

Combiner classification et visualisation pour l'exploration de grands réseaux

Nathalie Villa-Vialaneix

<http://www.nathalievilla.org>

nathalie.villa@univ-paris1.fr

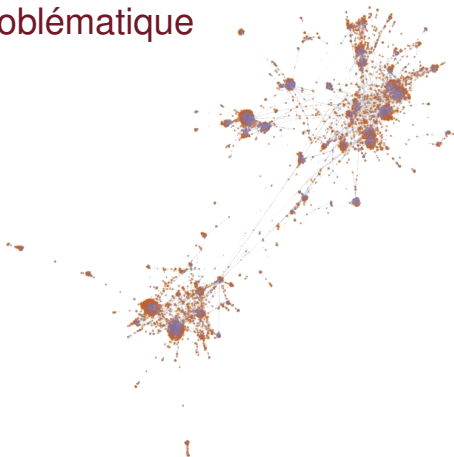
Journées MSTGA, INRA Toulouse, 7-8/11/2011

Travail joint avec Fabrice Rossi (SAMM, Université Paris 1)

Plan

- 1 Visualisation de graphes : pourquoi ?
- 2 Visualisation par classification hiérarchique
 - Obtention d'une hiérarchie de classifications
 - Visualisation d'une hiérarchie de classifications
 - Exemples et comparaisons

Contexte et problématique

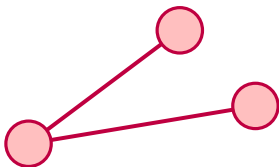


Un réseau (graphe) $\mathcal{G} = (V, E, W)$ avec

- n sommets $V = \{x_1, \dots, x_n\}$;
- des arêtes, E , pondérées par des poids $W_{ij} = W_{ji} \geq 0$ ($W_{ii} = 0$).

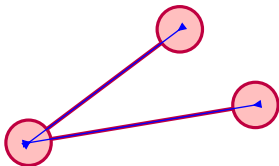
Visualisation : un outil pour comprendre le graphe

Approche courante pour explorer le graphe : visualisation par **algorithme de forces** (FDP), par exemple **[Fruchterman and Reingold, 1991]**



Visualisation : un outil pour comprendre le graphe

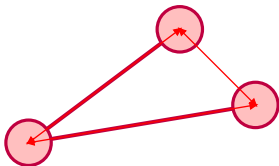
Approche courante pour explorer le graphe : visualisation par **algorithme de forces** (FDP), par exemple **[Fruchterman and Reingold, 1991]**



- **forces attractives** : le long des arêtes (analogues à des ressorts)

Visualisation : un outil pour comprendre le graphe

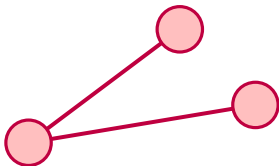
Approche courante pour explorer le graphe : visualisation par **algorithme de forces** (FDP), par exemple [**Fruchterman and Reingold, 1991**]



- **forces attractives** : le long des arêtes (analogues à des ressorts)
- **forces répulsives** : entre toutes les paires de sommets (analogues à des forces électriques)

Visualisation : un outil pour comprendre le graphe

Approche courante pour explorer le graphe : visualisation par **algorithme de forces** (FDP), par exemple [**Fruchterman and Reingold, 1991**]



- **forces attractives** : le long des arêtes (analogues à des ressorts)
- **forces répulsives** : entre toutes les paires de sommets (analogues à des forces électriques)

algorithme **itératif** jusqu'à stabilisation des positions des sommets.

Limites des approches FDP

- **lents** (difficiles à mettre en œuvre pour des graphes de très grande taille) ;

Limites des approches FDP

- **lents** (difficiles à mettre en œuvre pour des graphes de très grande taille) ;
- privilégient l'**esthétique** à l'interprétabilité :
 - *Tendance* : arêtes courtes et de tailles uniformes
 - *Conséquence négative* : regroupement des nœuds de forts degrés au centre de la figure

Limites des approches FDP

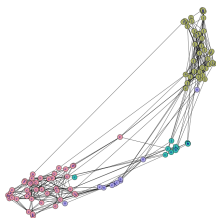
- **lents** (difficiles à mettre en œuvre pour des graphes de très grande taille) ;
- privilégient l'**esthétique** à l'interprétabilité :
 - *Tendance* : arêtes courtes et de tailles uniformes
 - *Conséquence négative* : regroupement des nœuds de forts degrés au centre de la figure

Or, **approche naturelle d'exploration d'un graphe** :

- 1 repérer la **structure macroscopique** : recherche de “communautés” et de leurs relations ;
- 2 **affiner les détails** dans certaines parties.

