Structure et dynamique des réseaux

Clémence Magnien, Lionel Tabourier, Fabien Tarissan

LIP6 – CNRS and Université Pierre et Marie Curie

prenom.nom@lip6.fr

Outline

Outline

Définition et notation

Un graphe $G = (V, E)$ est un couple d'ensembles.

- V est l'ensemble des *sommets* (ou *nœuds*)
- $E \subseteq (V \times V)$ est l'ensemble des *arêtes* (ou *liens*).

On note :

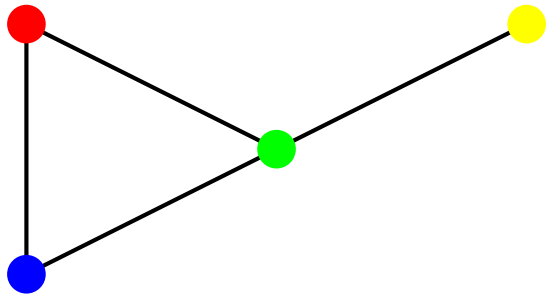
- $n = |V|$ le nombre de sommets
- $m = |E|$ le nombre d'arêtes

u et v sont voisins s'il y a une arête entre eux.

Degré : $d^\circ(v)$: nombre de voisins de v

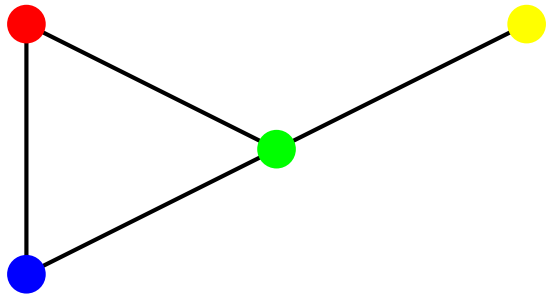
Degré moyen, densité

- degré moyen du graphe,
- densité du graphe,



Degré moyen, densité

- degré moyen du graphe, $d^\circ(G) = \frac{\sum_v d^\circ(v)}{n}$
- densité du graphe, $\delta = \frac{2m}{n(n-1)}$



degrés : **2**, **2**, **3**, **1** ; degré moyen 2

$n = 4$, $m = 4$, $\delta = \frac{8}{12} = 0.66..$

Connexité

Chemin de u à v : suite d'arêtes $(u, v_1), (v_1, v_2), \ldots, (v_{k-1}, v)$
Longueur = k (nombre d'arêtes)

Composante connexe : ensemble maximal de sommets t. q. $\exists$ un chemin entre toutes les paires de sommets.

Graphe connexe : une seule composante connexe

Distributions

Distribution : manière synthétique de représenter une série de valeur.

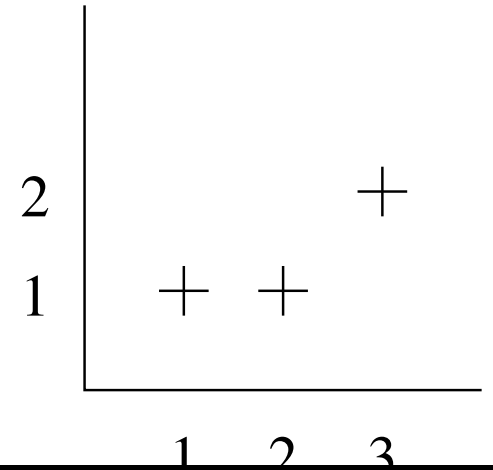
$\rightarrow$ combien de fois la valeur x apparaît dans la série ?

Distributions

Distribution : manière synthétique de représenter une série de valeur.
$\rightarrow$ combien de fois la valeur x apparaît dans la série ?

Exemple/rappel avec la distribution des degrés :
4 nœuds, degrés : 2 3 3 1

$1 \rightarrow 1, 2 \rightarrow 1, 3 \rightarrow 2$



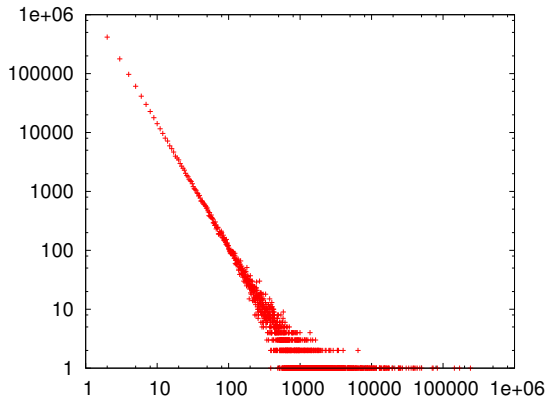
Caractérisation de la distribution

L'un des intérêt : permet de caractériser qualitativement la série de valeurs.

