Graph clustering and community detection in networks

Michel Habib
habib@liafa.univ-paris-diderot.fr
http://www.liafa.univ-paris-diderot.fr/~habib

STRUCO, 13 november 2013

Schedule of the talk

Introduction : a very hot subject

Community detection in graphs
- Clustering by similarity
- The bipartite case
- Clustering by betweeness parameters
- Clustering by edge ratio

Clustering of huge graphs in distributed computing

Conclusions
- New research directions

Community detection applications

- Targeting Marketing (Recommender algorithms such as in Amazon Website [1])
- Social control (NSA, CIA, FBI and others, see Snowden)
- Applications to the search of "Bad" behaviour as used in the US army.
- Many personal data are available for free on social networks

...

1. Toufik Bennouas our PhD student made the first version

- Graph partitionning for huge data structures
- Real applications in Biology to discover relational structure between micro-organisms (virus or bacteria) and pieces of genoms (attached to identified functions).

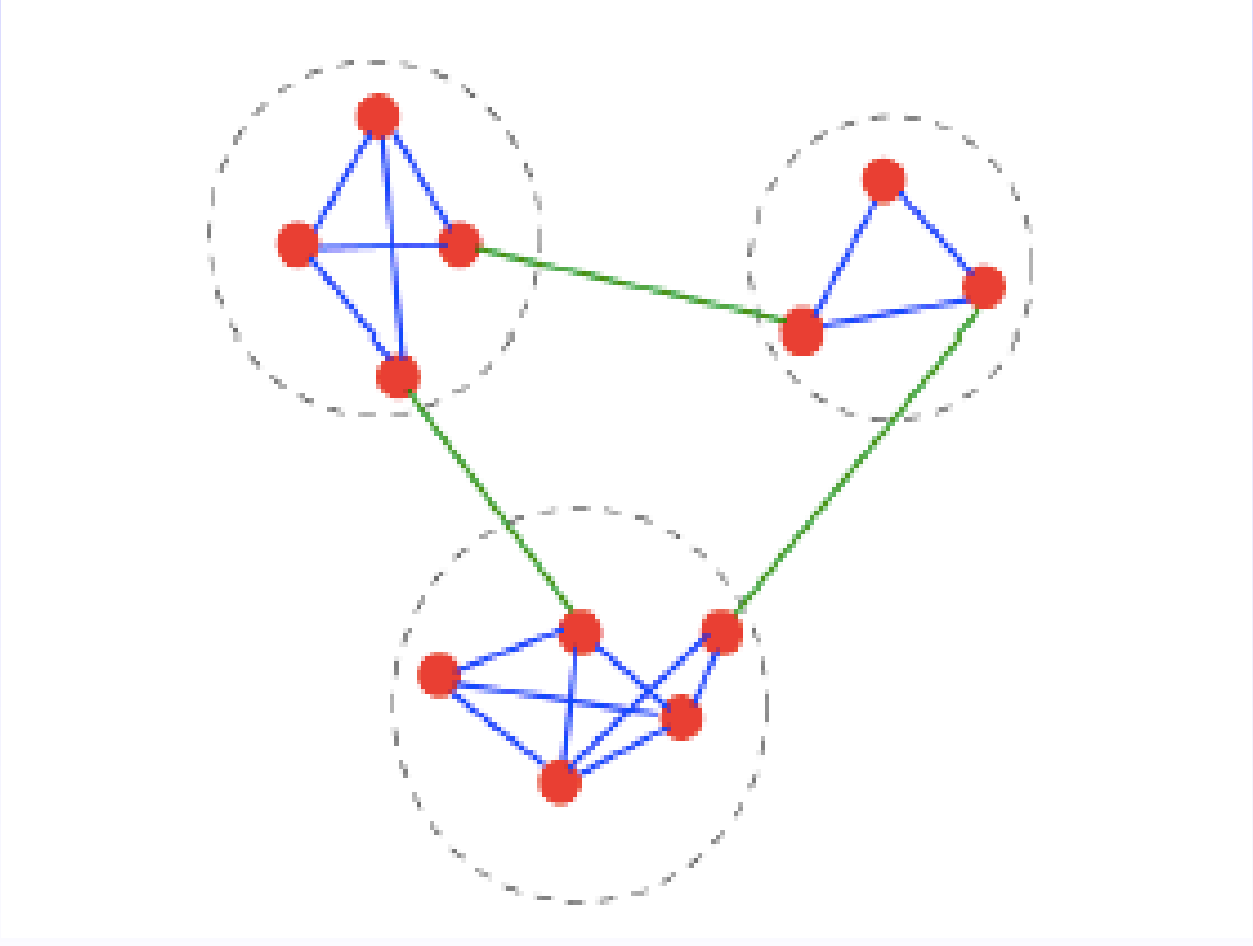
Classification

Georges Louis Leclerc, Comte de Buffon, dans son Histoire naturelle (1749) :
<< Le seul moyen de faire une méthode instructive et naturelle [de classification] est de mettre ensemble les choses qui se ressemblent et de séparer celles qui diffèrent les unes des autres >>