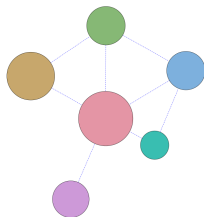
Mettre en valeur les “communautés” lors de la représentation

- 1 **approche globale** : représenter tous les sommets du graphe en modifiant les forces de manière à mettre en valeur les zones denses : **[Noack, 2007]** algorithme **LinLog**



Mettre en valeur les “communautés” lors de la représentation

- 1 **approche globale** : représenter tous les sommets du graphe en modifiant les forces de manière à mettre en valeur les zones denses :
[Noack, 2007] algorithme **LinLog**
- 2 faire une classification des sommets puis une **représentation simplifiée** **[Herman et al., 2000]**
 - **classification** des sommets en classes V_1, \dots, V_C ;
 - représenter le **graphe des classes** : sommets V_1, \dots, V_C (aire proportionnelle à $|V_j|$) et arêtes d'épaisseur proportionnelle à $\sum_{x_k \in V_i, x_{k'} \in V_j} W_{ij}$

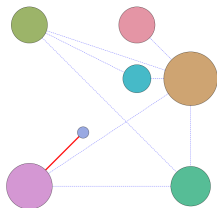


Mettre en valeur les “communautés” lors de la représentation

- 1 **approche globale** : représenter tous les sommets du graphe en modifiant les forces de manière à mettre en valeur les zones denses : **[Noack, 2007]** algorithme **LinLog**
 - 2 faire une classification des sommets puis une **représentation simplifiée [Herman et al., 2000]**
 - **classification** des sommets en classes V_1, \dots, V_C ;
 - représenter le **graphe des classes** : sommets V_1, \dots, V_C (aire proportionnelle à $|V_j|$) et arêtes d'épaisseur proportionnelle à $\sum_{x_k \in V_i, x_{k'} \in V_j} W_{ij}$
- problème** : Modifier FDP pour sommets de tailles variables.

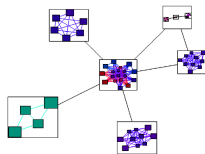
Mettre en valeur les “communautés” lors de la représentation

- 1 **approche globale** : représenter tous les sommets du graphe en modifiant les forces de manière à mettre en valeur les zones denses : **[Noack, 2007]** algorithme **LinLog**
- 2 faire une classification des sommets puis une **représentation simplifiée**
 approche alternative : **classification et placement simultanés** sur une carte auto-organisatrice **[Rossi and Villa-Vialaneix, 2010]**



Mettre en valeur les “communautés” lors de la représentation

- 1 **approche globale** : représenter tous les sommets du graphe en modifiant les forces de manière à mettre en valeur les zones denses : **[Noack, 2007]** algorithme **LinLog**
- 2 faire une classification des sommets puis une **représentation simplifiée**
- 3 approche hybride : **représentations hiérarchiques** dont la précision augmente
[Auber et al., 2003, Auber and Jourdan, 2005, Seifi et al., 2010]



Approches combinant classification et visualisation...

... basées sur la **modularité** :

$$Q(V_1, \dots, V_C) = \frac{1}{2m} \sum_{k=1}^C \sum_{x_i, x_j \in \mathcal{V}_k} (W_{ij} - P_{ij})$$

avec P_{ij} poids d'un « modèle nul » (poids dépendant seulement des degrés des sommets et non de leurs classes) :

$$P_{ij} = \frac{d_i d_j}{2m}$$

avec $d_i = \frac{1}{2} \sum_{j \neq i} W_{ij}$.

Interprétation

Une bonne classification **maximise la modularité** :

- $Q \nearrow$ quand (x_i, x_j) sont dans la **même classe** et $W_{ij} \gg P_{ij}$
 - $Q \searrow$ quand (x_i, x_j) sont dans **deux classes différentes** et $W_{ij} \ll P_{ij}$
- car

$$Q(C) + \frac{1}{2m} \sum_{k \neq k'} \sum_{i \in C_k, j \in C_{k'}} (W_{ij} - P_{ij}) = 0.$$

Interprétation

Une bonne classification **maximise la modularité** :

- $Q \nearrow$ quand (x_i, x_j) sont dans la **même classe** et $W_{ij} \gg P_{ij}$
- $Q \searrow$ quand (x_i, x_j) sont dans **deux classes différentes** et $W_{ij} \ll P_{ij}$
car

$$Q(C) + \frac{1}{2m} \sum_{k \neq k'} \sum_{i \in C_k, j \in C_{k'}} (W_{ij} - P_{ij}) = 0.$$

- Modularité : **aide à séparer des hubs** (\neq spectral clustering ou critère de coupe minimal) mais **petit défaut de résolution** (voir **[Fortunato and Barthélémy, 2007]**).

Interprétation

Une bonne classification **maximise la modularité** :

- $Q \nearrow$ quand (x_i, x_j) sont dans la **même classe** et $W_{ij} \gg P_{ij}$
- $Q \searrow$ quand (x_i, x_j) sont dans **deux classes différentes** et $W_{ij} \ll P_{ij}$
car

$$Q(C) + \frac{1}{2m} \sum_{k \neq k'} \sum_{i \in C_k, j \in C_{k'}} (W_{ij} - P_{ij}) = 0.$$

- Modularité : **aide à séparer des hubs** (\neq spectral clustering ou critère de coupe minimal) mais **petit défaut de résolution** (voir **[Fortunato and Barthélémy, 2007]**).

Problème : Optimisation = **Pb NP-complet** (pas possible pour des graphes de plus de qq centaines de sommets)

Plan

- 1 Visualisation de graphes : pourquoi ?
- 2 Visualisation par classification hiérarchique
 - Obtention d'une hiérarchie de classifications
 - Visualisation d'une hiérarchie de classifications
 - Exemples et comparaisons

Présentation générale de la méthodologie

2 étapes combinées :

- Obtention d'une **hiérarchie de classifications** (par optimisation de la modularité) + **test de la significativité du partitionnement** à chaque niveau ;
- **Représentations** des divers niveaux de la hiérarchie avec algorithme de forces modifié.