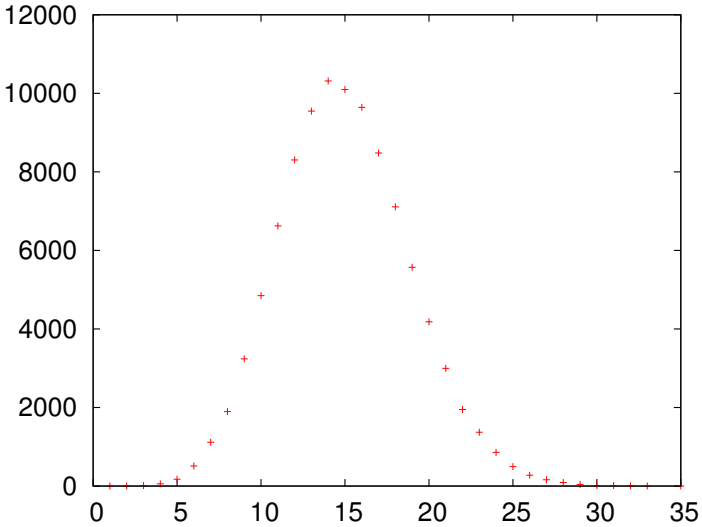
Loi de puissance

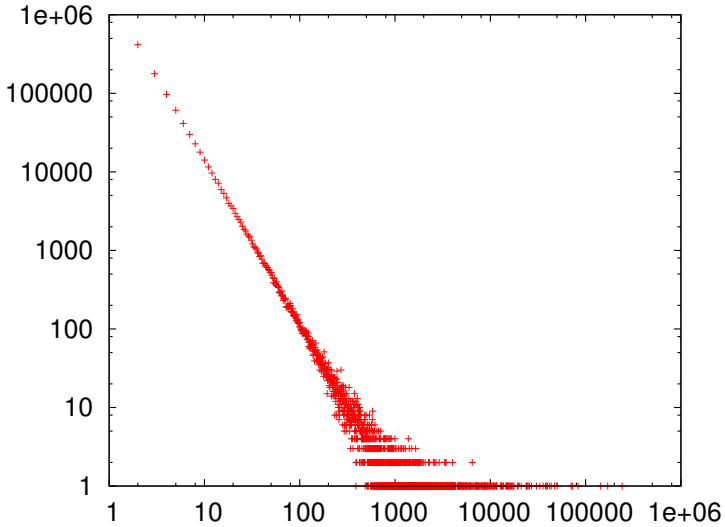
- $N_k \sim k^{-\alpha}$
- droite en échelle log-log

Distribution hétérogène : non homogène (comportements variés), en pratique, souvent proche d'une loi de puissance



Distributions hétérogènes vs homogènes





Homogène

Notion de normalité (et d'exceptions)

Hétérogène

Tous les comportements existent

$\rightarrow$ pas de notion de normalité

Distributions normalisées

Distribution des valeurs, deux choix :

- N_k : nombre d'occurrences de la valeur k
- p_k : proportion de la valeur k dans la série

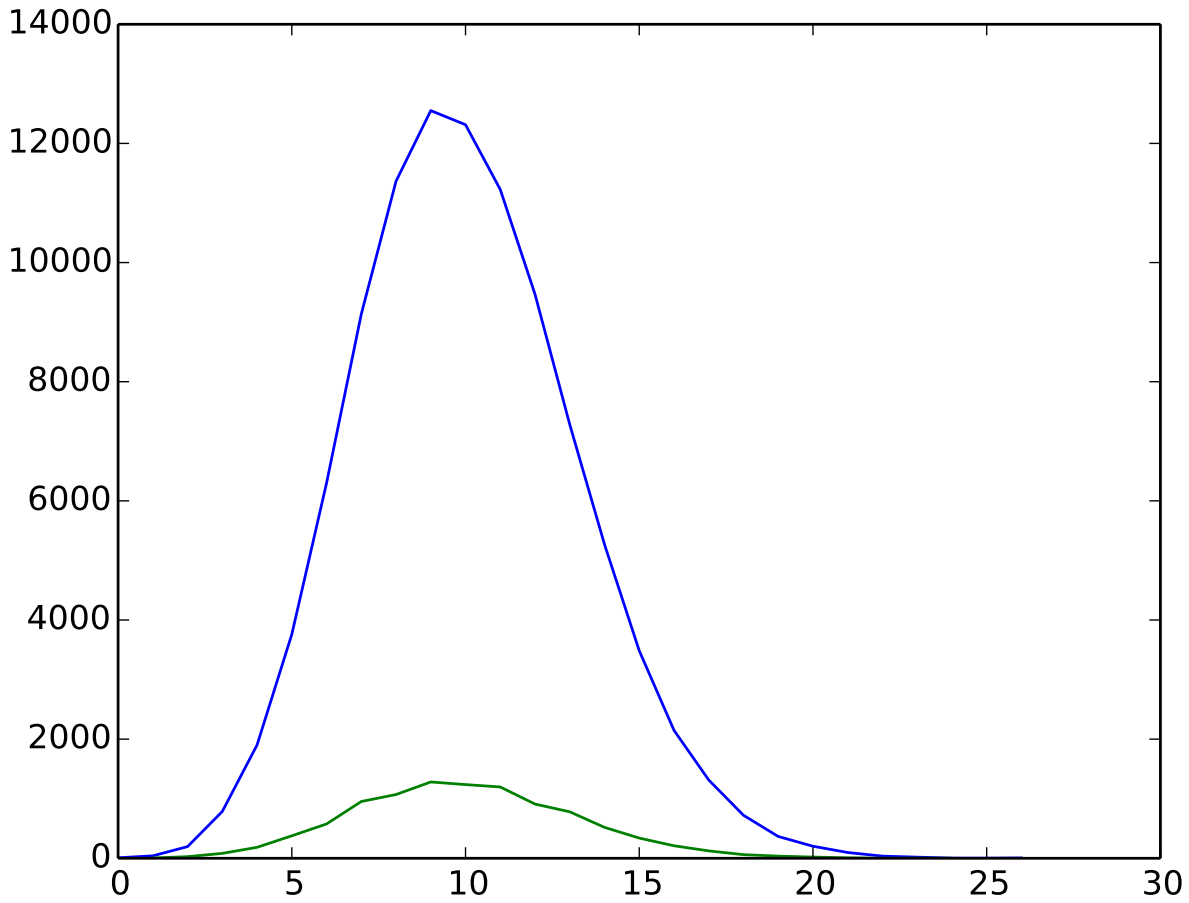
$\rightarrow$ Distribution normalisée

$$p_k = \frac{N_k}{n}$$

A priori, simplement une modification de la valeur en y.

