Optimizing a Hierarchical Partition of a Complex Network

François Queyroi

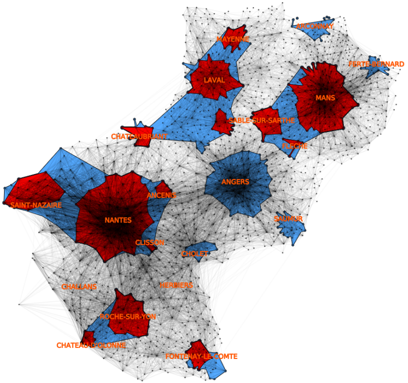
LaBRI, University of Bordeaux

ComplexNetworks, LIP6 – July 9, 2013

Hierarchical community structure

A partition of the vertices where each cluster can be recursively subdivided into a new partition.

Example : *Pays-de-la-Loire* commuters network (1999) with the *ZAUER* classification (INSEE).



State of the art (algorithms producing or *using* a hierarchical partition)

- **Agglomerative:**
  - Hierarchical Clustering based on distances [Ward Jr, 1963]
  - Iterative call of a "flat" clustering algorithm the compound graph induced by the previous partition [Blondel et al., 2008]
- **Divisive:**
  - Edge filtering [Girvan and Newman, 2002, Radicchi et al., 2004]
  - Iterative call of a "flat" clustering algorithm on each previously detected cluster [Auber et al., 2003]
- **Others:**
  - Probabilistic models [Clauset et al., 2006]
  - *Hierarchical Infomap* [Rosvall and Bergstrom, 2011] (hybrid method)
  - Dendrogram filtering using a scale parameter [Pons and Latapy, 2011]

Search space size

For $i = 1 \ldots 10$

- $\#\{$**Flat Partition**$\}$ **:** (*Bell number*)
  1, 2, 5, 15, 52, 203, 877, 4140, 21147
- $\#\{$**Dendrograms**$\}$ **:** (*Double factorial*)
  1, 1, 3, 15, 105, 945, 10395, 135135, 2027025, 34459425
- $\#\{$**Hierarchical Partitions**$\} \geq$ ([Flajolet and Sedgewick, 2009])
  1, 1, 4, 26, 236, 2752, 39208, 660032, 12818912, 282137824

Search space size

For $i = 1 \ldots 10$

- $\#\{$**Flat Partition**$\}$ **:** (*Bell number*)
  1, 2, 5, 15, 52, 203, 877, 4140, 21147
- $\#\{$**Dendrograms**$\}$ **:** (*Double factorial*)
  1, 1, 3, 15, 105, 945, 10395, 135135, 2027025, 34459425
- $\#\{$**Hierarchical Partitions**$\} \geq$ ([Flajolet and Sedgewick, 2009])
  1, 1, 4, 26, 236, 2752, 39208, 660032, 12818912, 282137824

Problems:

- Evaluate the quality of a hierarchical partition of a graph.
- Exploring the space of hierarchical partitions

Hierarchical Partition

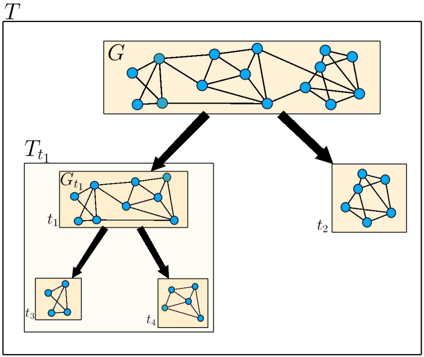
Let $G = (V, E)$ be a graph, a hierarchical partition is a (partition) tree T where :

- Each node $t \in T$ corresponds to a subset $C_t \subset V$.
- σ_t is the list of successors of t and we have $C_t = \bigcup_{f \in \sigma_t} C_f$.
- $\mathcal{F}(T)$ (the *leaves*) is the subset of nodes having $\sigma_t = \emptyset$.
- $N_i(T)$ is the ith level of T *i.e.* the set of nodes at a distance **at most** i from the root such as $\bigcup_{t \in N_i(T)} C_t = V$.
- T_t (resp. G_t) is the subtree rooted in t (resp. the subgraph induced by C_t).

