Similarités topologiques de nœuds dans les graphes de terrain

Rushed Kanawati

LIPN, CNRS UMR 7030
Université Paris Sorbonne Cité
http://lipn.fr/∼kanawati
rushed.kanawati@lipn.univ-paris13.fr

January 21, 2014

Plan

Problèmes

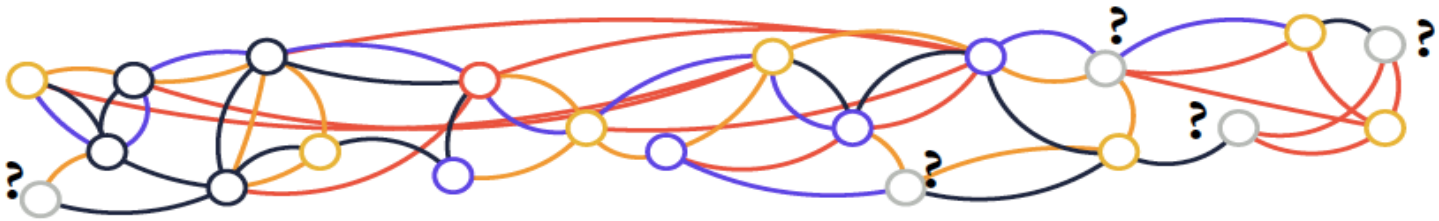
1. Problème 1 : Mesurer la similarité entre deux nœuds d'un graphe

2. Problème 2 : Mesurer la similarité entre nœuds appartenant à deux graphes ayant le même nombre de nœuds (problème de mise en correspondance de graphes)

Applications

1. Prévision de labels/fonctions d'un nœud
2. Prévision de liens (recommendation)
3. Détection de communautés.

Prévision de labels/fonctions I



Problème

Soit G un graph pour lequel on connait la classification/fonction de certains de ses nœuds.

Nous cherchons à trouver les labels/fonctions des nœuds non classifiés.

Prévision de labels/fonctions II

Approche

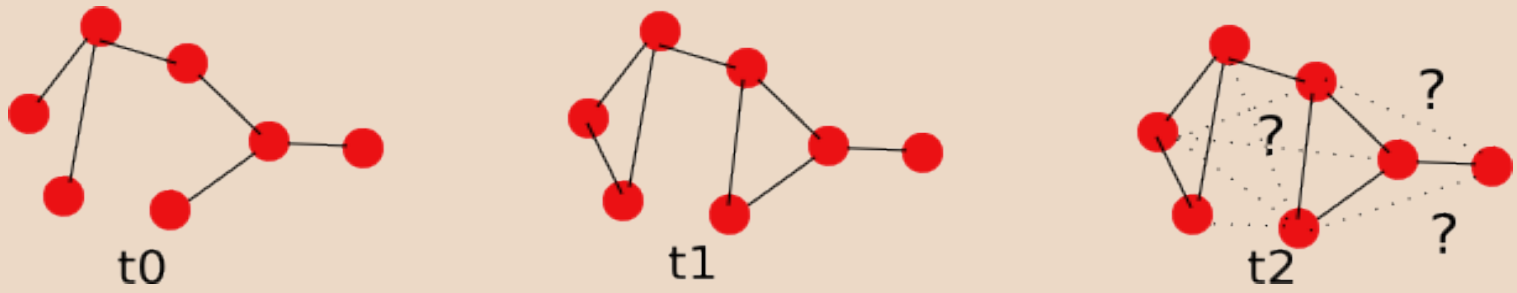
1. Initialiser une matrice de similarité S_{ij} entre les nœuds avec :

$$S_{ij} = \begin{cases} s_{ij} & \text{la similarité fonctionnelle si i et j sont classifiés} \\ \delta_{ij} & \text{si i ou j n'est pas encore classifié} \end{cases}$$

2. Mettre à jour les entrées S_{ij} d'une manière itérative pour i ou j non classifié en appliquant la règle : la classe d'un nœud i est la classe du nœud le plus similaire à i