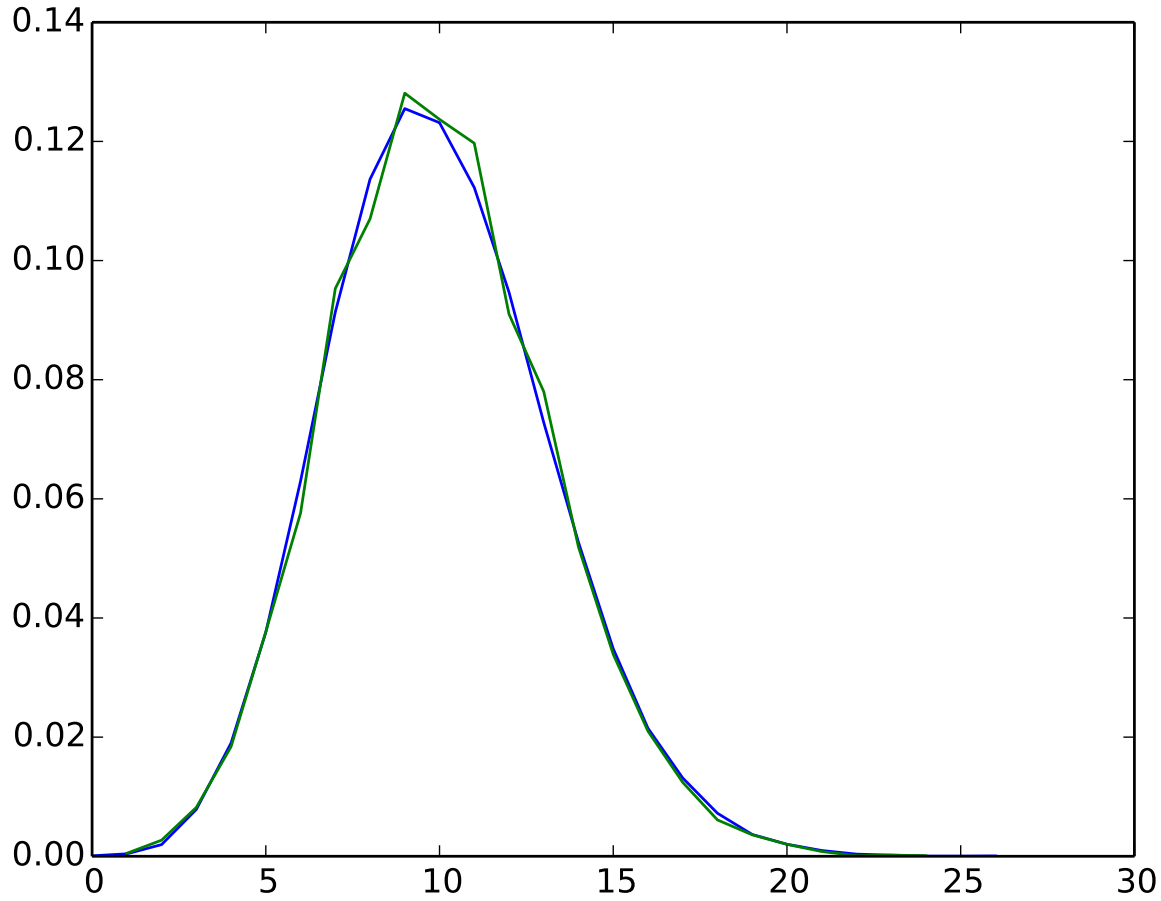
Distributions normalisées

Permet de comparer des graphes de tailles différentes :



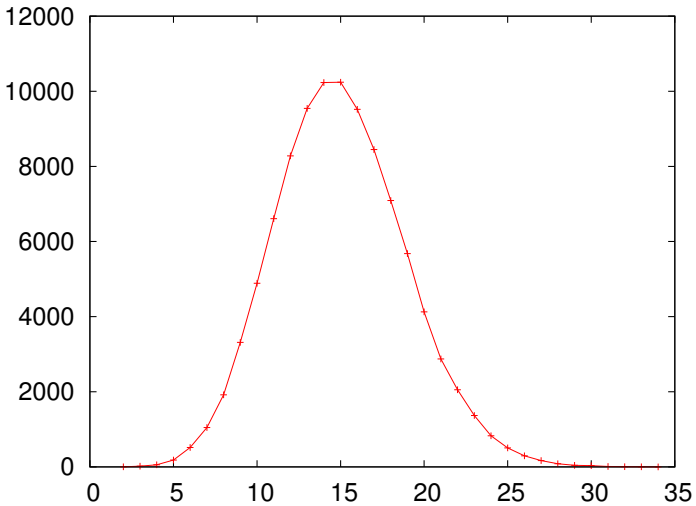
Distributions normalisées

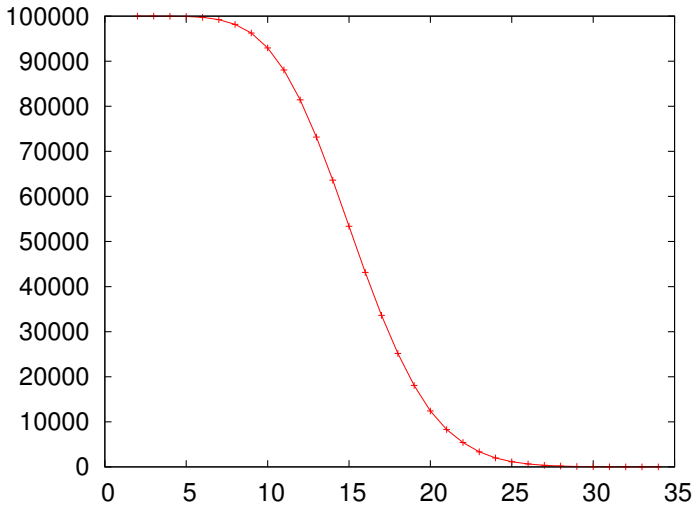
Permet de comparer des graphes de tailles différentes :



Distribution cumulative inverse

- N_k : nombre d'occurrences de valeurs égales k
- C_k : nombre d'occurrences de valeurs supérieures ou égal à k