Plan

- 1 Visualisation de graphes : pourquoi ?
- 2 Visualisation par classification hiérarchique
 - Obtention d'une hiérarchie de classifications
 - Visualisation d'une hiérarchie de classifications
 - Exemples et comparaisons

Méthodes d'approximation de l'optimisation de la modularité

Méthode rapide : « Algorithme de Louvain » **[Blondel et al., 2008]**
(méthode gloutonne multi-niveaux) : peut traiter des graphes avec des millions de nœuds ; légèrement sous efficace en terme d'optimisation.

Méthodes d'approximation de l'optimisation de la modularité

Méthode rapide : « Algorithme de Louvain » **[Blondel et al., 2008]**
(méthode gloutonne multi-niveaux) : peut traiter des graphes avec des millions de nœuds ; légèrement sous efficace en terme d'optimisation.
[Noack and Rotta, 2009] : travail expérimental comparatif ⇒
recommandation pour des améliorations de l'approche précédente (rapide et plus efficace)

Méthodes d'approximation de l'optimisation de la modularité

Méthode rapide : « Algorithme de Louvain » **[Blondel et al., 2008]**
(méthode gloutonne multi-niveaux) : peut traiter des graphes avec des millions de nœuds ; légèrement sous efficace en terme d'optimisation.
[Noack and Rotta, 2009] : travail expérimental comparatif ⇒
recommandation pour des améliorations de l'approche précédente (rapide et plus efficace)
Ici : méthodologie basée sur **[Noack and Rotta, 2009]** avec quelques améliorations (dont ajout d'un test de connexité)

Description de l'approche de classification retenue

Étape 1 : Obtention d'une première partition

- **Construction gloutonne d'une partition**

- Initialiser les classes : $C_1^{(1)} = \{x_1\}, \dots, C_n^{(1)} = \{x_n\}$
- Répéter : fusionner 2 classes qui maximisent

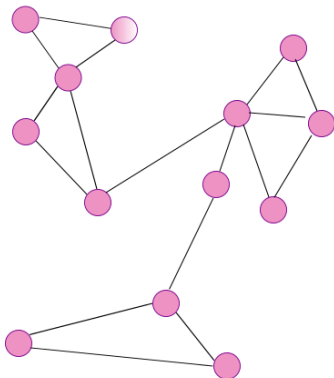
$$\text{Sig}(C_i, C_j) = \frac{\Delta Q_{C_i, C_j}}{\sqrt{\text{deg}(C_i)\text{deg}(C_j)}}$$

jusqu'à ne plus pouvoir augmenter la modularité par fusion.

Description de l'approche de classification retenue

Étape 1 : Obtention d'une première partition

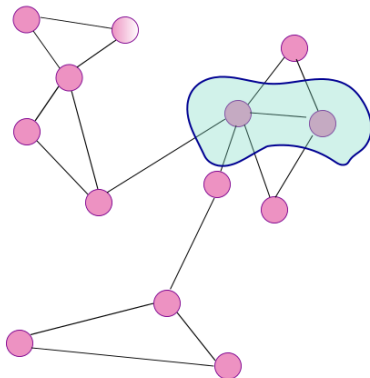
- **Construction gloutonne d'une partition**



Description de l'approche de classification retenue

Étape 1 : Obtention d'une première partition

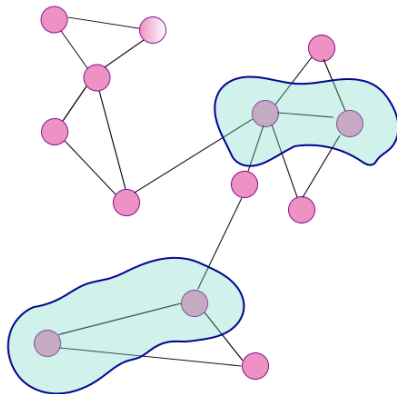
- **Construction gloutonne d'une partition**



Description de l'approche de classification retenue

Étape 1 : Obtention d'une première partition

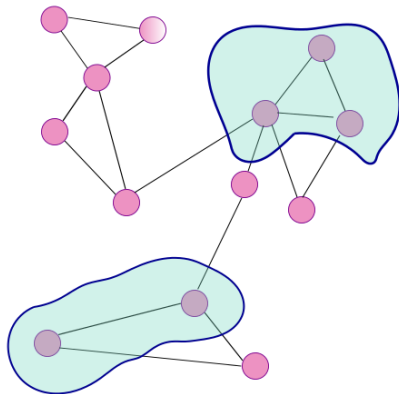
- **Construction gloutonne d'une partition**



