Fouille de graphes

Quentin Fournier <quentin.fournier@polymtl.ca>

Les diapositives ont été créées par Daniel Aloise
<daniel.aloise@polymtl.ca>

17 novembre 2021

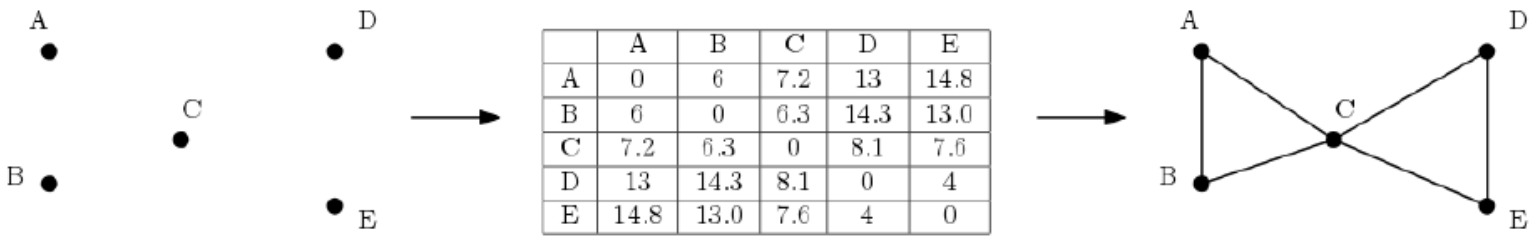
Graphes

- De nombreux jeux de données ont des interprétations graphiques naturelles :

Données	Sommets	Arêtes
Réseaux sociaux	personnes	amitiés
Internet	pages	liens/références
Commerces	produits/clients	ventes
Réseaux génétiques	gènes	interactions

Transformation de données embedding

- Un ensemble de données multidimensionnelles définit des graphes : ajoute une arête (X_i, X_j) si X_i et X_j sont *proches*.



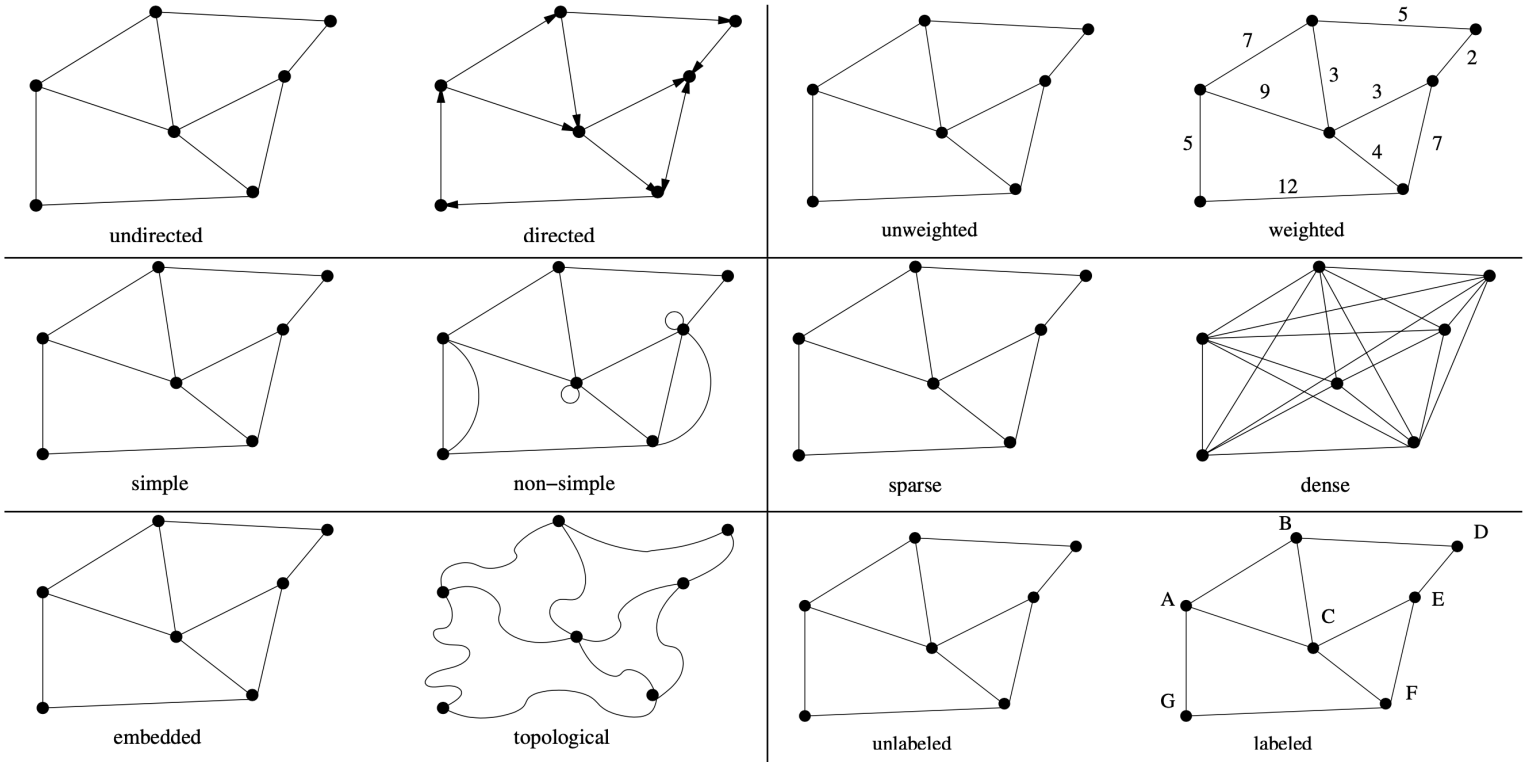
	A	B	C	D	E
A	0	6	7.2	13	14.8
B	6	0	6.3	14.3	13.0
C	7.2	6.3	0	8.1	7.6
D	13	14.3	8.1	0	4
E	14.8	13.0	7.6	4	0

