

Principe de l'application

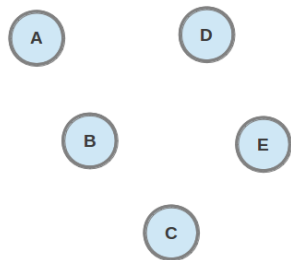
- ▶ Partage entre utilisateurs
 - ▶ texte
 - ▶ image
 - ▶ vidéo
 - ▶ géolocalisation
 - ▶ fichier
- ▶ Recommandations d'événements
 - ▶ personnalisées
 - ▶ possibilité de faire suivre à d'autres utilisateurs
- ▶ Objectif
 - ▶ modéliser les échanges d'informations
 - ▶ établir profils et liens entre utilisateurs
 - ▶ faire des recommandations personnalisées

Graphes

- ▶ Graphe $G = (V, E)$

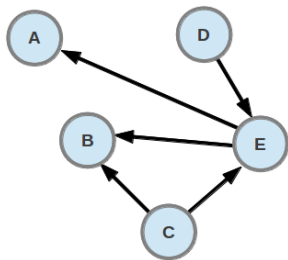
Graphes

- ▶ Graphe $G = (V, E)$
- ▶ Nœuds $V = \{v_1, \dots, v_n\}$



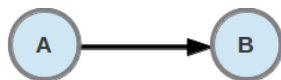
Graphes

- ▶ Graphe $G = (V, E)$
- ▶ Nœuds $V = \{v_1, \dots, v_n\}$
- ▶ Arrêtes $E = \{e_1, \dots, e_m\}$



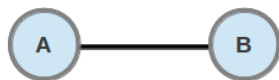
Graphes

- ▶ Graphe $G = (V, E)$
- ▶ Nœuds $V = \{v_1, \dots, v_n\}$
- ▶ Arrêtes $E = \{e_1, \dots, e_m\}$
 - ▶ $e = (u, v) \in V \times V$



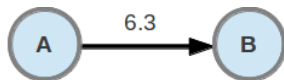
Graphes

- ▶ Graphe $G = (V, E)$
- ▶ Nœuds $V = \{v_1, \dots, v_n\}$
- ▶ Arrêtes $E = \{e_1, \dots, e_m\}$
 - ▶ $e = (u, v) \in V \times V$
 - ▶ $e = \{(u, v), (v, u)\}$



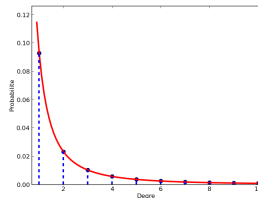
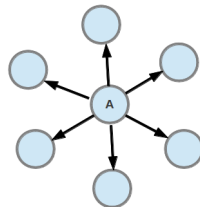
Graphes

- ▶ Graphe $G = (V, E)$
- ▶ Nœuds $V = \{v_1, \dots, v_n\}$
- ▶ Arrêtes $E = \{e_1, \dots, e_m\}$
 - ▶ $e = (u, v) \in V \times V$
 - ▶ $e = \{(u, v), (v, u)\}$
 - ▶ $e = (u, v, l) \in V \times V \times L$



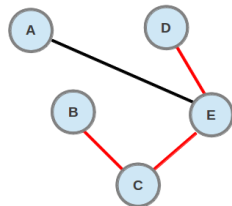
Statistiques

- ▶ Degré :
 - ▶ entrant, sortant
 - ▶ moyenne, minimum, maximum
 - ▶ distribution (loi puissance)



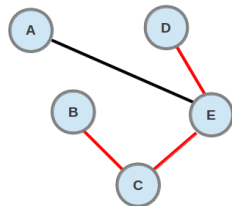
Statistiques

- ▶ Degré :
 - ▶ entrant, sortant
 - ▶ moyenne, minimum, maximum
 - ▶ distribution (loi puissance)
- ▶ Diamètre
 - ▶ réel
 - ▶ efficace



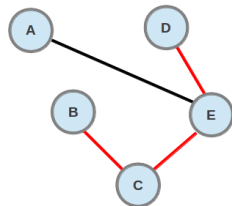
Statistiques

- ▶ Degré :
 - ▶ entrant, sortant
 - ▶ moyenne, minimum, maximum
 - ▶ distribution (loi puissance)
- ▶ Diamètre
 - ▶ réel
 - ▶ efficace
- ▶ Conductance / expansion



Statistiques

- ▶ Degré :
 - ▶ entrant, sortant
 - ▶ moyenne, minimum, maximum
 - ▶ distribution (loi puissance)
- ▶ Diamètre
 - ▶ réel
 - ▶ efficace
- ▶ Conductance / expansion
- ▶ Matrice d'adjacence, matrice laplacienne



