# Évaluation des algorithmes de détection de communautés: Une méthode orientée-tâche

Zied Yakoubi

LIPN CNRS UMR 7030
Université Paris Nord, Villetaneuse, France
prénom.nom@lipn.univ-paris13.fr

18 octobre, JFGG 2012

# Plan

# Plan

# Structure communautaire dans les réseaux complexes

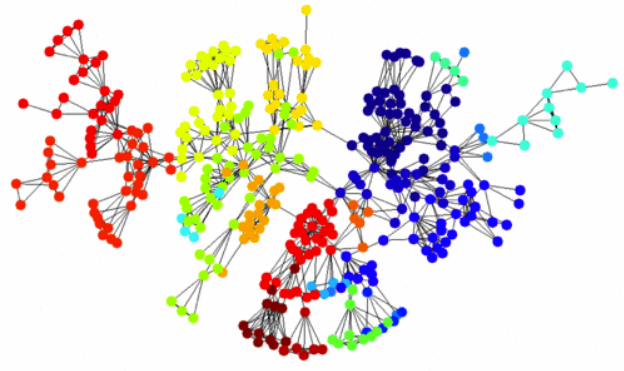


Figure: Les co-auteurs physiciens qui ont publié ensemble

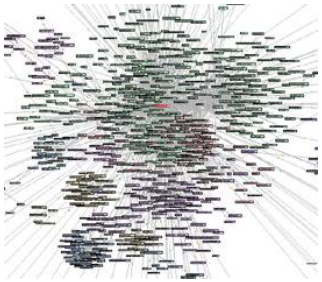


Figure: Corpus des emails d'Enron

## Communauté

- Ensemble de nœuds fortement connectés entre eux et faiblement connectés avec le reste du réseau.
- Un sous-graphe dont les nœuds sont plus liés entre eux qu'avec les autres nœuds

# Détection de communautés : quel algorithme choisir ?

## Algorithmes

FastGreedy(2004),WalkTrap (2007), LPA(2007), Louvain (2008), InfoMap(2008), Licod(2011), etc.

## Critères d'évaluation

- Complexité en temps et en mémoire
- Tailles des communautés retrouvées: micro/macro
- La stabilité des communautés calculées
- La modularité $Q$

# Problématique

## Modularité de Newman (2004)

$$Q(\mathcal{P}) = \sum_{c \in \mathcal{P}} e(\mathcal{C}) - a(\mathcal{C})^2 \quad (1)$$

$e(\mathcal{C}) = \frac{\sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} A_{ij}}{2 \times m_G}$ la fraction des liens à l'intérieur de la communauté $\mathcal{C}$ du graphe, $a(\mathcal{C}) = \frac{\sum_{i \in \mathcal{C}} \sum_{j \in V} A_{ij}}{2.m_G}$ est la fraction des liens incidents à un nœud dans $\mathcal{C}$ du graphe **aléatoire**

- La meilleure partition a la modularité la plus élevée.

## Limites

- Limite de résolution
- **N'identifie pas la topologie des réseaux**

# Plan

# Évaluation fondée sur la réalité du Terrain

## Principe

Comparer la répartition trouvé par l'algorithme avec la classification issue de la réalité de terrain

$P$: Partition trouvée, R: partition réelle

### Pureté

$$Purete(P,R) = \frac{1}{|V|}\sum_{j=1}^{k} max_i(|p_k \cap r_i|) \quad (2)$$

### Rand

$$Rand(P,R) = \frac{a+d}{a+b+c+d} \quad (3)$$

avec a,b,c et d sont respectivement le nombre de paires de nœuds qui sont dans la même communauté en $P$ et $R$, dans la même communauté en $P$ mais dans différentes communautés dans $R$, dans différentes communautés en $P$ mais dans la même communauté en $R$, dans différentes communautés en en $P$ et $R$

# Évaluation fondée sur la réalité du Terrain

## Mutual Information (I)

$$I(P,R) = \sum_i \sum_j U(i,j) \log\left(\frac{U(i,j)}{U(i)U'(j)}\right) \quad (4)$$

où $U'(j) = \frac{|R_j|}{N}$ est l'entropie du $R$, $U(i,j) = \frac{P_i \cap R_j}{N}$

## Normalized Mutual Information (NMI)

$$NMI(P,R) = \frac{I(P,R)}{\sqrt{H(P)H(R)}} \quad (5)$$

$H$ calcule l'entropie conjointe.