Skienna, 2017

- Les graphes définissent aussi des données multidimensionnelles : transformation spectrale.

Taxonomie des graphes

EXAMEN: savoir les graphes et leurs fonctions!
Le prof l'a expliqué en classe mais pas eu le temps de le noter
Demander qui fait quoi sur Slack



on associe un label à chacun des sommets

Skienna, 2017

Twitter: réseau dirigé: car une personne ne nous follow pas en retour
Fb: qq1 qu'on follow va nous follow aussi

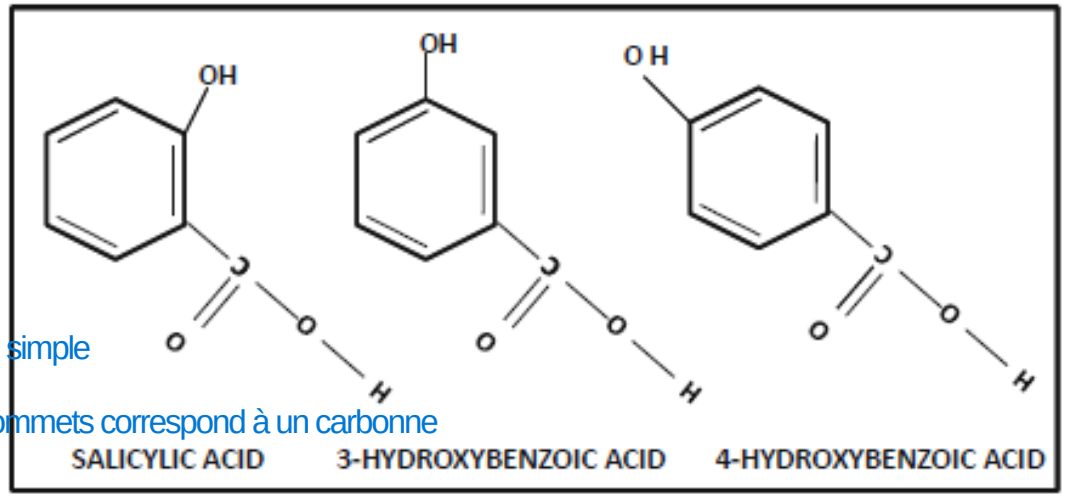
Applications

- Il y a deux types principaux d'applications pour lesquelles la fouille de graphes est naturelle :
  1. Dans des applications telles que les données chimiques et biologiques, une base de données de nombreux petits graphes est disponible
  2. Dans les applications telles que le Web et les réseaux sociaux, un seul grand graphe est disponible.