Description de l'approche de classification retenue

Étape 1 : Obtention d'une première partition

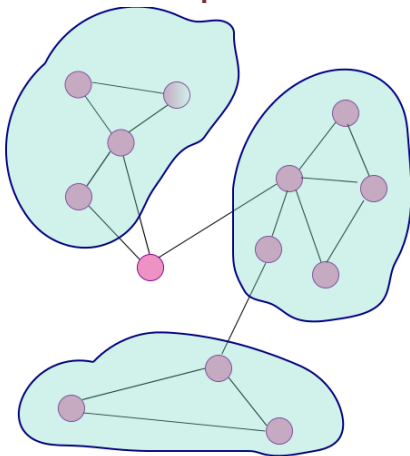
- **Construction gloutonne d'une partition**



Description de l'approche de classification retenue

Étape 1 : Obtention d'une première partition

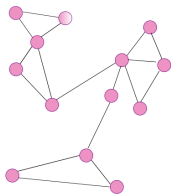
- **Construction gloutonne d'une partition**



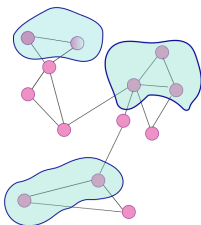
Description de l'approche de classification retenue

Étape 1 : Obtention d'une première partition

- **Construction gloutonne d'une partition**
- Au cours de l'algorithme, **conservation des étapes correspondant à une diminution de 25% du nombre de classes**

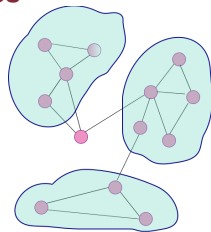


Niveau 1



Niveau 2

...
...

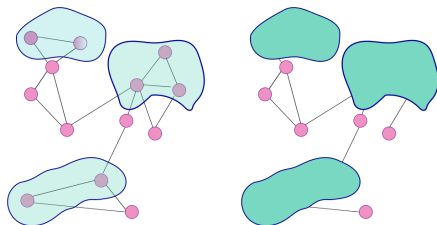


Niveau L (final)

Description de l'approche de classification retenue

Étape 1 : Obtention d'une première partition

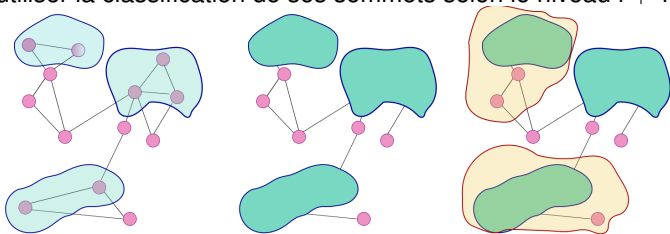
- **Construction gloutonne d'une partition**
- Au cours de l'algorithme, **conservation des étapes correspondant à une diminution de 25% du nombre de classes**
- **Raffinement multi-niveaux** des niveaux $l = L - 1 \rightarrow 1$
 - Déterminer le graphe induit par la classification au niveau $l...$



Description de l'approche de classification retenue

Étape 1 : Obtention d'une première partition

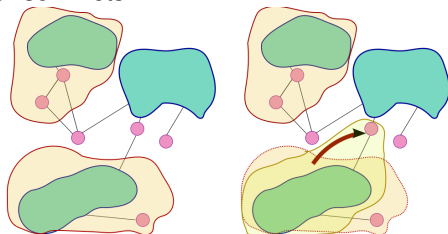
- **Construction gloutonne d'une partition**
- Au cours de l'algorithme, **conservation des étapes correspondant à une diminution de 25% du nombre de classes**
- **Raffinement multi-niveaux** des niveaux $l = L - 1 \rightarrow 1$
 - Déterminer le graphe induit par la classification au niveau l ...
 - ... et utiliser la classification de ses sommets selon le niveau $l + 1$...



Description de l'approche de classification retenue

Étape 1 : Obtention d'une première partition

- **Construction gloutonne d'une partition**
- Au cours de l'algorithme, **conservation des étapes correspondant à une diminution de 25% du nombre de classes**
- **Raffinement multi-niveaux** des niveaux $l = L - 1 \rightarrow 1$
 - Déterminer le graphe induit par la classification au niveau $l...$
 - ... et utiliser la classification de ses sommets selon le niveau $l + 1...$
 - ... pour effectuer des changements de classes opportunistes de certains « super-sommets »



Description de l'approche de classification retenue

Étape 1 : Obtention d'une première partition

- **Construction gloutonne d'une partition**
- Au cours de l'algorithme, **conservation des étapes correspondant à une diminution de 25% du nombre de classes**
- **Raffinement multi-niveaux** des niveaux $l = L - 1 \rightarrow 1$
- **Contrôle de connexité** : au niveau L , vérifier la connexité et scinder les classes non connexes selon leurs composantes connexes

Description de l'approche de classification retenue

Étape 1 : Obtention d'une première partition

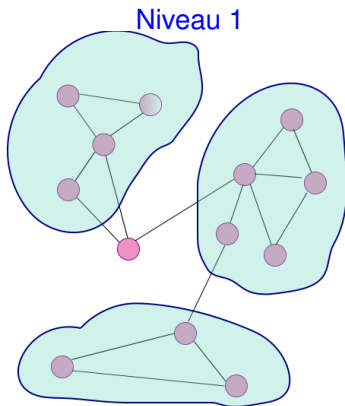
Étape 2 : Itérations jusqu'à stabilisation

- Fusion éventuelle des classes de la classification de niveau L ;
- Raffinement multi-niveaux aux niveaux 1 , L et $L + 1$;
- Contrôle de connexité.