Prévision de liens

Problème

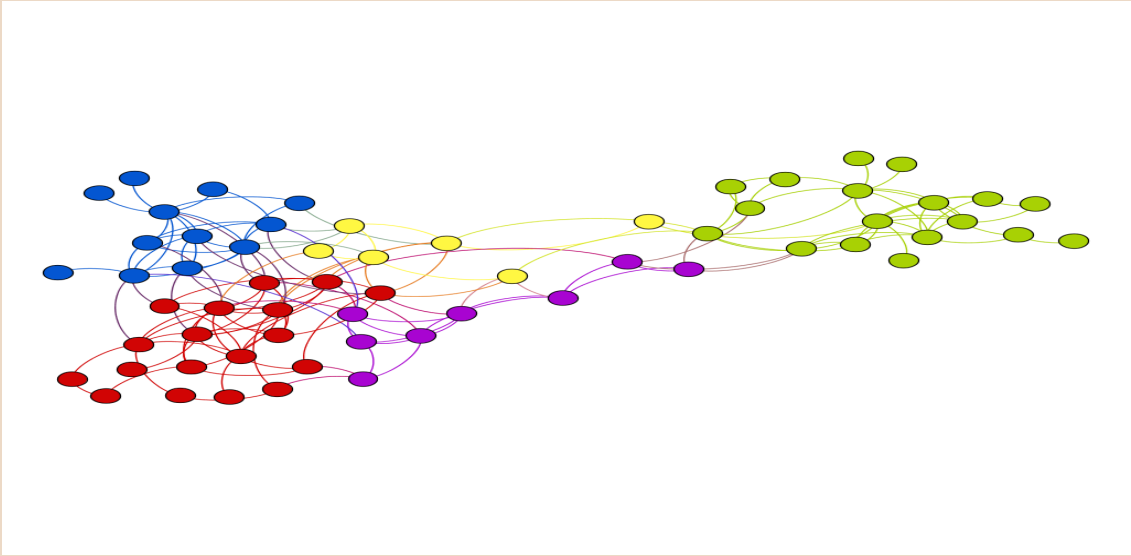


Approche

1. Soit $G_t =< V, E >$ un graphe **connexe**.
2. Soit $sim^m(x, y)$ une mesure de similarité dyadique,
3. Soit $\mathcal{L} = \{(x, y) : x \in V, y \in V, (x, y) \notin E\}$
4. On trie $\mathcal{L}$ en fonction de sim^m, le k-top couples sont prédits.
5. Evaluation : Précision/ Rappel en fonction des liens établis dans G_{t+1}.

Détection de communautés

Problème



Approche

1. Calculer la matrice de similarité $S_{ij} = sim^m(i,j)$
2. Appliquer un algorithme de clustering sur S_{ij} (ex. K-means, DBSCAN, CAH)

Notations : Rappel

Un graphe $G =< V, E \subseteq V \times V >$:

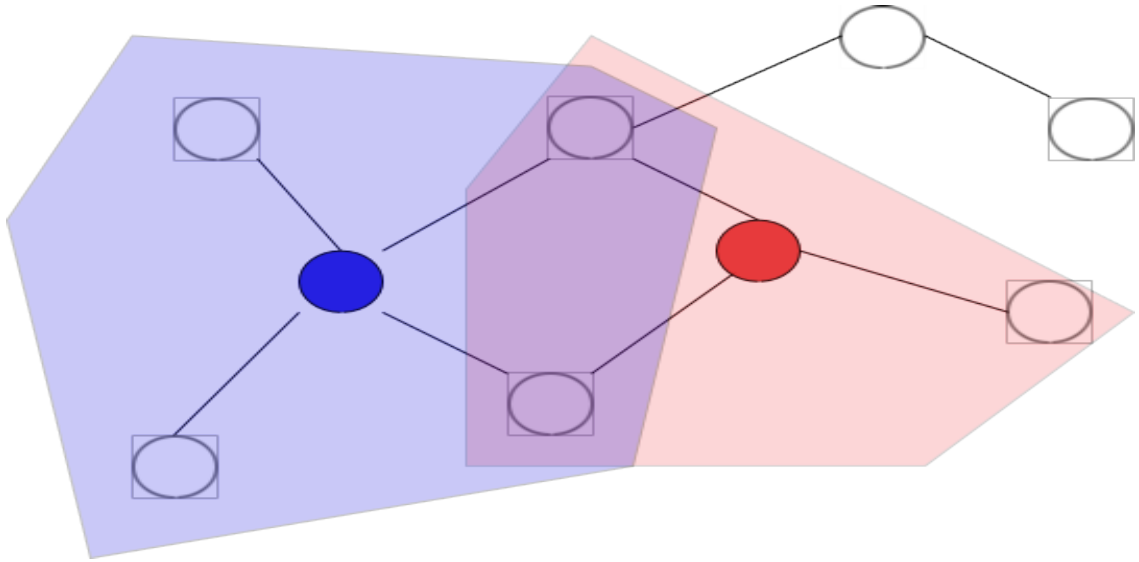
V est l'ensemble de nœudes (i.e. acteurs sociaux)
E est l'ensemble de liens *sociaux*.

Notations :

- A_G est la matrice d'adjacence de G : $a_{ij} \neq 0$ si les nœuds $(v_i, v_j) \in E$, 0 sinon.
- $\Gamma(v)$ est l'ensemble de voisins de v.
  $\Gamma(v) = \{x \in V : (x, v) \in E\}$.
- Le degré d'un nœud $d(v) =\| \Gamma(v) \|$
- $d(x, y)$ est la distance géodésique entre les deux nœuds x et y.

Similarité structurelle : Définition I

- Chaque nœud est similaire à lui même.
- Deux nœuds sont similaires si ils ont des voisins similaires.



Mesures centrées voisinage commun I

Voisins communs (VC)

$$sim^{VC}(x,y) = \| \Gamma(x) \cap \Gamma(y) \|$$

A^2_G

Jaccard

$$sim^{Jaccard}(x,y) = \frac{\|\Gamma(x) \cap \Gamma(y)\|}{\|\Gamma(x) \cup \Gamma(y)\|}$$

`igraph.Graph.similarity_jaccard`

Mesures centrées voisinage commun II

Cosine (ou indice de Salton)

$$sim^{cos}(x,y) = \frac{\|\Gamma(x) \cap \Gamma(y)\|}{\sqrt{\|\Gamma(x)\| \times \|\Gamma(y)\|}}$$

Adamic-Adar (AA)

$$sim^{AA}(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log(\|\Gamma(z)\|)}$$

`igraph.Graph.similarity_inverse_log_weighted`

Mesures centrées voisinage commun III

Allocation de ressource (RA)

$$sim^{RA}(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\|\Gamma(z)\|}$$

Densité du voisinage commun (ND)

$$sim^{ND}(x,y) = \frac{2 \times \|\{(u,v) \in Eu, v \in \Gamma(x) \cap \Gamma(y)\}\|}{\|\Gamma(x) \cap \Gamma(y)\| \times (\|\Gamma(x) \cap \Gamma(y)\| - 1)}$$

Sørensen Index (Dice)

$$sim^{Sørensen}(x,y) = \frac{2 \times \|\Gamma(x) \cap \Gamma(y)\|}{\|\Gamma(x)\| + \|\Gamma(y)\|}$$

`igraph.Graph.similarity_dice`

Mesures centrées voisinage commun IV

HPI (Hub Promoted Index)