Génération de graphes

- ▶ Objectif : pouvoir générer des graphes aux propriétés similaires

Génération de graphes

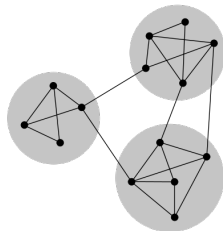
- ▶ Objectif : pouvoir générer des graphes aux propriétés similaires
- ▶ Méthodes
 - ▶ Attachement préférentiel (Newman), Fitness model :
 - ▶ lorsqu'un réseau grandit, les nouveaux nœuds ne se connectent pas aléatoirement
 - ▶ ils se connectent aux nœuds les plus attirants
 - ▶ Kronecker (Leskovec)

Génération de graphes

- ▶ Objectif : pouvoir générer des graphes aux propriétés similaires
- ▶ Méthodes
 - ▶ Attachement préférentiel (Newman), Fitness model :
 - ▶ lorsqu'un réseau grandit, les nouveaux nœuds ne se connectent pas aléatoirement
 - ▶ ils se connectent aux nœuds les plus attirants
 - ▶ Kronecker (Leskovec)
- ▶ Optimiser les paramètres des modèles pour obtenir des graphes aux propriétés similaires.

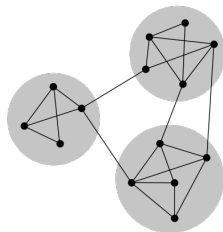
Détection de communautés

- ▶ Idée :
 - ▶ Un graphe peut contenir des sous-graphes très connectés
 - ▶ Deux utilisateurs d'un même sous-graphe peuvent avoir un profil similaire



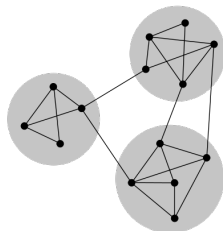
Détection de communautés

- ▶ Idée :
 - ▶ Un graphe peut contenir des sous-graphes très connectés
 - ▶ Deux utilisateurs d'un même sous-graphe peuvent avoir un profil similaire
- ▶ Méthode :
 - ▶ Sélection d'un critère de qualité de partitionnement (modularité)
 - ▶ Sélection des deux sous-graphes maximisant le critère
 - ▶ Itération sur les sous-graphes jusqu'à ce que le critère n'augmente plus



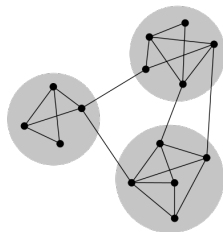
Détection de communautés

- ▶ Idée :
 - ▶ Un graphe peut contenir des sous-graphes très connectés
 - ▶ Deux utilisateurs d'un même sous-graphe peuvent avoir un profil similaire
- ▶ Méthode :
 - ▶ Sélection d'un critère de qualité de partitionnement (modularité)
 - ▶ Sélection des deux sous-graphes maximisant le critère
 - ▶ Itération sur les sous-graphes jusqu'à ce que le critère n'augmente plus
- ▶ Graphe *monolabel* (Newman 2006)



Détection de communautés

- ▶ Idée :
 - ▶ Un graphe peut contenir des sous-graphes très connectés
 - ▶ Deux utilisateurs d'un même sous-graphe peuvent avoir un profil similaire
- ▶ Méthode :
 - ▶ Sélection d'un critère de qualité de partitionnement (modularité)
 - ▶ Sélection des deux sous-graphes maximisant le critère
 - ▶ Itération sur les sous-graphes jusqu'à ce que le critère n'augmente plus
- ▶ Graphe *monolabel* (Newman 2006)
- ▶ Graphe *multilabel* (Lelarge, Massoulié, Xu 2013)



Propagation de l'information

- ▶ Détection de communauté : structure du graphe

Propagation de l'information

- ▶ Détection de communauté : structure du graphe
- ▶ Modéliser la propagation de l'information
 - ▶ Théorie de la survie
 - ▶ Influences dans un réseau, théorie de la percolation

Autres problèmes

- ▶ Statique vs dynamique :
 - ▶ Beaucoup de modèles/méthodes basées sur des graphes statiques
 - ▶ En réalité, $G(t) = (V(t), E(t))$
 - ▶ Nécessité de méthodes adaptées (online)

Autres problèmes

- ▶ Statique vs dynamique :
 - ▶ Beaucoup de modèles/méthodes basées sur des graphes statiques
 - ▶ En réalité, $G(t) = (V(t), E(t))$
 - ▶ Nécessité de méthodes adaptées (online)
- ▶ Parcimonie des recommandations :
 - ▶ Beaucoup de possibilités de recommandations
 - ▶ Ne pas submerger de recommandations
 - ▶ Introduction d'une pénalité