Obtention d'une hiérarchie de classifications

But : Limiter le défaut de résolution de la modularité.

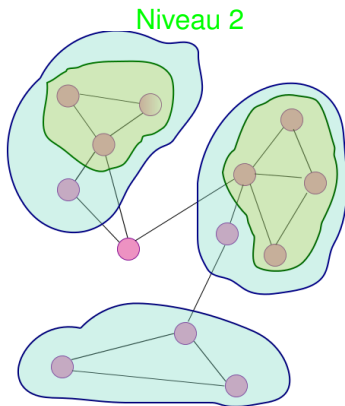
Principe : Itérer l'optimisation de la modularité dans chacune des classes trouvées.



Obtention d'une hiérarchie de classifications

But : Limiter le défaut de résolution de la modularité.

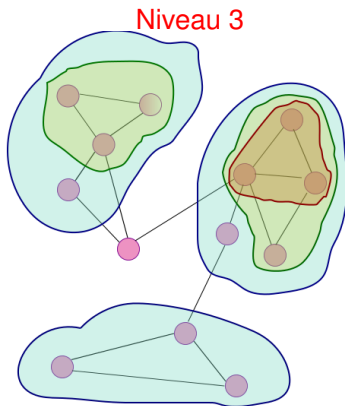
Principe : Itérer l'optimisation de la modularité dans chacune des classes trouvées.



Obtention d'une hiérarchie de classifications

But : Limiter le défaut de résolution de la modularité.

Principe : Itérer l'optimisation de la modularité dans chacune des classes trouvées.



Critère d'arrêt de la hiérarchie

Limite des algorithmes de classification : ils fournissent toujours un résultat !

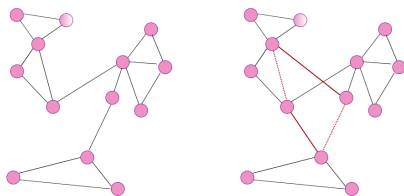
Critère d'arrêt de la hiérarchie

Limite des algorithmes de classification : ils fournissent toujours un résultat !

Significativité d'une partition :

- 1 **Générer des graphes aléatoires** dans l'ensemble des graphes de même distribution de degrés ;

Approche utilisée : algorithme MCMC de **[Roberts Jr., 2000]** par permutations aléatoires d'arêtes du graphe observé



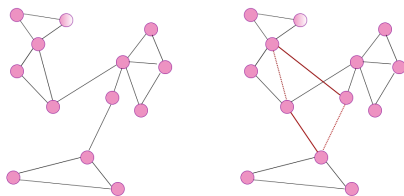
Critère d'arrêt de la hiérarchie

Limite des algorithmes de classification : ils fournissent toujours un résultat !

Significativité d'une partition :

- 1 **Générer des graphes aléatoires** dans l'ensemble des graphes de même distribution de degrés ;

Approche utilisée : algorithme MCMC de **[Roberts Jr., 2000]** par permutations aléatoires d'arêtes du graphe observé



Après $Q|E|$ permutations, obtention d'un graphe aléatoire pour la distribution uniforme dans l'ensemble des graphes de distribution de degrés fixée.

Critère d'arrêt de la hiérarchie

Limite des algorithmes de classification : ils fournissent toujours un résultat !

Significativité d'une partition :

- 1 **Générer des graphes aléatoires** dans l'ensemble des graphes de même distribution de degrés ;
- 2 Chercher l'optimum de la modularité ;
- 3 Déterminer la **p -value de l'optimum selon la distribution empirique** ;

Critère d'arrêt de la hiérarchie

Limite des algorithmes de classification : ils fournissent toujours un résultat !

Significativité d'une partition :

- 1 **Générer des graphes aléatoires** dans l'ensemble des graphes de même distribution de degrés ;
- 2 Chercher l'optimum de la modularité ;
- 3 Déterminer la **p -value de l'optimum selon la distribution empirique** ;
- 4 Si **maximum de la modularité observée significatif** (supérieure à la modularité maximum de 100 graphes aléatoires), on conserve la partition ; sinon, on arrête le partitionnement.

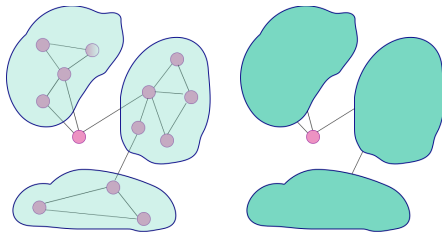
Plan

- 1 Visualisation de graphes : pourquoi ?
- 2 Visualisation par classification hiérarchique
 - Obtention d'une hiérarchie de classifications
 - Visualisation d'une hiérarchie de classifications**
 - Exemples et comparaisons

Principe général de la visualisation d'une classification

À partir d'un graphe $\mathcal{G} = (V, E, W)$ et d'une classification C_1, \dots, C_K ,

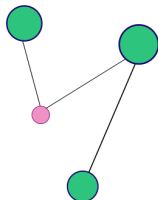
- **déterminer le graphe induit** par la classification :
- sommets : classes ;
- arêtes pondérées par la somme des arêtes entre les classes.



