool to illustrate the graph created.

Example references include [Cormen et al., 2009].

2 ORGANIZATION

The students can form groups with at most 4 students (they can work alone too).

The projet must be sent to me before **March 17th** in a compressed folder containing :

- the name of the student(s)
- the csv dataset **data.csv**
- the algorithm **build_graph.py**
- the algorithm **match**

You can reach me by email.

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 obviously not mandatory to use this repo !

4 VALIDATION

The project is not mandatory and can offer an extra credit, apart from the two credits based on being present in the class.

REFERENCES

[Cormen et al., 2009] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). Introduction to Algorithms, Third Edition.