Remark : A flat partition $\mathcal{C}$ is a hierarchical partition with $N_1(\mathcal{C}) = \mathcal{F}(\mathcal{C})$.
Example (Commuters networks) : Classify the cities according to the administrative region and then according to the "departement".

Hierarchical Partition

Let $G = (V, E)$ be a graph, a hierarchical partition is a (partition) tree T.



- $\sigma_{t_1} = \{t_3, t_4\}$
- $N_1(T) = \{t_1, t_2\}$
- $\mathcal{F}(T) = N_2(T) = \{t_3, t_4, t_2\}$

Quality measure

A *quality measure* is a function $\Phi(G, \mathcal{C}) \rightarrow \mathbb{R}$.
If $\Phi(G, A) > \Phi(G, B)$, A is said "to be better" than B.

Examples:

- Modularity Q [Newman, 2006]
- Mancoridis index MQ [Mancoridis et al., 1998]
- Performance [van Dongen, 2000]
- Map Equation [Rosvall and Bergstrom, 2008]
- Surprise [Aldecoa and Marín, 2011]

Additive quality measure

An *additive quality measure* is a quality measure which can be written:

$$\Phi(G, \mathcal{C}) = \sum_{i}^{k} \phi(G, \mathcal{C}, C_i)$$

where $\phi(G, \mathcal{C}, C_i)$ is the *gain* of the cluster C_i.

Examples:

- Modularity Q [Newman, 2006]
- Mancoridis index MQ [Mancoridis et al., 1998]
- Performance [van Dongen, 2000]
- Map Equation [Rosvall and Bergstrom, 2008]
- ~~Surprise [Aldecoa and Marín, 2011]~~

Strongly additive quality measure [Pons and Latapy, 2011]

A *strongly additive quality measure* is a quality measure which can be written:

$$\Phi(G,\mathcal{C}) = \sum_{i}^{k} \phi(G, C_i)$$

where $\phi(G, C_i)$ is the *gain* of the cluster C_i
(does not depend on the way $(V \setminus C_i)$ is partitioned).

Examples:

- Modularity Q [Newman, 2006]
- Mancoridis index MQ [Mancoridis et al., 1998] (weighted version)
- Performance [van Dongen, 2000]
- ~~Map Equation [Rosvall and Bergstrom, 2008]~~
- ~~Surprise [Aldecoa and Marín, 2011]~~

Hierarchical quality measure [Queyroi et al., 2011]

Let $G = (V, E)$ be a graph with a hierarchical partition T and ϕ be the gain function for a cluster. The *hierarchical quality measure* is

$$\Phi(G, T, q) = \sum_{i \in \sigma_{r(T)}} \phi(G, C_i)\,(1 + q \times \Phi(G_i, T_i, q))$$

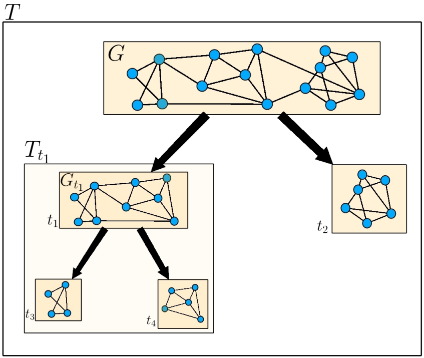
with $q \in [0, 1]$. $r(T)$ is the root of T.

Remarks :

- $\Phi(G, T, q)$ is a one-variable polynomial.
- The variable q can be used to favour deeper hierarchical clustering.