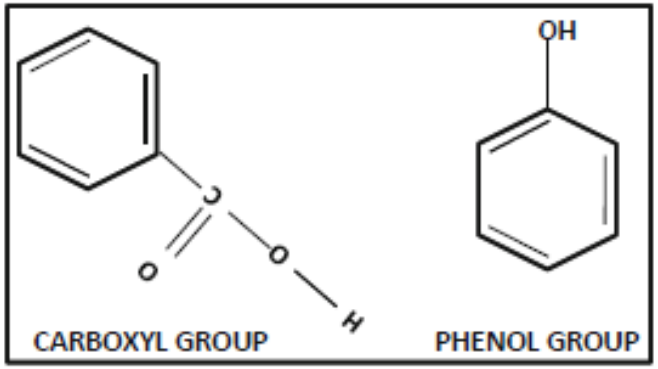
- Dans cette séance, nous nous intéresserons au premier type d'application.

Exemples d'applications

Graphe non orienté non simple
non topologique
étiquetté: chacun des sommets correspond à un carbonne



DATABASE OF PHENOLIC ACIDS

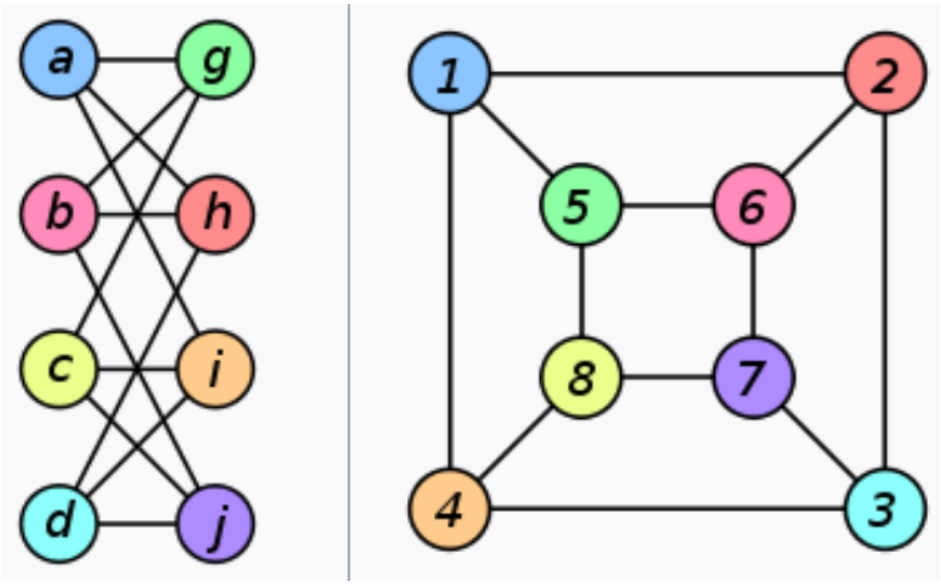


FREQUENT SUBSTRUCTURES OF PHENOLIC ACIDS

Aggarwal, 2015

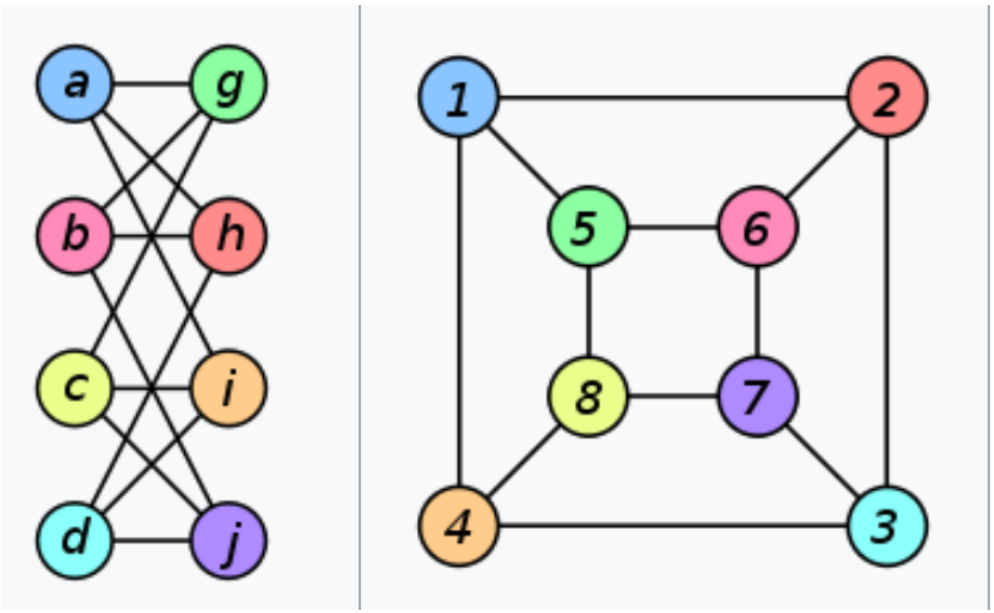
Fouille de graphes

- Comment-on mesure la distance ou la similarité entre deux graphes ?