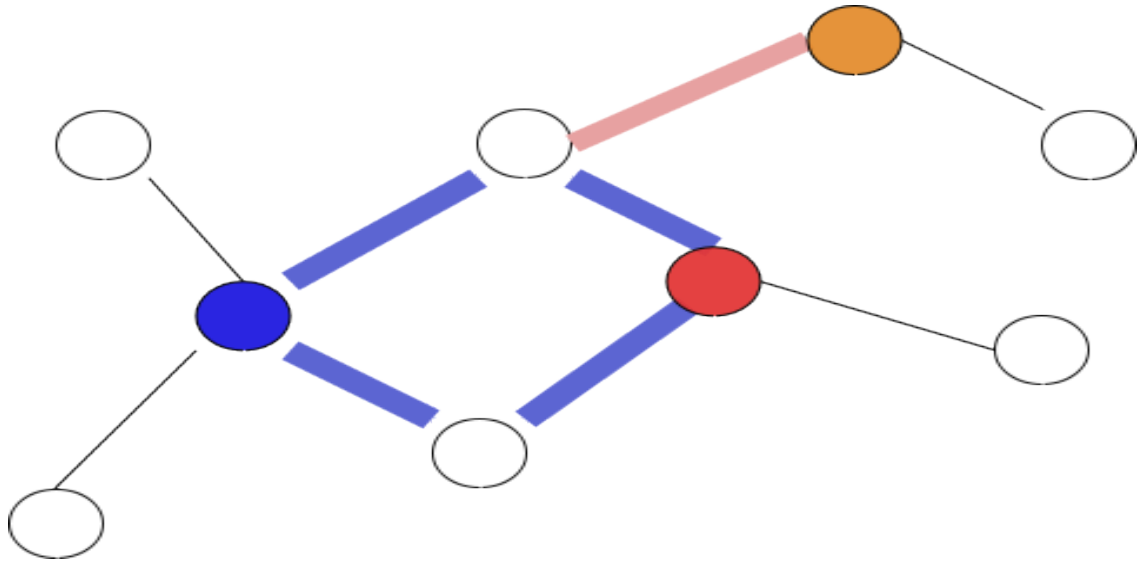
$$sim^{HPI}(x,y) = \frac{\|\Gamma(x) \cap \Gamma(y)\|}{min(\|\Gamma(x)\|, \|\Gamma(y)\|)}$$

HDI (Hub Depressed Index)

$$sim^{HPI}(x,y) = \frac{\|\Gamma(x) \cap \Gamma(y)\|}{max(\|\Gamma(x)\|, \|\Gamma(y)\|)}$$

Similarité structurelle : Définition II

- Chaque nœud est similaire à lui même.
- Deux nœuds sont similaires si ils sont connectés par des chemins courts.



Mesures centrées chemins

1. Mesures basées sur les distances géodésiques
2. Mesures basées sur les marches aléatoires

Mesures centrées chemins I

Proximité

$$sim^{proxi}(x,y) = \frac{1}{dist(x,y)}$$

`igraph.Graph.shortest_paths`

Katz

- $sim^{katz}(x,y) = \sum_{l=1}^{\infty} \beta^l \times \parallel \sigma^l(x,y) \parallel$
- $\sigma^l(x,y)$: nombre de chemins reliant x à y de longueur l.
- $\beta << 1$
- Version Matricielle : $sim^{Katz} = (I - \beta \times A_G)^{-1} - I$
- Version tronquée : $sim^{t-katz} = \sum_{l=1}^{l_{max}} \beta^l A^l$

Mesures centrées chemins II

Indice de chemins locaux (LPI)

$$sim^{LPI} = A^2 + \epsilon A^3$$

Intermédiarité de chemin (PBC)

Soit $\sigma^{dist(x,y)}(x,y)$ l'ensemble de plus courts chemins reliant x et y

l'intermédiarité d'un chemin $p \in \sigma^{dist(x,y)}(x,y)$:

$$BC(p) = \sum_{i,j \in V} \frac{\|\sigma^{dist(i,j)}(i,j|p)\|}{\|\sigma^{dist(i,j)}(i,j)\|}$$

$$sim^{PBC}(x,y) = \max_{p \in \sigma^{dist(x,y)}(x,y)} BC(p)$$

Mesures basées sur les marches aléatoires I

Temps de commutation moyen (CT)

Le temps moyen d'un marcheur alésoir d'aller de x à y puis revenir à x

$$sim^{CT}(x,y) = \frac{1}{L^+_{xx} + L^+_{yy} + 2L^+_{xy}}$$

où $L^+ = (D - A)^{-1}$ est le pseudo-inverse de la matrice laplacienne du graphe cible.