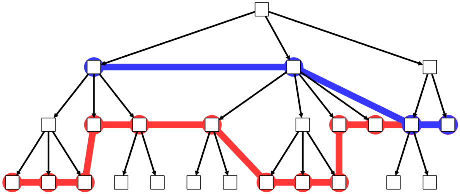
Without loss of generality, we will use as hierarchical quality criterion:

$$\Phi(G, T) = \int_0^1 \Phi(G, T, q)dq$$



$$\Phi(G, T, q) = \phi(G, C_{t_1}) + \phi(G, C_{t_2}) + q\ \phi(G, C_{t_1})\,(\phi(G_{t_1}, C_{t_3}) + \phi(G_{t_1}, C_{t_4}))$$
$$\Phi(G, T) = \phi(G, C_{t_1}) + \phi(G, C_{t_2}) + \frac{1}{2}\ \phi(G, C_{t_1})\,(\phi(G_{t_1}, C_{t_3}) + \phi(G_{t_1}, C_{t_4}))$$

Objective: Find the best sub-hierarchical partition induced by a hierarchy



Equivalent to

- Find the best subset of (non-horizontal) cuts
- Find the best subset of nodes of T to remove

Number of solutions is $\mathcal{O}(2^{|T|})$ $\Rightarrow$ **Use a heuristic**

The gain of node removal

Let $t \in T$, The gain of the removal of t is the difference between the quality of T and the quality of $T \setminus \{t\}$.

$$\Delta_t \Phi(G, T) = \Phi(G, T \setminus \{t\}) - \Phi(G, T)$$

The removal of a internal node t has several effects:

- The nodes of T_t are now "higher" in the hierarchy.
- σ_t is no longer a partition of G_t but is part of a partition of $G_{a(t)}$ (where $a(t)$ is the direct ancestor of t).

The gain of node removal

Let $t \in T$, The gain of the removal of t is the difference between the quality of T and the quality of $T \setminus \{t\}$.

$$\Delta_t\Phi(G,T) = \Phi(G, T \setminus \{t\}) - \Phi(G,T)$$

Algorithm 1: Greedy Optimisation of T.

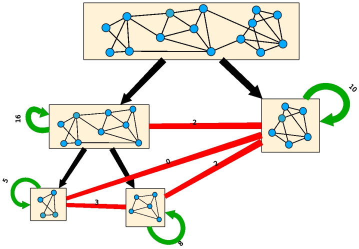
Input: $G = (V,E)$ a graph, T a hierarchical clustering of G
Output: T' a hierarchical clustering of G

```
T' = T;
t_max = arg max_{t ∈ T'} Δ_t Φ(G, T');
while Δ_{t_max} Φ(G, T) > 0 do
    T' = T' \ {t_max};
    t_max = arg max_{t ∈ T'} Δ_t Φ(G, T');
end
return T';
```

Time complexity

The time complexity of the procedure is $\mathcal{O}(h^2|E| + |T|^3)$ where $|T|$ is the number of nodes in T and h the height of T.

- The number of internal/external edges are computed in $\mathcal{O}(h^2|E|)$.
- Computation of $\phi(G, C_t)$ in constant time.
- Computation of $\Phi(G, T)$ (and $\Delta_t\Phi(G, T)$) is done using a tree traversal.



Modularity [Newman, 2006]

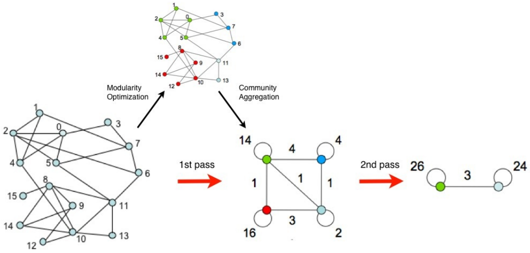
Let $G = (V, E)$ be a graph with a flat clustering $\mathcal{C}$

$$Q(G, \mathcal{C}) = \sum_{i}^{k} \frac{e_i}{|E|} - \left(\frac{d_i}{2|E|}\right)^2$$

where

- e_i : number of *internal* edges for cluster C_i
- d_i : sum of the degrees of vertices in C_i

[Blondel et al., 2008] introduced a heuristic (*Louvain* algorithm) for maximizing the modularity.