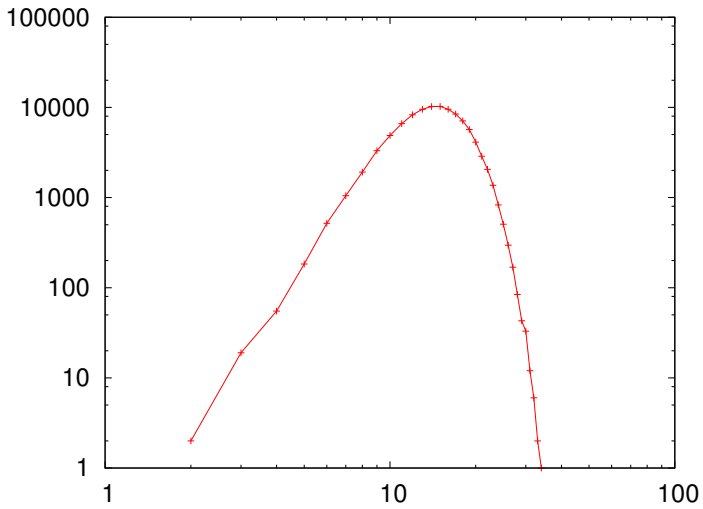


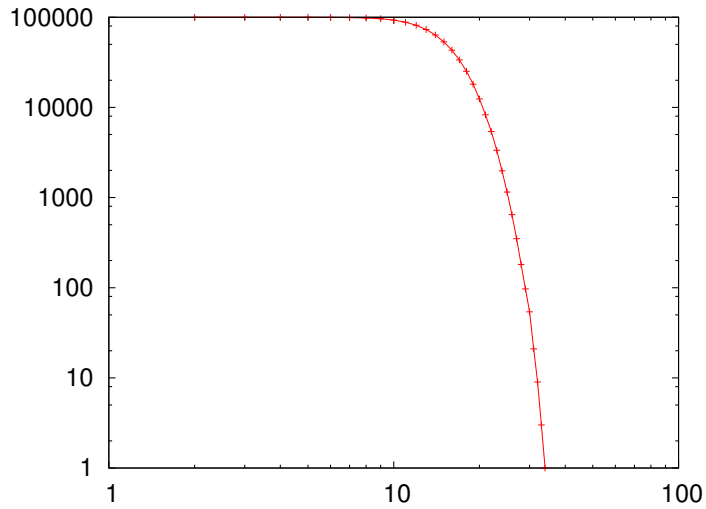


Échelle linéaire

Distribution cumulative inverse

- N_k : nombre d'occurrences de valeurs égales k
- C_k : nombre d'occurrences de valeurs supérieures ou égal à k

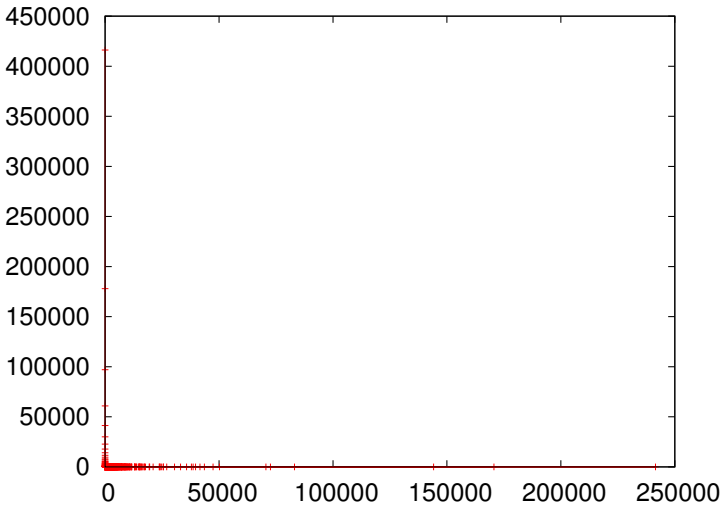


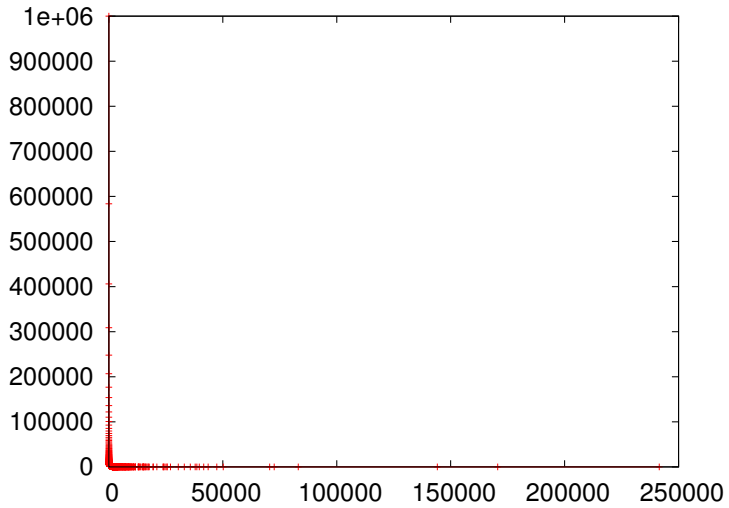


Échelle log-log

Distribution cumulative inverse

- N_k : nombre d'occurrences de valeurs égales k
- C_k : nombre d'occurrences de valeurs supérieures ou égal à k

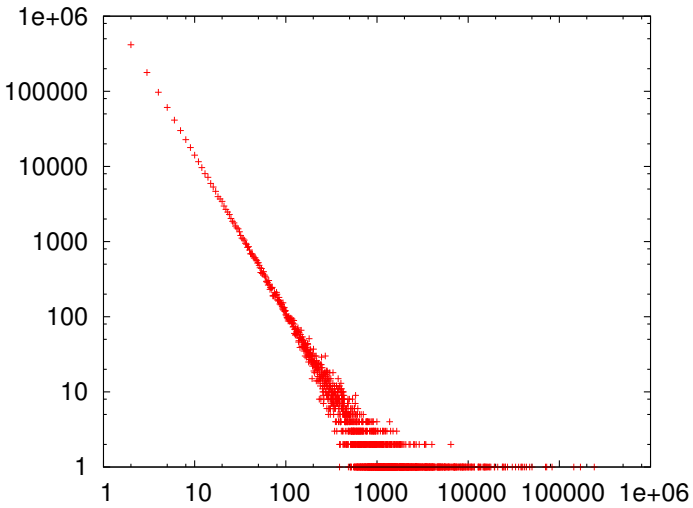


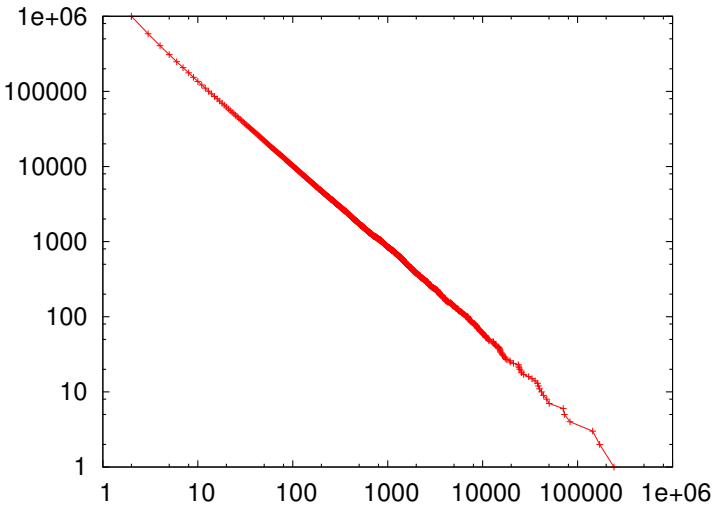


Échelle linéaire

Distribution cumulative inverse

- N_k : nombre d'occurrences de valeurs égales k
- C_k : nombre d'occurrences de valeurs supérieures ou égal à k