# Évaluation fondée sur la réalité du Terrain

**Benchmark réel** Zachary, Strik, Football Pajek datasets [1]

- Réseaux de petite taille

**Benchmark artificiel** Girvan et Newman [Girvan 2002], LFR Benchmark [Lancichinetti et Fortunato (2009)]

- Manque de réalisme

[1] `http://vlado.fmf.uni-lj.si/pub/networks/data/`

# Plan

# Motivation

- Classification non-supervisée $\Longleftrightarrow$ Détection de communautés dans les réseaux complexes
  - *Les liens expriment la similarité entre les nœuds*
- Classification de données $\longrightarrow$ Détection de communautés
  - Zhenping Li et al., 2008
  - Tianbao Yang et al., 2010
- **Classification de données $\longleftarrow$ Détection de communautés**
  - Tatyana et al., 2008
  - Clara et al., 2011

# Principe de l'approche

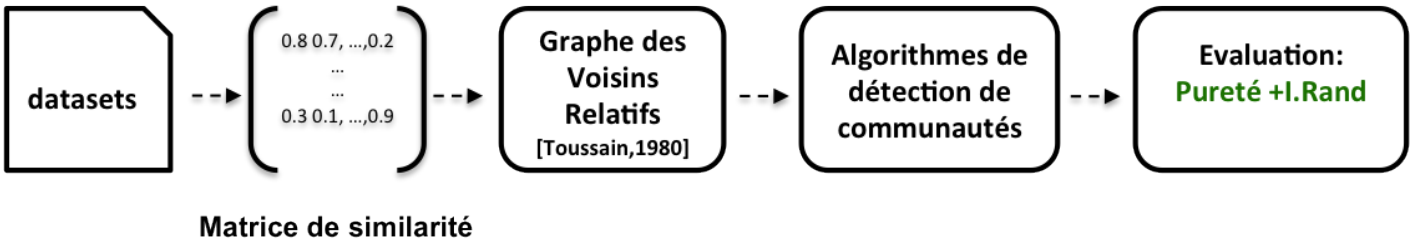



Figure: Application des algorithmes de détection de communautés pour la classification non-supervisée

# Graphes des Voisins Relatifs (GVR)

Etant donnée un graphe G, deux nœuds sont connectés si et seulement si :

$$d(x_i, x_j) \leq \max\{d(x_i, x_l), d(x_j, x_l)\}, \text{ pour tout } l \neq i, j \quad (6)$$

$d(x_i, x_j)$ est la fonction de distance entre les deux nœuds $x_i$ et $x_j$ .