Graphes équivalents

Exemple



$f(a) = 1$

$f(b) = 6$

$f(c) = 8$

$f(d) = 3$

$f(g) = 5$

$f(h) = 2$

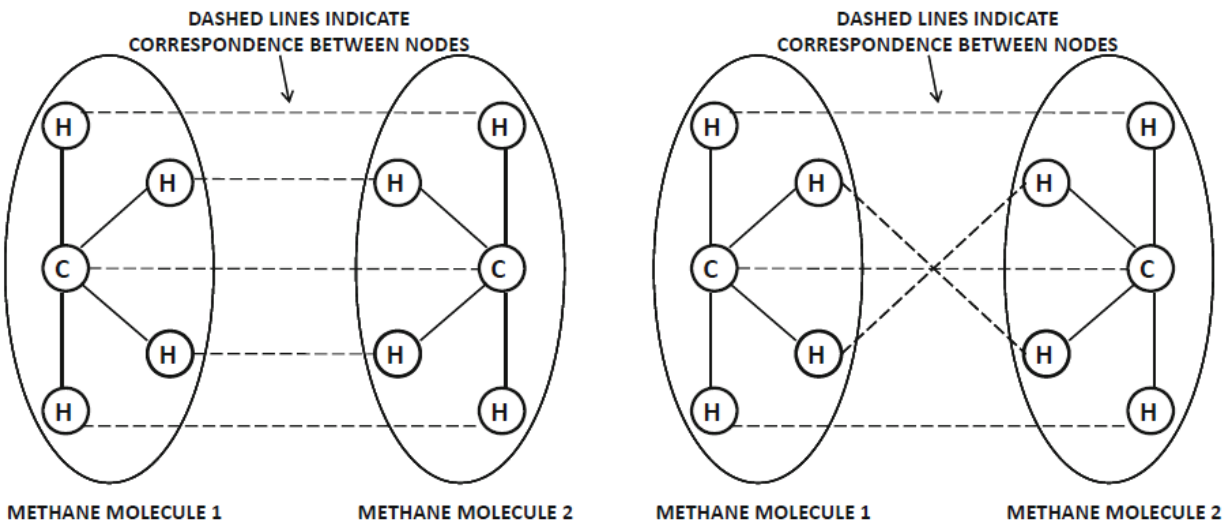
$f(i) = 4$

$f(j) = 7$

Ces deux graphes sont en réalité isomorphes.
une distance de 0

Isomorphisme de graphes

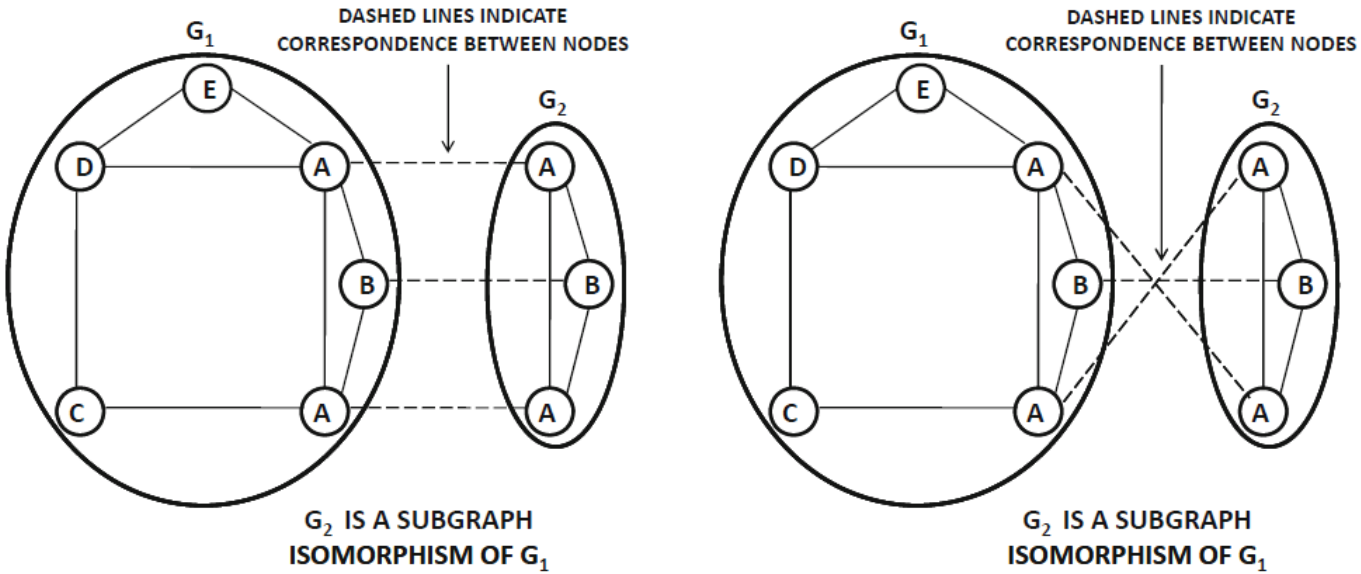
- Le problème de savoir si deux graphes sont isomorphes est NP-complet.