Clustering

Formally, given a data set, the goal of clustering is to divide the data set into clusters such that the elements assigned to a particular cluster are similar or connected in some predefined sense. However, not all graphs have a structure with natural clusters. Nonetheless, a clustering algorithm outputs a clustering for any input graph. If the structure of the graph is completely uniform, with the edges evenly distributed over the set of vertices, the clustering computed by any algorithm will be rather arbitrary.

Intuition behind this definition



- The formal definition of clustering differs from a domain to another (image processing, financial analysis, biology, social networks . . . or from an author to another one.
- The only way to compare methods is to use (Benchmarks made up with graphs designed with a natural clustering), see Santo Fortunato's surveys paper : Community detection in graphs, Archiv 2010.
- We aim at robust methods (i.e. little change in the data do not modifiy the resulting clustering).
- Graphs considered are often sparse .

Communities

- 2 kind of comunities.
- Explicit : example : Liafa PhD from 2000 to 2010,
  Or : Members of STRUCO
  We hope that people are proud to belong to such communities !
  Can be obtained with access to some database.

- Implicit : members do not know their appartenance to such a community.
  These community are obtained by computation from data, more interesting algorithmically.
  Examples : Members of parliament who voted similarly some important laws in some time period.
  Or : At distance at most 2 of Jarik in Facebook, sharing the same taste on wine, musics and politics.
  This information is not in some database (I hope so !) and has to be extracted from various sources.
  ...
- Let us focuse on community computation or extraction.

2 antagonist or orthogonal definitions

1. Clustering by similarity :
   Put together vertices playing the same role in the network.
2. Clustering by edge ratio :
   A community is a non-empty set of vertices $S \subseteq V(G)$ which are more intensely connected with each other than with vertices in $V(G)$-S.
   Another implicit assumption, community are connected subgraphs.

- Example : Let us consider a cycle on n vertices. Substitute every vertex of the cycle by a clique of size k. On this example the 2 definitions give the same decomposition, and the community are the cliques.
- But if instead we substitute the empty graph with k vertices. This partition does not optimize the Newmann modularity.

Modules

- $M \subseteq V(G)$ is a module iff
  $\forall x, y \in M,\ N(x) - M = N(y) - M$
- Modular decomposition defines a unique tree structure.
- Unfortunately most (real life) graphs are prime.
- Hard to generalize this notion. Umodules ...

Roles in graphs

- Role comme from social networks theory in sociology in the 1970. Vertices are individuals equiped with types (cop, judge, robber, professor, politician ...)
- Two vertices play the same role in the network iff they have the same colors (or types) in the neighbourhood.
- Good idea, but unfortunately it is NP-hard to compute these vertices (Fiala and Paulusma 2005).

- Modules are computable in linear time, but no use for practical issues
- Roles NP-hard to compute
- Can we define approximation of modular decomposition ?

Particular case of bipartite graphs

- For bipartite graphs, we can only cluster by similarity.
- The idea is to compute a partitioning of the vertices into bicliques (or quasi-bicliques) maximal under inclusion. Biclique are complete bipartite.
  Applications : Amazon the graph Customers–products . . ..
- In some applications a covering (not a partition) of the vertices is required.

Complexity status

- The computation of a maximum size biclique in a bipartite is NP-hard.
- The enumeration problem is # P-complet.
- Only heuristics are available, but they work not so badly.
- The edges could be weighted, and the method can be recursive.

Summarization techniques = optimisation the encoding of the bipartite.

Research directions

- Compute the bimodular decomposition it further decomposes the graph.
- Find an approximation calculus for roles (what is the parametrized complexity of this problem).

Betweeness en français les sociologues disent : Centralité d'intermédiarité.
We compute for every edge the number of geodesic paths (shortest paths) using this edge ;
The main hypothesis under this technique is that edges that belong to many shortest paths must be between clusters.