Échelle log-log

Distribution cumulative inverse

- N_k : nombre d'occurrences de valeurs égales k
- C_k : nombre d'occurrences de valeurs supérieures ou égal à k

Distributions homogènes/hétérogènes

- Se distingue sur la distribution normale ou cumulative

Ex : loi de puissance

- $N_k \sim k^{-\alpha} \Longrightarrow C_k \sim k^{-\alpha+1}$

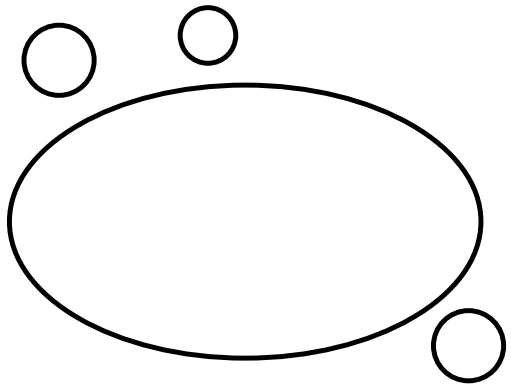
Rq: à rapprocher de $\int x^{-\alpha}\,\mathrm{d}x \sim x^{-\alpha+1}$

Outline

Connexité (rappel)

Pour les graphes de terrain

En général, composante géante
$\rightarrow$ Contient la plupart des sommets



Qu : Comment identifier cette composante principale ?
Comment compter le nombre de composantes connexes dans un graphe ?

Algorithme de parcours en largeur (BFS)

Algorithm 1: Parcours en largeur d'un graphe G à partir d'un sommet s fixé.

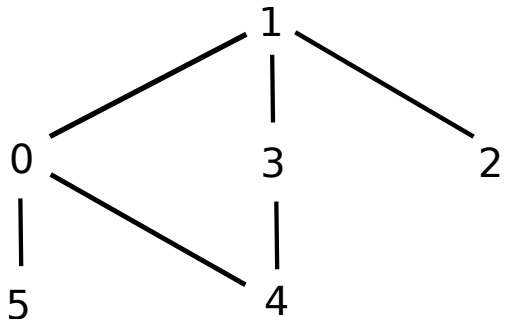
```
begin
    F ← CréerFileVide()
    MettreDansFile(F,s)
    Marquer(s)
    while F est non vide do
        u ← RetirerPremier(F)
        Afficher u
        for v voisin de u dans G do
            if NonMarqué(v) then
                MettreDansFile(F,v)
                Marquer(v)
            end
        end
    end
end
```

Algorithme de parcours en largeur (BFS)

Propriété du parcours en largeur :

- à partir d'un sommet : on voit toute sa composante connexe
  $\rightarrow$ 1 parcours en largeur par composante
- Complexité : $\mathcal{O}(m)$
- Variante mémorisant la filiation : arbre (recouvrant) des plus courts chemins

Exercice



1. Que donne le parcours en largeur du graphe suivant à partir du nœud 3 ?
2. Comment modifier l'algorithme pour qu'il renvoie l'arbre des plus courts chemins à partir d'un nœud fixé ?
3. Appliquer cet algorithme sur l'exemple précédent, à partir du nœud 3 et dessiner l'arborescence associée.

Retour sur la notion de densité locale

Plusieurs moyens de capturer cette notion.
Par exemple :

- coefficient de clustering:
- transitive ratio:

Autrement dit :

Retour sur la notion de densité locale

Plusieurs moyens de capturer cette notion.
Par exemple :

- coefficient de clustering: $\mathtt{cc}(G) = \frac{\sum_v \frac{\Delta(v)}{\vee(v)}}{n}$
- transitive ratio: $\mathtt{tr}(G) = \frac{3\Delta(G)}{\vee(G)}$

Autrement dit :

- coefficient de clustering: calcul d'une valeur pour chaque nœud (de degré ≥ 2) puis moyenne
- transitive ratio: calcul direct

Transitive ratio *vs* Clustering coefficient

Soit $n \in \mathbb{N}$, on nommera G_n les graphes composés de $2n + 1$ sommets et $3n$ liens tels que :

- un unique nœud n_0 est lié à tous les autres nœuds du réseau.
- tous les nœuds autres que n_0 ont un degré 2.

Exercice :

1. Dessiner le cas G_4.
2. Calculez les valeurs des différents coefficients pour G_4.
3. Comment évoluent ces coefficients lorsque n tend vers l'infini.
4. En déduire une interprétation de ces coefficients en terme de probabilités.

Calcul du nombre de triangles

Les deux coefficients reposent sur le nombre de triangles. Comment compter le nombre de triangle auxquels appartient un nœud n ?

Solution naive : Pour toute paire de voisins (u_1, u_2) de v, tester si le lien (u_1, u_2) existe.

Questions :

1. Quelle est la complexité de l'algorithme ?
2. Dans quel cas le calcul est-il coûteux ?
3. Est-ce un problème pour les réseaux rencontrés en pratique ?

Calcul (rapide) du nombre de triangles

Autre point de vue : Soit une arête (u, v) fixée, combien de triangles contiennent cette arête ?

Idée : Il y a un triangle s'il existe un sommet w voisin de u et v.

Solution : Il suffit donc de calculer la taille de l'intersection des voisinages de u et de v. Efficace si les listes de voisins sont triées.

Calcul (rapide) du nombre de triangles

Autre point de vue : Soit une arête (u, v) fixée, combien de triangles contiennent cette arête ?

Idée : Il y a un triangle s'il existe un sommet w voisin de u et v.

Solution : Il suffit donc de calculer la taille de l'intersection des voisinages de u et de v. Efficace si les listes de voisins sont triées.