- Le problème devient encore plus difficile lorsque les étiquettes de sommets se répètent.



Aggarwal, 2015

Isomorphisme de sous-graphe

NOTION TRÈS IMPORTANTE



Aggarwal, 2015

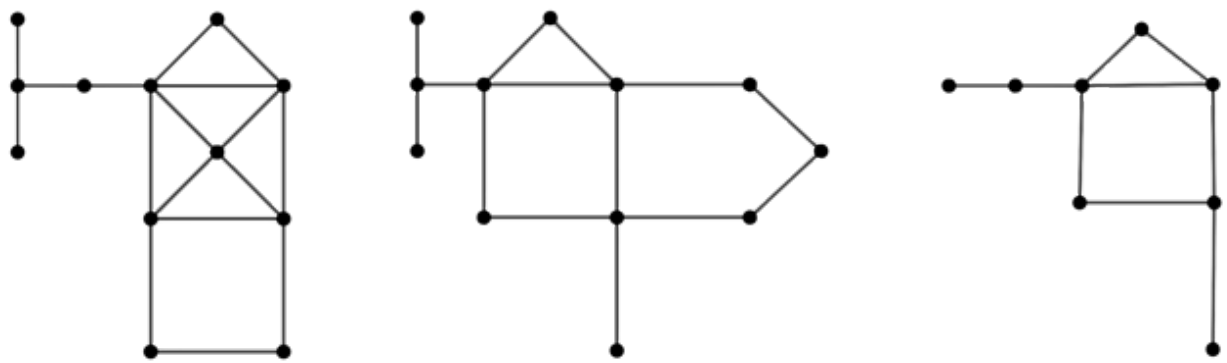
Le problème de savoir si un graphe est sous-graphe isomorphe à un autre est aussi NP-complet.