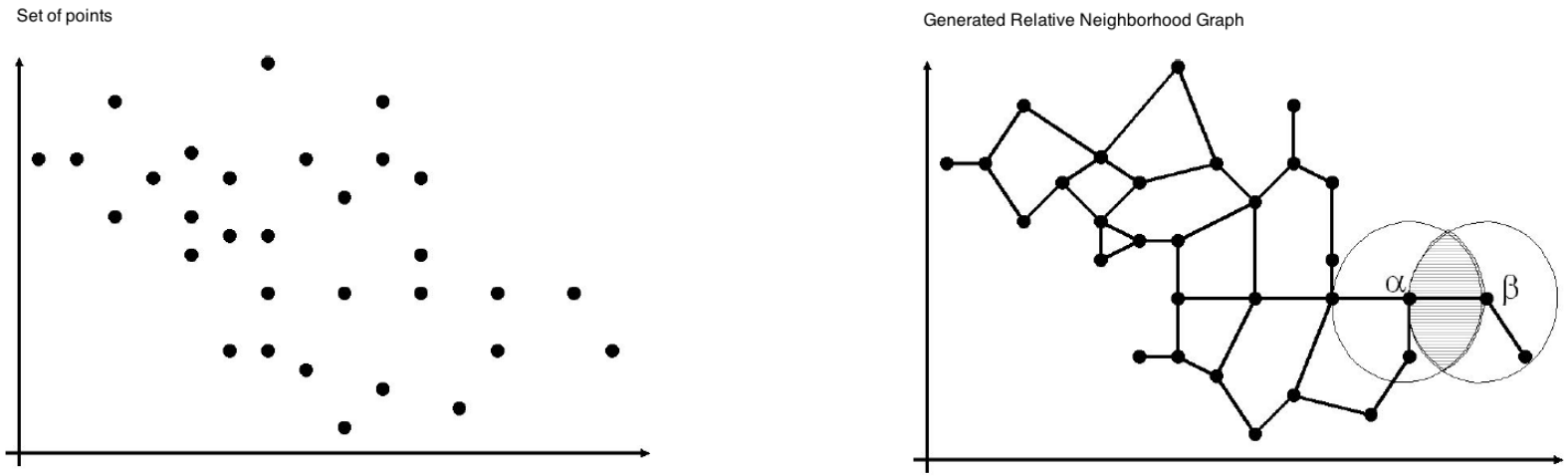



Figure: Exemple de génération d'un GVR à partir d'un jeux de données

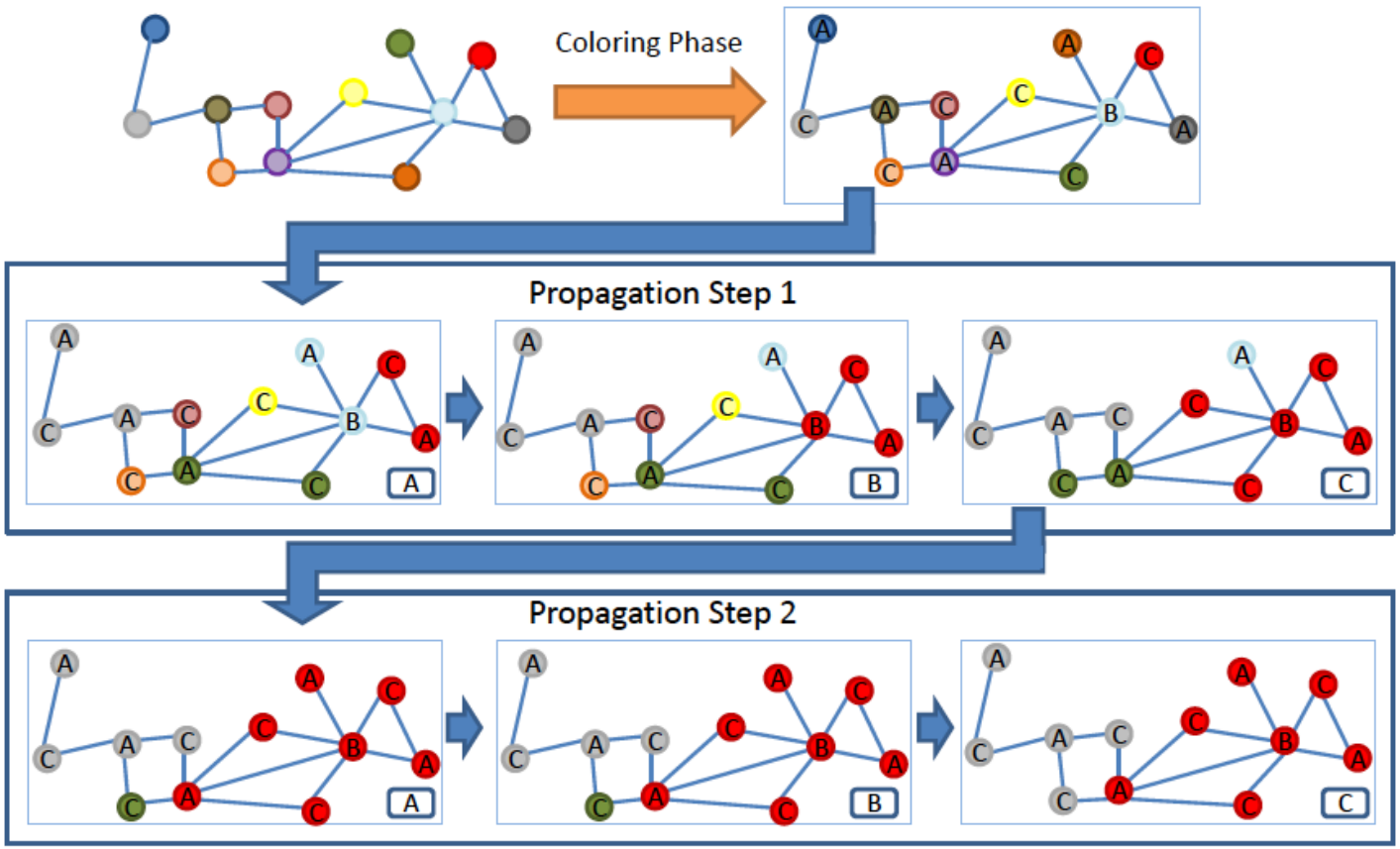
Complexité algorithmique: $O(n^3)$

# Plan

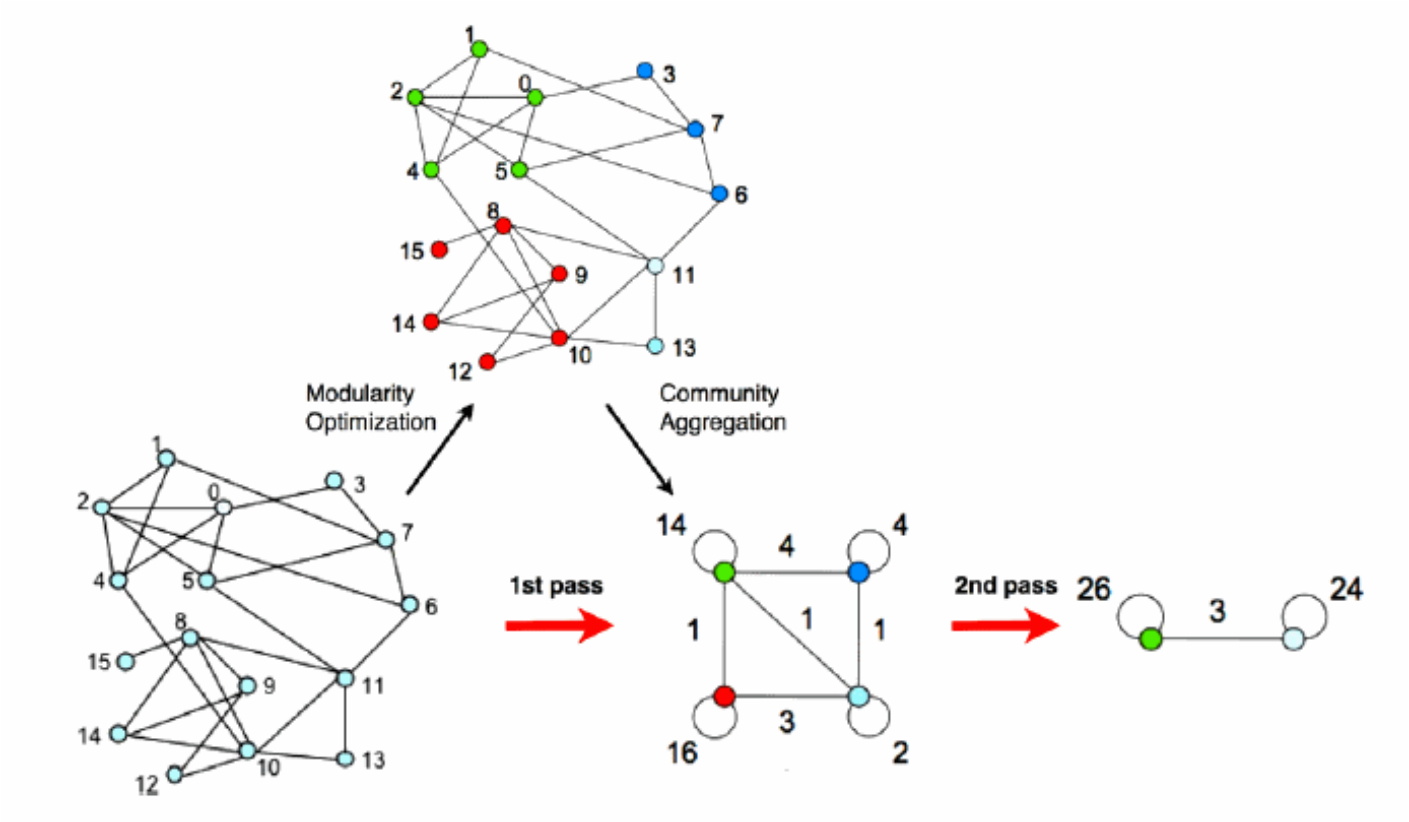
# Algorithmes étudiés

- Algorithme de Propagation de Label (LPA) :version semi-synchrone [ Cordasco et al. (2011)]
- Méthode de Louvain [Blondel et al. (2008)]
- Licod [Kanawati (2011)]

# LPA semi-synchrone



# Méthode de Louvain



# Identification de Leaders pour la détection de communautés (Licod)

## Principe

Une communauté se construit autour d'un ensemble de **Leader**

## L'algorithme

Soient $G =< V, E >$ un graphe, $\Gamma(x \in V)$ est l'ensemble des nœuds voisins de $x$.