Algorithm 3: Taille de l'intersection de deux listes triées U et V de taille $d^\circ(u)$ et $d^\circ(v)$

```
i_u = 0
i_v = 0
nb = 0
while (i_u < d°(u)) and (i_v < d°(v)) do
    if U[i_u] < V[i_v] then
        i_u++
    else
        if U[i_u] > V[i_v] then
            i_v ++
        else
            nb++ // un triangle a été trouvé
            i_u++
            i_v++
        end
    end
end
```

Calcul (rapide) du nombre de triangles

Remarques :

- Algorithme long si u ou v a un fort degré
- Le triangle (u, v, w) peut être détecté à partir de 3 intersections de listes

Idée : On va réduire le plus possible la tailles des listes dont on calcule l'intersection :

1. on trie les sommets par degrés décroissants et on renumérote le graphe
2. on ne considère que les liens (u, v) tels que $u < v$
3. on ne cherche les sommets w voisins de u et v que si $w < u$ (et donc $w < v$).

Calcul (rapide) du nombre de triangles

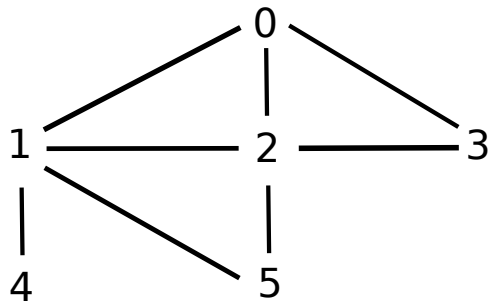
Algorithm 4: Calcul du nombre de triangles

Result: Tableau `tr` qui contient le nombre de triangle de chaque sommet

```
tr: tableau de taille n initialisé à 0;
for u = 0; u < n; u++ do
    for i=0; i < d°(u); i++ do
        v = i-ème voisin de u
        if u < v then
            U = liste des voisins de u
            V = liste des voisins de v
            iu = 0 , iv = 0
            while (iu < d°(u)) and (iv < d°(v)) and (U[iu] < u)
            and (V[iv] < u) do
                if U[iu] < V[iv] then
                    iu++;
                else
                    if U[iu] > V[iv] then
                        iv ++;
                    else
                        tr[u]++;
                        tr[v]++;
                        tr[g->links[u][iu]]++;
                        iu++;
                        iv++;
                    end
                end
            end
        end
    end
end
```

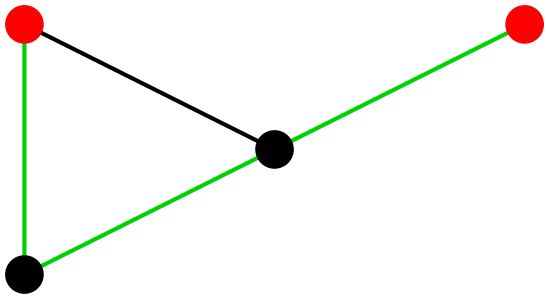
Exercice

Appliquer l'algorithme précédent sur le graphe suivant :



Distances et diamètre (rappel)

chemin de u à v = suite d'arêtes u...v

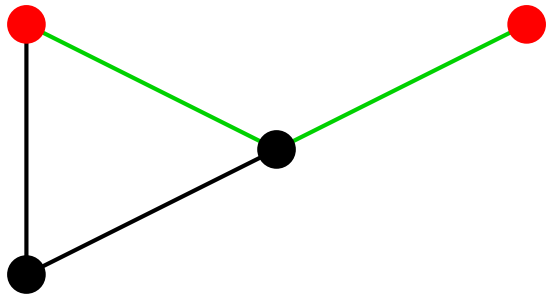


un chemin de longueur 3

Distances et diamètre (rappel)

chemin de u à v = suite d'arêtes u...v

distance $d(u, v)$ = longueur d'*un* plus court chemin



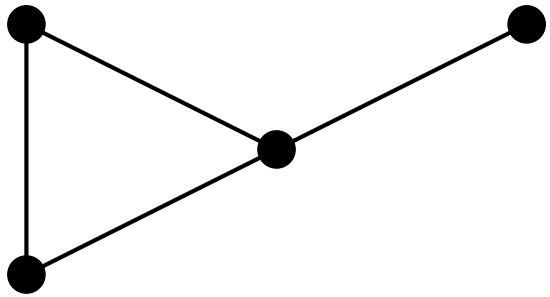
un plus court chemin ; longueur 2 $\Rightarrow$ distance = 2

Distances et diamètre (rappel)

chemin de u à v = suite d'arêtes u...v

distance $d(u, v)$ = longueur d'*un* plus court chemin

distance moyenne = moyenne des distances entre chaque paire de sommets



Distance moyenne = $\frac{8}{6}$

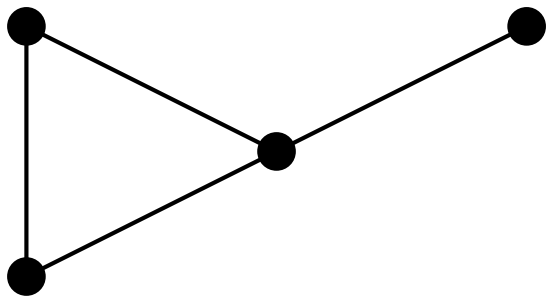
Distances et diamètre (rappel)

chemin de u à v = suite d'arêtes u...v

distance $d(u, v)$ = longueur d'*un* plus court chemin

distance moyenne = moyenne des distances entre chaque paire de sommets

diamètre Δ = plus grande distance (sur ensemble de paires)



diamètre = 2

Calcul des distances

Distance d'un sommet vers tous les autres :
parcours en largeur modifié.

Complexité : $\mathcal{O}(m)$

Calcul des distances

Distance d'un sommet vers tous les autres :
parcours en largeur modifié.