Maximum commun sous-graphe (MCS)

Enregist Minute 40

Le nb de sommets du plus grande nb de graphes à G1 et G2

- Le MCS entre une paire de graphes $G_1 = (N_1, E_1)$ et $G_2 = (N_2, E_2)$ est un graphe $G_0 = (N_0, E_0)$ qui est sous-graphe isomorphe à G_1 et G_2, et pour lequel la taille de l'ensemble de noeuds N_0 est aussi grande que possible.



Un sous-graphe isomorphe d'un graphe est un sous-graphe qui a la même structure de données que le graphe original. Cela signifie que le sous-graphe a les mêmes sommets et les mêmes arêtes que le graphe original, mais peut être disposé de manière différente.

Mesures de similarité

- En utilisant la MCS :

taille 1er graphe

$$d(G_1, G_2) = |G_1| + |G_2| - 2|MCS(G_1, G_2)|$$

ceci est égal au nombre de nœuds non-correspondants entre les deux graphes
* abus de notation : $|G|$ égal à $|N|$

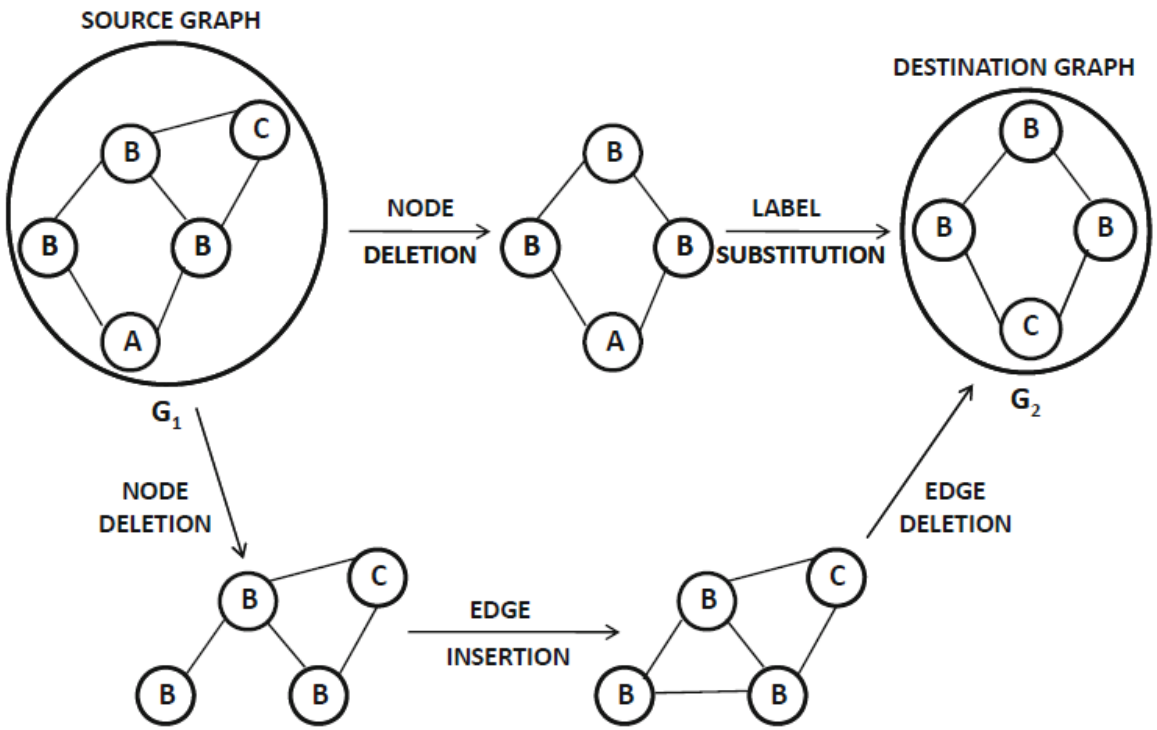
- Pas idéal ! Normalisation requise :

$$d_{norm}(G_1, G_2) = 1 - \frac{|MCS(G_1, G_2)|}{|G_1| + |G_2| - |MCS(G_1, G_2)|}$$