Remarks

- The iterative call leads to the construction of a hierarchy
- The output of the algorithm is the first level $N_1(T)$
- The produced hierarchical clustering can be of interest for some applications
- Some internal clusters may be irrelevant (the heuristic used may not be able to group a lot of vertices in one step)

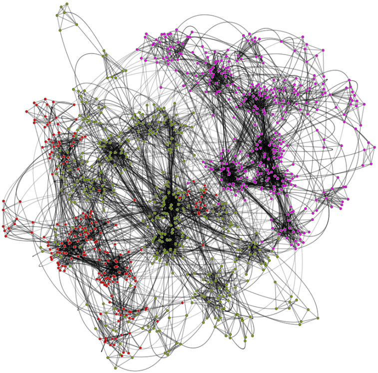
Modularity gain

We use $\forall t \in T$

$$\phi(G, C_t) = \frac{e_t}{e_{a(t)}} - \left(\frac{\sum_{u \in C_t} deg_{G_{a(t)}}(u)}{2e_{a(t)}}\right)^2$$

Hierarchical LFR Benchmark [Lancichinetti and Radicchi, 2008]

is a "realistic" random graphs model with a two level community structure.



Hierarchical LFR Benchmark [Lancichinetti and Radicchi, 2008]

is a realistic random graphs model with a two level community structure.

Parameters (relevant here) :

- $\mu = \mu_1 + \mu_2$: average proportion of edges from a micro-community...
- μ_1 : ... to another macro-community.
- μ_2 : ... to the same macro-community.

Hierarchical LFR Benchmark [Lancichinetti and Radicchi, 2008]

is a realistic random graphs model with a two level community structure.

Parameters (relevant here) :

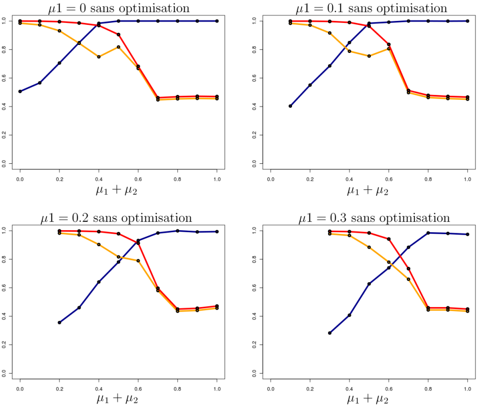
- $\mu = \mu_1 + \mu_2$: average proportion of edges from a micro-community...
- μ_1 : ... to another macro-community.
- μ_2 : ... to the same macro-community.

We want :

- $\mu_1, \mu_2 \simeq 0$: Only detect micro-communities.
- $\mu_1 < 0.5, \mu_2 \geq 0.5$: Only detect macro-communities.
- $\mu_1 < \mu_2 < 0.5$: Detect both micro and macro communities.

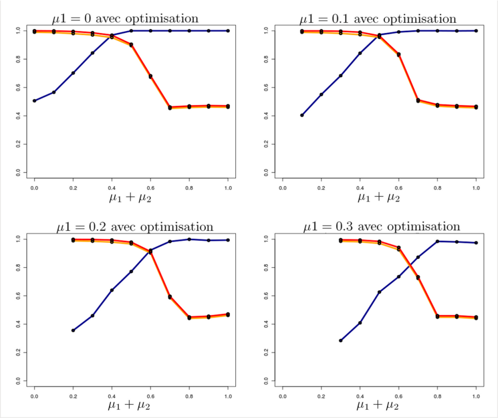
We compare our results to predictions using the *Variation of Information* distance:

- First produced level ($N_1(T)$) vs macro-communities
- Second produced level ($N_2(T)$) vs micro-communities
- Last produced level ($\mathcal{F}(T)$) vs micro-communities