Complexité : $\mathcal{O}(m)$

Distance moyenne, diamètre

Besoin de toutes les distances
$\rightarrow \mathcal{O}(nm)$

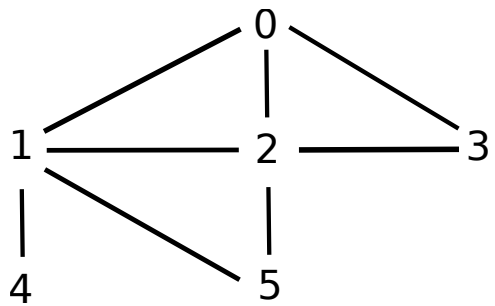
possible d'approximer

Exercice

Que ce soit pour la distance moyenne ou le diamètre, on a besoin de savoir calculer la distance entre 2 nœuds du graphes.

Exercice :

1. Proposer une variante de l'algorithme de BFS pour calculer toutes les distances à partir d'un nœud du graphe.
2. Appliquer l'algorithme précédent sur le graphe suivant (en prenant le nœud 5 comme origine) :



Approximations

Approximation : étant donner une propriété P, estimer cette propriété pour un graphe donné.

Exemples : degré moyen, diamètre moyen, ...

Méthode générique :

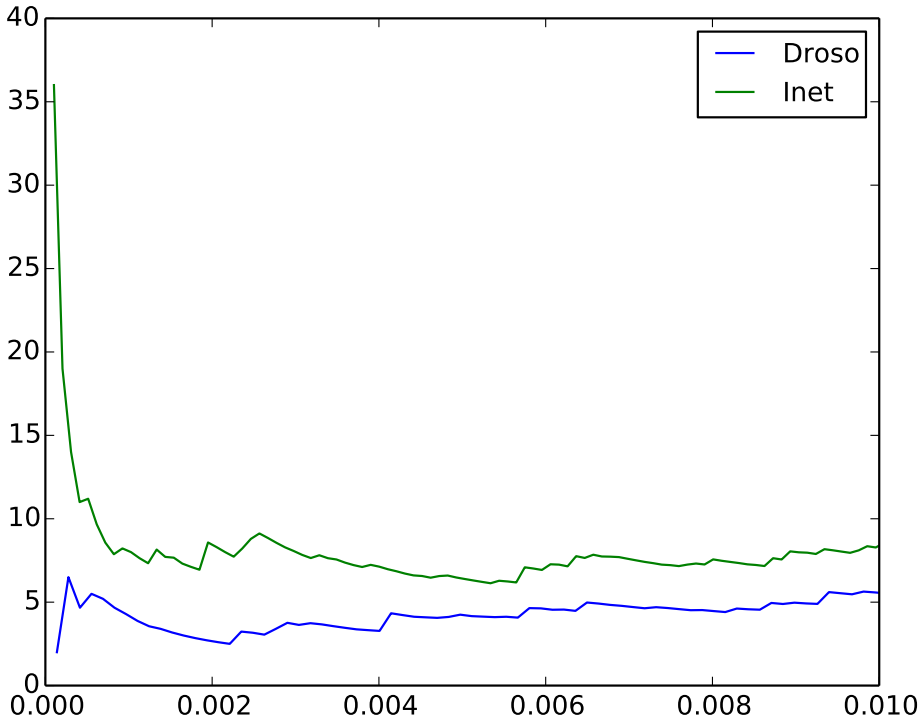
1. Choisir un nœud de G aléatoirement
2. Estimer la propriété recherchée pour v
3. Recommencer à l'étape 1 tant que l'approximation souhaitée n'est pas obtenue

Questions :

- Comment exprimer la notion "d'approximation souhaitée" ?
- Comment savoir si cette approche donne une bonne approximation ou pas ?

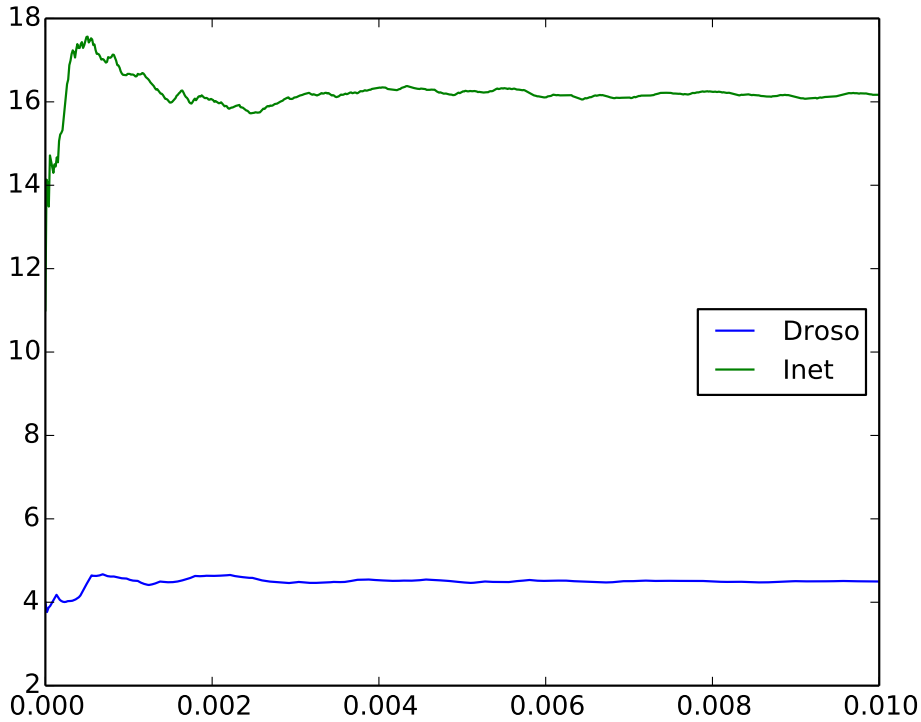
Degré moyen

Application de la méthodologie précédente sur le degré moyen :