Principe général de la visualisation d'une classification

À partir d'un graphe $\mathcal{G} = (V, E, W)$ et d'une classification C_1, \dots, C_K ,

- **déterminer le graphe induit** par la classification :
 - sommets : classes ;
 - arêtes pondérées par la somme des arêtes entre les classes.
- **représenter** le graphe induit :
 - classes représentées par un symbole de surface proportionnelle à $|C|$;
 - arêtes représentées par un trait d'épaisseur proportionnelle à la contribution à la modularité.



Représenter une hiérarchie de classifications

Principe général

- partir de la classification la plus grossière
- développer les classes par ordre de plus faible réduction de la modularité.

Représenter une hiérarchie de classifications

Principe général

- partir de la classification la plus grossière
- développer les classes par ordre de plus faible réduction de la modularité.

Problèmes

- 1 tenir compte de la taille des sommets : **[Tunkelang, 1999]**
(modification des forces de l'algorithme de Fruchterman et Reingold)

Représenter une hiérarchie de classifications

Principe général

- partir de la classification la plus grossière
- développer les classes par ordre de plus faible réduction de la modularité.

Problèmes

- 1 tenir compte de la taille des sommets : **[Tunkelang, 1999]**
(modification des forces de l'algorithme de Fruchterman et Reingold)
- 2 anticiper l'espace nécessaire au développement d'une classe donnée : estimation récursive de l'espace nécessaire à la représentation par la méthode précédente (en partant du niveau le plus fin)

Représenter la contribution à la modularité des liens entre classes

Fusion des classes C_i et C_j induit une modification de modularité égale à

$$\Delta Q_{C_i, C_j} = \frac{1}{m} \left(W_{C_i, C_j} - \frac{d(C_i)d(C_j)}{2m} \right)$$

Représenter la contribution à la modularité des liens entre classes

Fusion des classes C_i et C_j induit une modification de modularité égale à

$$\Delta Q_{C_i, C_j} = \frac{1}{m} \left(W_{C_i, C_j} - \frac{d(C_i)d(C_j)}{2m} \right)$$

Proposition

- arêtes telles que $\Delta Q_{C_i, C_j} < 0$ en pointillés bleus (lien faible au sens de la modularité) d'épaisseur égale à 1 ;
- arêtes telles que $\Delta Q_{C_i, C_j} > 0$ en rouge (lien fort au sens de la modularité) d'épaisseur comprise entre 1 et 5 (proportionnellement à la valeur) ;

Plan

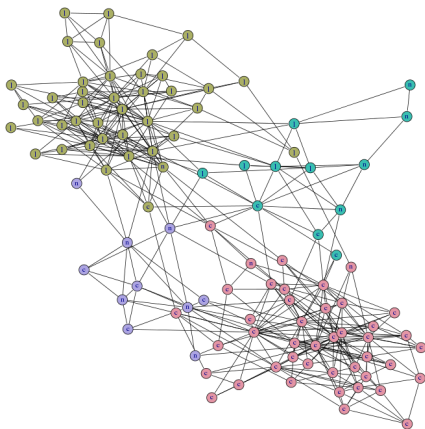
- 1 Visualisation de graphes : pourquoi ?
- 2 Visualisation par classification hiérarchique
 - Obtention d'une hiérarchie de classifications
 - Visualisation d'une hiérarchie de classifications
 - Exemples et comparaisons

« Political books »

Description :

- sommets : 105 livres politiques américains ;
- arêtes pondérées par le nombre d'achats communs sur amazon.

Représentation FDP

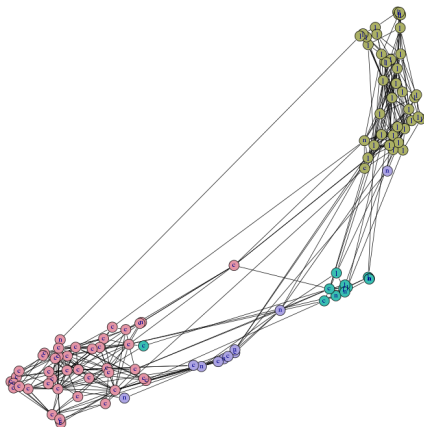


« Political books »

Description :

- sommets : 105 livres politiques américains ;
- arêtes pondérées par le nombre d'achats communs sur amazon.

Représentation LinLog

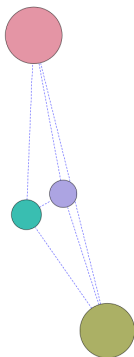


« Political books »

Description :

- sommets : 105 livres politiques américains ;
- arêtes pondérées par le nombre d'achats communs sur amazon.

Représentation hiérarchique (niveau 1/2)

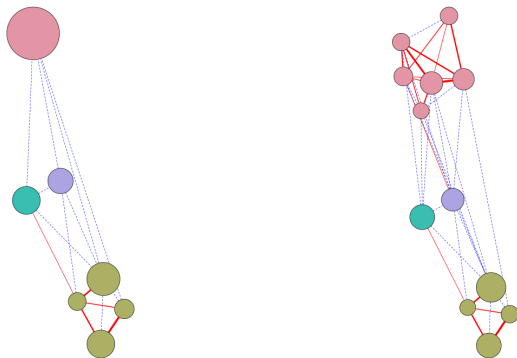


« Political books »

Description :

- sommets : 105 livres politiques américains ;
- arêtes pondérées par le nombre d'achats communs sur amazon.