A graph *clustering* is a partition of its vertices into parts called en **communities** or **clusters**) such that :

- Every cluster has few edges to other cluster
- Every cluster a many inside edges
- Edges can be weighted

Partitionning

Name : Graph partitionning
Data : a graph G and valuations $\rho : V(G) \to N$ and $\omega : E(G) \to N$, 2 positive integers k, k'
Question : Does there exist a partition of $V(G)$ en $V_1, \ldots, V_h$ such that :
$\rho(V_i) \leq k$ and the sum of the valuation of the edges joining the V_i is less than k' ?

Well known result :

Graph partitionning is NP-hard.

There exists a whole family of clustering methods. They can be distinguished by the :

- parameter to optimize
- method top down or bottom up.
- computation time

Newman's modularity the reference measure 2002

G a graph and $P = V_1..V_k$ a *partition* of $V(G)$ in k parts. Roughly **Newman's modularity** of a partition is the ratio of *internal edges* minus the ratio of internal edges of the same partition but on the random graph.

More formally, let V_i and V_j be two parts. E_{ij} be the set of edges joining V_i to V_j.

The ratio of edges joining V_i and V_j is

$$e_{ij} = \frac{|E_{ij}|}{m}$$

If the graph was a random graph the probability of the existence of an edge uv would be

$$\frac{d^+(u) \times d^-(v)}{m^2}$$

let a_{ij} be the ratio of edges joining V_i to V_j in the random graph

$$a_{ij} = \sum_{u \in V_i, v \in V_j} \frac{d^+(u) \times d^-(v)}{m^2}$$

Then Newman's modularity of the partition P, denoted by $Q(P)$ is :

$$Q(P) = \sum_{i=1}^{i=k} e_{ii} - a_{ii}$$

Or equivalently :

$$Q(P) = \sum_{i=1}^{i=k} \left(\frac{|E_{ii}|}{m} - \sum_{u,v \in V_i} \frac{d^+(u) \times d^-(v)}{m^2} \right)$$

So Newman modularity is a number between -1 and 1. A value near to 1 is supposed to indicate a good clustering.

- It exists several variations on this definition in the literature (variations on the definition of a_{ij}) to measure the quality of a partition.
  We can take the ratio between internal and external edges instead of comparing to the random graph.

But

- It is NP-hard for a given graph to find the partition with highest Newman's modularity.
- This measure is not robust. Some experiments show that many partitions have the same Newman's modularity value, and therefore the heurisitics are not robust since a little change on the data can change the obtained partition.
- Fabien de Montgolfier, Mauricio Soto, Laurent Viennot : Asymptotic Modularity of Some Graph Classes. ISAAC 2011 : 435-444.

- Many different heuristics
- Very few "graph based" (using ideas from graph theory)
- One method compute the edges which belong to a maximum number of shortest paths between vertices. (These edges are supposed to be joining two parts).
  Delete these edges and recurse.
- Roswall, Bergstrom 2008
  One other method used a random search on the graph to discover some of its structure.
- Using centrality or a diametral path.

Map and Reduce

Using MapReduce as introduced by Google to play with distributed data with redundancy, is not so adapted for graph algorithms applied on huge graphs.

Graph Search using MapReduce

Only layered search can be done within this framework and exploiting some parallelism but not in linear time.
Even BFS with a queue data strucutre is not possible.

2 Graph programming languages

Pregel from Google 2010
Giraf for the Hadoop platform (available free)

Good clustering required

In order to distribute the data of the huge graph.
A lot of experimental and theoretical research has to be done with these tools.

Research problem

Can we used some knowledge on the structure of the graph when clustering ?

- small world hypothesis . . .
- Some work by D. Krastch if the graph has a dominating path.

Some technique proposed by T. Uno 2013

For some graph mining method, you are looking for maximal cliques. But the result can be a huge number, for example 800 000.
But we can transform the graph applying the following rule for every pair of vertices x, y :
If $|N(x) \cap N(y)| > \mathit{threshold}$ then add the edge xy.
This decreases the number of maximal cliques of the graph.
T. Uno used this to find spam Web sites, using this technique on the Web graph.

- What kind of preporcessing could improve the clustering ?
- Can we smooth a graph in order to discover a kind of modular decomposition ?

For the clustering by edge ratio, can we find parameters for which the number of good partitions in a given graph is small ?
This will ensure some coherence in the results.

Anticlustering techniques

How can we modify the graph in order that find good cluster is difficult ?
This area of research is now starting.

The two ways of clustering do not have the same applications.

- If you want to find some structure in a social network I would propose clustering by similarity.
- To manage a huge graph in a distributed system I would suggest to cluster by edge ratio.

Thank you for your attention !