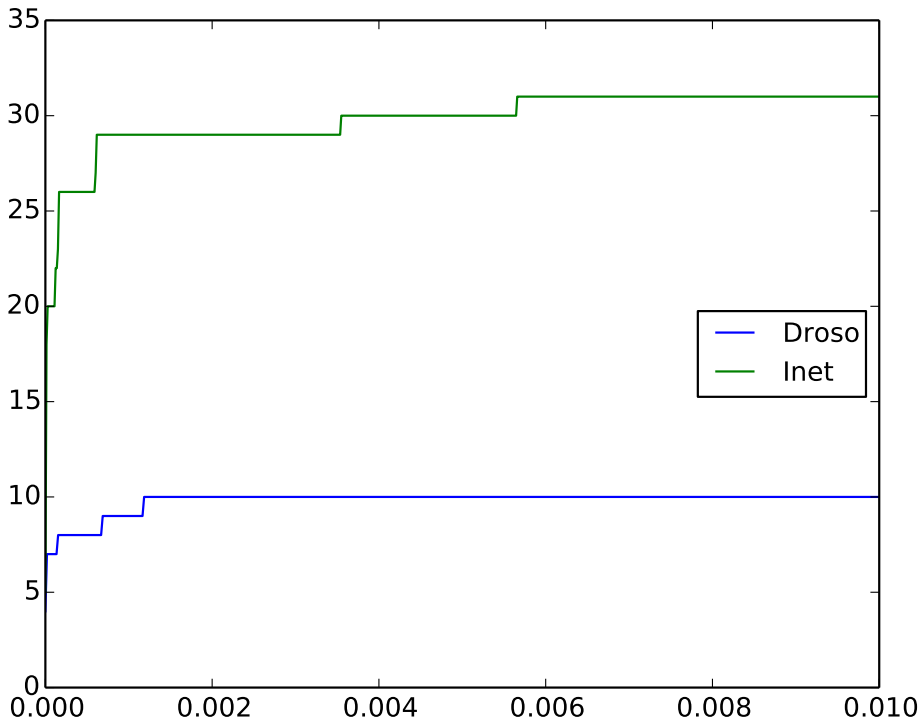
Distance moyenne

Application de la méthodologie précédente sur la distance moyenne :



Distance moyenne

Application de la méthodologie précédente sur le diamètre :



Approximations

Qualité de l'approximation : dépend de la nature de la propriété.

Autre approche :

- Calculer des bornes (inf, sup) de la propriété recherchée
- S'appuyer sur la propriété pour "guider" la recherche

Par exemple : Pour tout nœud v, soit max_v la plus grande des distances à v dans G. Alors le diamètre D du graphe est tel que :

$$max_v \leq D \leq 2max_v$$

Exercice : expliquer pourquoi.

Centralité d'intermédiarité

Soit G un graphe, v, s et t des nœuds du graphe. On appelle *centralité intermédiaire* (betweeness centrality) la valeur :

$$BC(v) = \sum_{s \neq t \neq v} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

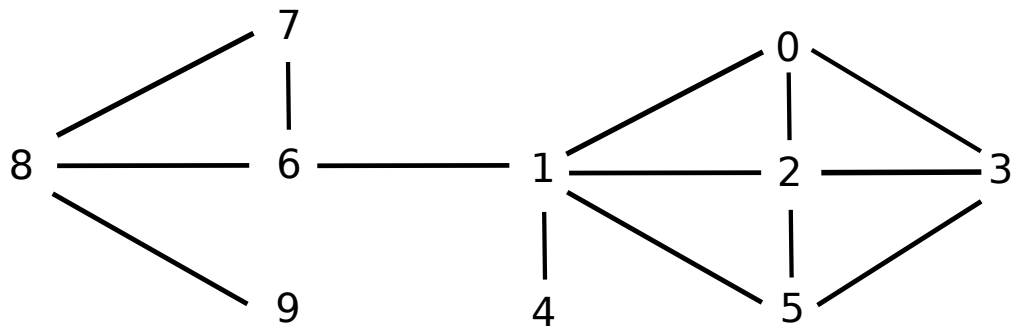
avec:

- σ_{st} : nombre de plus courts chemins entre s et t
- $\sigma_{st}(v)$: nombre de plus courts chemins entre s et t passant par v

Signification

Capture l'importance d'un nœud v dans un graphe en tant qu'intermédiaire :

- rôle dans la propagation de signals/messages/virus
- rôle dans la connectivité du réseau



Exercice : claculer $BC(0)$, $BC(1)$ et $BC(4)$.

Algorithme

Besoin de connaitre **tous** les plus courts chemins entre s et t.

- Parcours en largeur : 1 plus court chemin
- Besoin de modifier l'algorithme. Idée :
  - Calculer un DAG (Directed Acyclic Graph) couvrant
  - S'appuyer sur le calcul des distances
  - On se sert de la profondeur (distance à la racine) pour décider si un nœud déjà marqué fait parti du DAG ou non

Exercice : Calculer le DAG des plus courts chemins partant de 3 dans le graphe précédent. Comparer au résultat d'un BFS partant du même nœud.

Algorithme

Exercice :

- Donner l'algorithme calculant le DAG des plus courts chemins, étant donné un graphe G et un nœud racine s.
- Étant donner un DAG des plus courts chemins, donner une formule permettant de calculer, pour chaque nœud v, le couple (ca, cd) où:
  - ca : le nombre de chemin ascendants de v
  - cd : le nombre de chemins descendants de v
- En déduire le nombre de plus courts chemins partant de s et passant par un nœud v pour tous les nœuds v du DAG
- Appliquer l'algorithme au DAG partant de 3 dans le graphe précédent
- Donner l'algorithme final pour calculer la centralité d'un nœud. Quelle est sa complexité ?

Centralité des liens

Dans l'exemple précédent :

- 1 a une centralité forte
- mais tous les liens partant de 1 n'ont pas la même importance

Il existe une définition identique pour calculer l'importance des liens dans la relation d'intermédiarité :

$$BC(u,v) = \sum_{s \neq t \neq u \neq v} \frac{\sigma_{st}(u,v)}{\sigma_{st}}$$

avec:

- σ_{st} : nombre de plus courts chemins entre s et t
- $\sigma_{st}(u,v)$: nombre de plus courts chemins entre s et t contenant le lien (u,v)

En lien avec la notion de communauté.

Détection de communautés

Exercice : proposer un algorithme de détection de communauté basée sur la centralité d'intermédiarité des liens. Quelle est sa complexité ?

Détection de communautés

Exercice : proposer un algorithme de détection de communauté basée sur la centralité d'intermédiarité des liens. Quelle est sa complexité ?

1. Calculer la centralité de tous les liens
2. Supprimer le lien ayant la centralité la plus élevée
3. Tester si une composante connexe isolée apparaît
4. Répéter l'étape 2 – 3 jusqu'à qu'il n'y ait plus de lien.