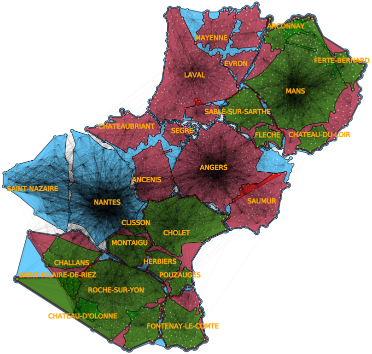


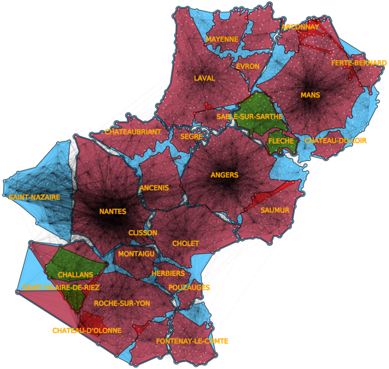
We compare our results to predictions using the *Variation of Information* distance:

- First produced level ($N_1(T)$) vs macro-communities
- Second produced level ($N_2(T)$) vs micro-communities
- Last produced level ($\mathcal{F}(T)$) vs micro-communities



Pays-de-la-Loire commuters network (INSEE – 1999).





Conclusion: I introduced

- a **hierarchical quality measure** that naturally extend additives measures
- a procedure to **improve hierarchical partitions**

Limitations: The procedure

- is a greedy algorithm
- needs to be apply on different hierarchies (Tulip plugin available soon!)

Future directions:

- Find useful **properties** of $\Phi(G, T)$
- Define a **distance** metric between hierarchical partitions

Thanks for your attention.

Références I

Aldecoa, R. and Marín, I. (2011).
Deciphering network community structure by surprise.
PloS one, 6(9):e24195.

Auber, D., Chiricota, Y., Jourdan, F., and Melançon, G. (2003).
Multiscale visualization of small world networks.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008).
Fast unfolding of communities in large networks.
Journal of Statistical Mechanics: Theory and Experiment, 2008:P10008.

Clauset, A., Moore, C., and Newman, M. (2006).
Structural inference of hierarchies in networks.
In *Proceedings of the 2006 conference on Statistical network analysis*, pages 1–13. Springer-Verlag.

Flajolet, P. and Sedgewick, R. (2009).
Analytic combinatorics.
Cambridge Univ Pr.

Girvan, M. and Newman, M. (2002).
Community structure in social and biological networks.
Proceedings of the National Academy of Sciences, 99(12):7821.

Références II

Lancichinetti, A. and Radicchi, F. (2008).
Benchmark graphs for testing community detection algorithms.
Physical Review E, 78(4):046110.

Mancoridis, S., Mitchell, B., Rorres, C., Chen, Y., and Gansner, E. (1998).
Using automatic clustering to produce high-level system organizations of source code.
In *Proceedings of the 6th International Workshop on Program Comprehension*, page 45. Citeseer.

Newman, M. E. J. (2006).
Modularity and community structure in networks.
Proceedings of the National Academy of Sciences, USA, 103:8577–8582.

Pons, P. and Latapy, M. (2011).
Post-processing hierarchical community structures: Quality improvements and multi-scale view.
Theoretical Computer Science, 412(8-10):892 – 900.

Queyroi, F., Delest, M., Fédou, J.-M., and Melançon, G. (2011).
Assessing the quality of multilevel graph clustering.
Data Mining and Knowledge Discovery.
à paraître.

Références III

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D. (2004).
Defining and identifying communities in networks.
Proceedings of the National Academy of Sciences of the United States of America, 101(9):2658–2663.

Rosvall, M. and Bergstrom, C. (2011).
Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems.
PloS one, 6(4):e18209.

Rosvall, M. and Bergstrom, C. T. (2008).
Maps of random walks on complex networks reveal community structure.
Proceedings of the National Academy of Sciences, 105(4):1118–1123.

van Dongen, S. (2000).
Graph Clustering by Flow Simulation.
PhD thesis, University of Utrecht.

Ward Jr, J. H. (1963).
Hierarchical grouping to optimize an objective function.
Journal of the American statistical association, 58(301):236–244.