Représentation hiérarchique (niveau 2/2)



Visualisation d'un grand graphe

Description : graphe issu d'un corpus d'archives médiévales

<http://graphcomp.univ-tlse2.fr/> :

- sommets : transactions du corpus et personnes actives (seigneurs, tenanciers). 3 918 individus impliqués dans 6 455 transactions (total : **10 373 sommets**) ;
- arêtes modélisent l'implication active d'une personne dans une transaction.

Visualisation d'un grand graphe

Description : graphe issu d'un corpus d'archives médiévales

<http://graphcomp.univ-tlse2.fr/> :

- sommets : transactions du corpus et personnes actives (seigneurs, tenanciers). 3 918 individus impliqués dans 6 455 transactions (total : **10 373 sommets**) ;
- arêtes modélisent l'implication active d'une personne dans une transaction.

Optimisation de la modularité : 48 classes composées de 10 à 740 sommets.

Visualisation d'un grand graphe

Description : graphe issu d'un corpus d'archives médiévales

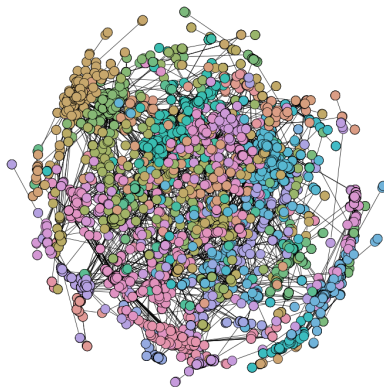
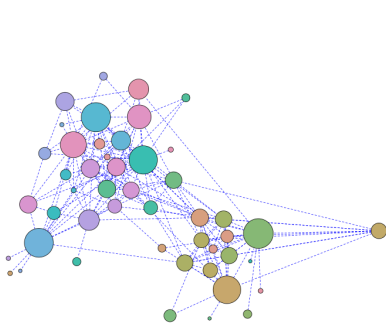
<http://graphcomp.univ-tlse2.fr/> :

- sommets : transactions du corpus et personnes actives (seigneurs, tenanciers). 3 918 individus impliqués dans 6 455 transactions (total : **10 373 sommets**) ;
- arêtes modélisent l'implication active d'une personne dans une transaction.

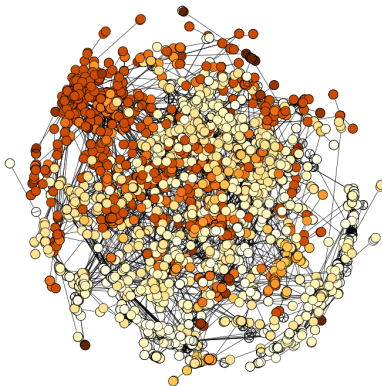
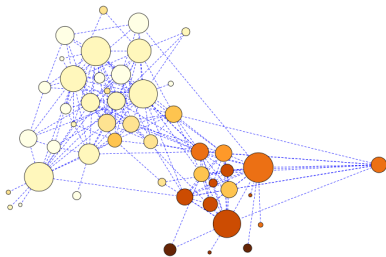
Optimisation de la modularité : 48 classes composées de 10 à 740 sommets.

Hiérarchie : 4 niveaux (limitée à 100 classes maximum) : 89 classes retenues sur le niveau le plus fin.