$d_{norm} \in [0, 1]$

Mesures de similarité

- La distance d'édition $Edit(G_1, G_2)$ est égal au coût minimum des opérations d'édition à appliquer au graphe G_1 pour le transformer en $G2$.



Aggarwal, 2015

Mesures de similarité

- Les mesures de distances présentées ne peuvent être utilisées que pour des petits graphes parce que :
  - MCS est NP-difficile
  - Le calcul de $Edit(G_1, G_2)$ est aussi NP-difficile
  - Il nous faut d'autres options plus performantes

Transformations basées sur un noyau

- Les méthodes basées sur des noyaux peuvent être utilisées pour un calcul de similarité plus rapide.
- De plus, ces méthodes de calcul de similarité peuvent être utilisées directement avec les SVMs.
- La similarité de noyau $\mathcal{K}(G_i, G_j)$ entre une paire de graphes G_i et G_j est le produit scalaire des deux graphes après leurs transformations hypothétiques dans un nouvel espace défini par la fonction $\phi(\cdot)$:

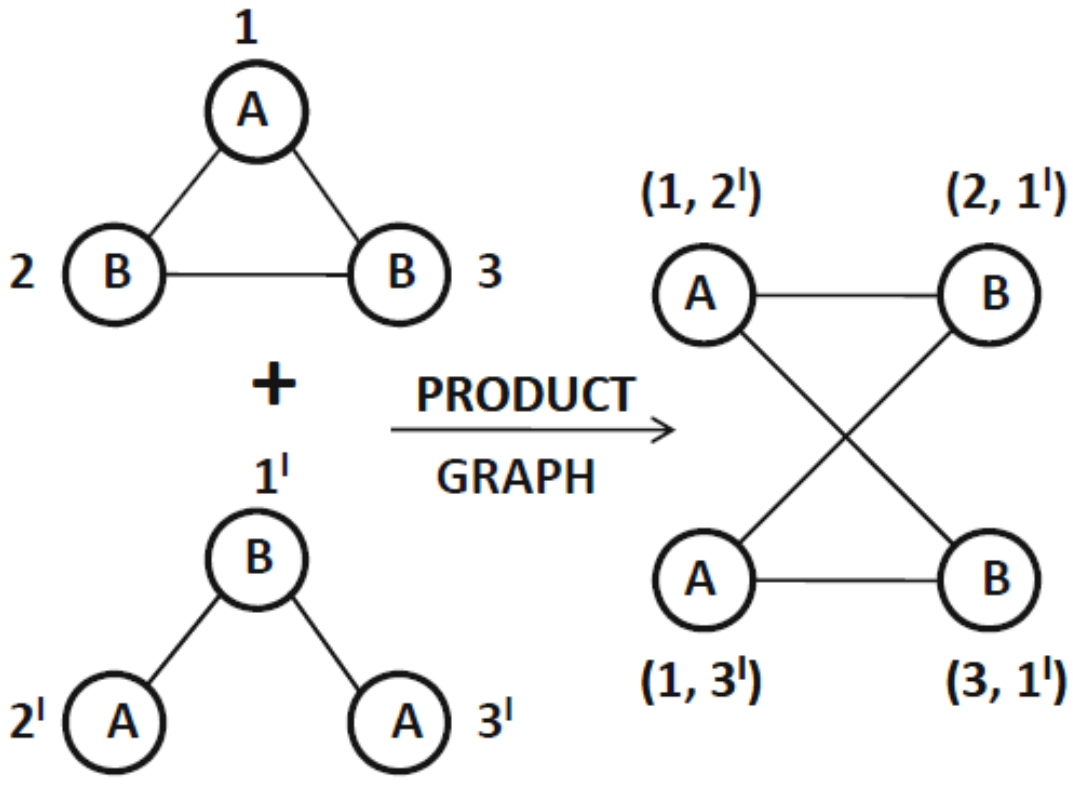
$$\mathcal{K}(G_i, G_j) = \phi(G_i) \cdot \phi(G_j)$$