1. Déterminer l'ensemble des Leaders $\mathcal{L}$
2. Réduire $\mathcal{L}$ en $\mathcal{C}$ des communautés de Leaders.
3. Chaque $x \in V$ trie les communautés $c \in \mathcal{C}$ dans un ordre décroissant selon le degré d'appartenance $P_x^0$
4. Pour chaque $x \in V$, calculer$P_x^t = \mathit{FusionVotes}(P_y^{t-1} y \in \{X\} \cup \Gamma(x))$

# Jeux de données

| Dataset | Glass | Iris | Wine | Vehicle |
|---|---|---|---|---|
| #Instances | 214 | 150 | 178 | 846 |
| #Attributs | 10 | 4 | 13 | 18 |
| Type d'attributs | Real | Real | Integer, Real | Integer |
| # Classes | 7 | 3 | 3 | 4 |

Table: Caractéristique des jeux de données

# Caractéristiques topologiques des GVRs

| Dataset | Caractéristiques | Euclidean | Cosine | Chebyshev |
|---|---|---|---|---|
| Glass | Densité | 0.012 | 0.012 | 0.170 |
| | Degré moyen | 2.60 | 2.58 | 36.38 |
| | Diamètre | 21 | 24 | 8 |
| | Coef.Clustering | 0.0116 | 0.0101 | 0.2487 |
| Iris | Densité | 0.017 | 0.019 | 0.110 |
| | Degré moyen | 2.56 | 2.84 | 16.45 |
| | Diamètre | 33 | 25 | 14 |
| | Coef.Clustering | 0.0442 | 0.0097 | 0.3336 |
| Wine | Densité | $0.36\ 10^{-}2$ | $0.38\ 10^{-}2$ | $0.56\ 10^{-}2$ |
| | Degré moyen | 3.07 | 3.26 | 4.81 |
| | Diamètre | 63 | 45 | 54 |
| | Coef.Clustering | 0.0024 | 0 | 0.0891 |
| Vehicle | Densité | 0.012 | 0.013 | 0.016 |
| | Degré Moyen | 2.13 | 2.46 | 2.88 |
| | Diamètre | 102 | 59 | 84 |
| | Coef.Clustering | 0 | 0 | 0.1644 |

Table: Caractéristiques topologiques des graphes

# Performances des algorithmes

| Dataset | Algo | Pureté | Rand | Modularité | # com |
|---|---|---|---|---|---|
| Glass | Louvain | 0.36 | 0.74 | **0.72** | 10 |
| | LPA | 0.17 | **0.74** | 0.55 | 64 |
| | LICOD | **0.81** | 0.57 | 0.36 | 2 |
| | K-means | 0.74 | 0.61 | 0 | k=2 |
| Iris | Louvain | 0.37 | 0.76 | 0.77 | 11 |
| | LPA | 0.16 | 0.69 | 0.58 | 40 |
| | LICOD | **0.78** | **0.79** | 0.57 | 5 |
| | K-means | 0.54 | 0.77 | 0 | k=5 |
| Wine | Louvain | 0.25 | **0.68** | 0.84 | 12 |
| | LPA | 0.08 | 0.66 | 0.62 | 59 |
| | LICOD | **0.83** | 0.67 | 0.49 | 2 |
| | K-means | 0.46 | 0.45 | 0 | k=2 |
| Vehicle | Louvain | 0.17 | 0.77 | 0.81 | 16 |
| | LPA | 0.04 | **0.80** | 0.50 | 223 |
| | LICOD | **0.47** | 0.68 | 0.65 | 13 |
| | K-means | 0.34 | 0.69 | 0 | k=13 |

Table: Performance des algorithmes

# Plan

# Conclusion et perspectives

## Conclusion

- Benchmarck produit à partir des données réelles
- Plate-forme pour l'évaluation des algorithmes de détection de communautés
- Limite: Coefficient de clustering faible

## Perspectives

Évaluation orientée prévision des liens

Merci de votre attention