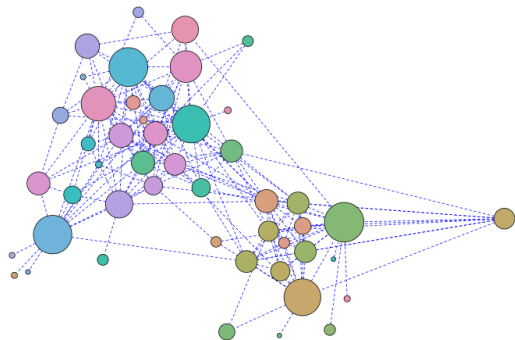
Visualisation d'un grand graphe



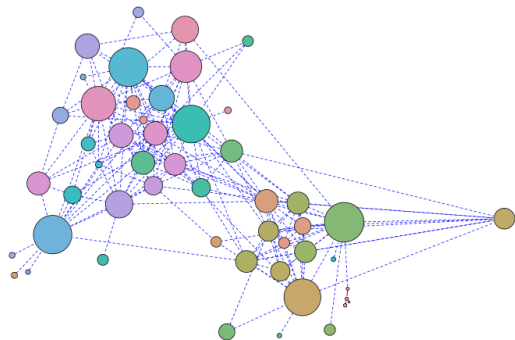
Visualisation d'un grand graphe



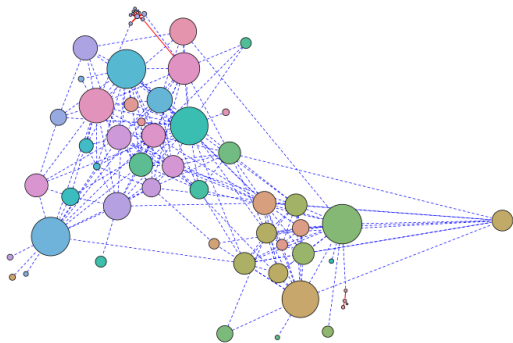
Visualisation d'un grand graphe



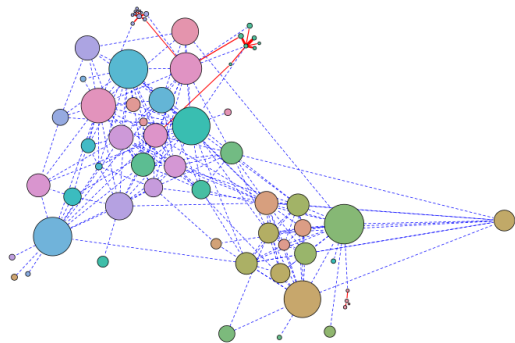
Visualisation d'un grand graphe



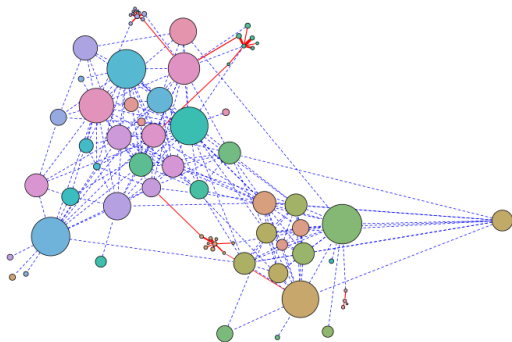
Visualisation d'un grand graphe



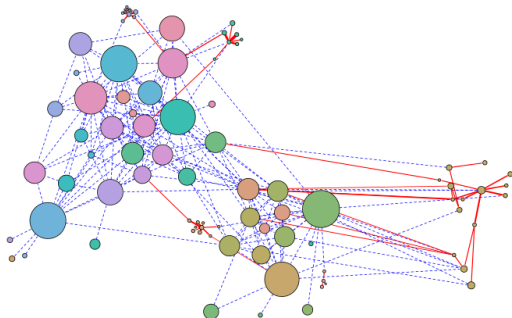
Visualisation d'un grand graphe



Visualisation d'un grand graphe



Visualisation d'un grand graphe



Conclusion et limites

Résumé des avantages de l'approche

- rapide
- totalement automatisée

Conclusion et limites

Résumé des avantages de l'approche

- rapide
- totalement automatisée

Principale limite : estimation récursive de la place occupée par une classe parfois surestimée \Rightarrow arêtes inutilement longues et biais de visualisation

Conclusion et limites

Résumé des avantages de l'approche

- rapide
- totalement automatisée

Principale limite : estimation récursive de la place occupée par une classe parfois surestimée \Rightarrow arêtes inutilement longues et biais de visualisation

Merci pour votre attention

Références



Auber, D., Chiricota, Y., Jourdan, F., and Melançon, G. (2003).
Multiscale visualization of small world networks.
In *INFOVIS'03*.



Auber, D. and Jourdan, F. (2005).
Interactive refinement of multi-scale network clusterings.
In *International Conference on Information Visualisation, International Conference*,
pages 703–709, Los Alamitos, CA, USA. IEEE Computer Society.



Blondel, V., Guillaume, J., Lambiotte, R., and Lefebvre, E. (2008).
Fast unfolding of communities in large networks.
Journal of Statistical Mechanics : Theory and Experiment, P10008 :1742–5468.



Fortunato, S. and Barthélémy, M. (2007).
Resolution limit in community detection.
In *Proceedings of the National Academy of Sciences*, volume 104, pages 36–41.
doi :10.1073/pnas.0605965104 ; URL :
<http://www.pnas.org/content/104/1/36.abstract>.



Fruchterman, T. and Reingold, B. (1991).
Graph drawing by force-directed placement.
Software-Practice and Experience, 21 :1129–1164.



Herman, I., Melançon, G., and Scott Marshall, M. (2000).

Graph visualization and navigation in information visualisation.

IEEE Transactions on Visualization and Computer Graphics, 6(1) :24–43.



Noack, A. (2007).

Energy models for graph clustering.

Journal of Graph Algorithms and Applications, 11(2) :453–480.



Noack, A. and Rotta, R. (2009).

Multi-level algorithms for modularity clustering.

In *SEA '09 : Proceedings of the 8th International Symposium on Experimental Algorithms*, pages 257–268, Berlin, Heidelberg. Springer-Verlag.



Roberts Jr., J. M. (2000).

Simple methods for simulating sociomatrices with given marginal totals.

Social Networks, 22(3) :273 – 283.



Rossi, F. and Villa-Vialaneix, N. (2010).

Optimizing an organized modularity measure for topographic graph clustering : a deterministic annealing approach.

Neurocomputing, 73(7-9) :1142–1163.



Seifi, M., Guillaume, J., Latapy, M., and Le Grand, B. (2010).



Visualisation interactive multi-échelle des grands graphes : application à un réseau de blogs.

In *Atelier EGC 2010, Visualisation et Extraction de Connaissances*, Hammamet, Tunisie.



Tunkelang, D. (1999).

A Numerical Optimization Approach to General Graph Drawing.

PhD thesis, School of Computer Science, Carnegie Mellon University.

CMU-CS-98-189.