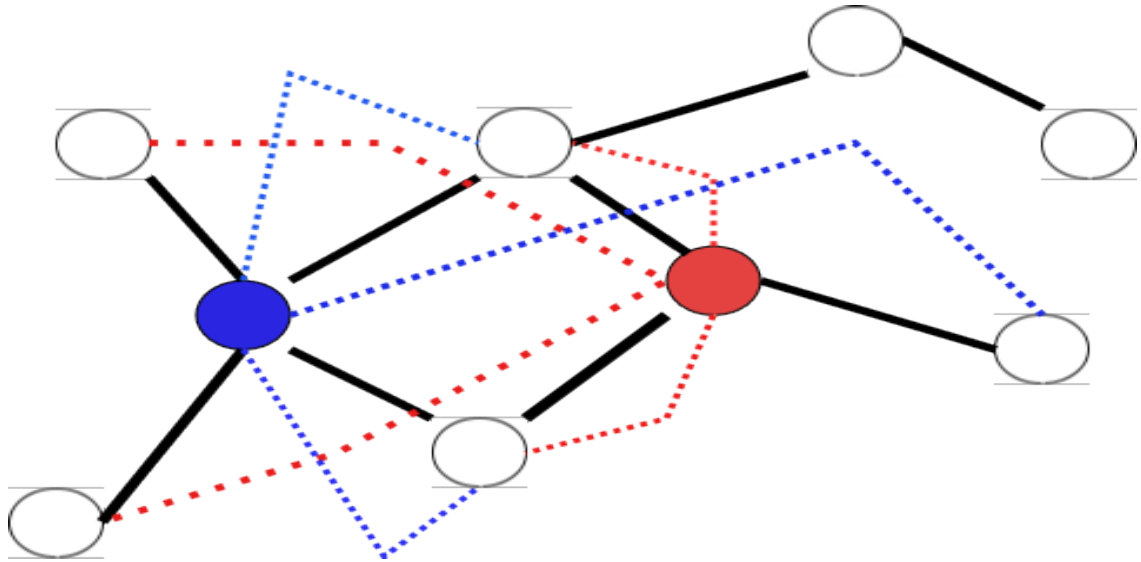
Indice de forêt de marices (MFI)

Le ratio du nombre d'arbres recouvrants (de marches aléatoires) enracinés dans x et contant y sur le nombre total de forêts recouvrant dans le graphe.

$sim^{MFI} = (I + L)^{-1}$

Similarité structurelle : Définition III

- Chaque nœud est similaire à lui même.
- Deux nœuds sont similaires si chacun est similaire aux voisins de l'autre.



TP 2 : I

1. Sur le site du cours; télécharger les deux graphes (en mode pickle) : `dblp72-75` et `dplp72-77`
2. Ecrire une fonction python/igraph `get_giant_cc` qui permet de générer le sous-graphe d'un graphe correspondant au plus grand composante connexe d'un graphe passé en paramètre.
3. Ecrire une fonction qui permet de retrouver les nouveaux liens qui apparaissent dans le graphe `dplp72-77` qui lient des nœuds qui se trouvent dans le plus grand composante connexe de `dblp72-75`
4. Développer des fonctions python/igraph pour le calcul des similarités suivantes : RA, ND, HPI, HDI, Katz tronquée, LPI

TP 2 : II

5 Comparer les précisions de prévision de liens appliqué sur le graphe `dblp72-75`, obtenus par l'emploi de similarité développées dans la question précédente, et aussi les similarités : Jaccard, Adamic-Adar, Dice. La précision est à calculer en fonction des liens retrouvés dans la question 2