- En pratique, la fonction $\phi(\cdot)$ n'est pas définie directement.
- Il y a plusieurs façons de définir une similarité de noyau pour les graphes.

Marches aléatoires

- Principe :
  - Compter les marches communes dans G_1 et G_2
  - Les marches sont des séquences de sommets avec répétition
- Calcul :
  - Construction du graphe produit de G_1 et G_2

Marches aléatoires



Aggarwal, 2015

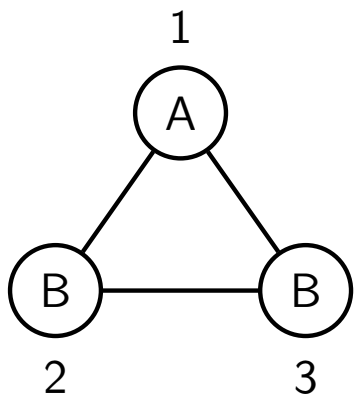
Chaque marche dans le graphe produit correspond à une séquence appariée en termes de sommets dans G_1 et G_2.

Graphe produit

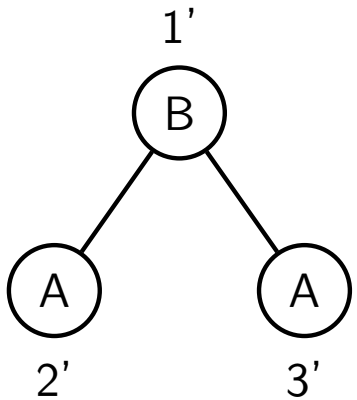
Les sommets du graphe produit de G_1 et G_2 correspondent aux paires de sommets de G_1 et de G_2 qui ont la même étiquette :

$$V_X = \{(v_1, v_2) : v_1 \in G_1 \wedge v_2 \in G_2 \wedge \text{label}(v_1) = \text{label}(v_2)\}$$

les étiquettes sont les mm



G_1



G_2

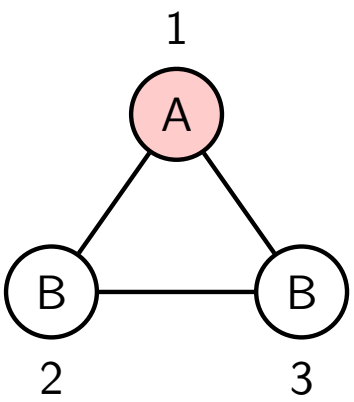
Graphe produit G_X

essayer tt les paires de sommets qu'on peut matcher ensemble

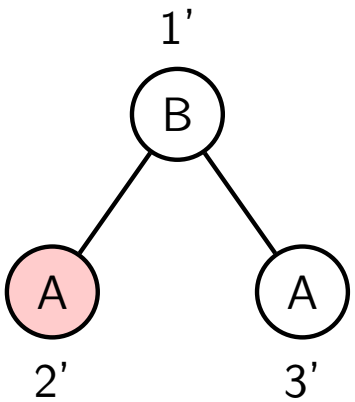
Graphe produit

Les sommets du graphe produit de G_1 et G_2 correspondent aux paires de sommets de G_1 et de G_2 qui ont la même étiquette :

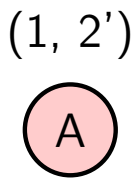
$$V_X = \{(v_1, v_2) : v_1 \in G_1 \wedge v_2 \in G_2 \wedge \text{label}(v_1) = \text{label}(v_2)\}$$



G_1



G_2

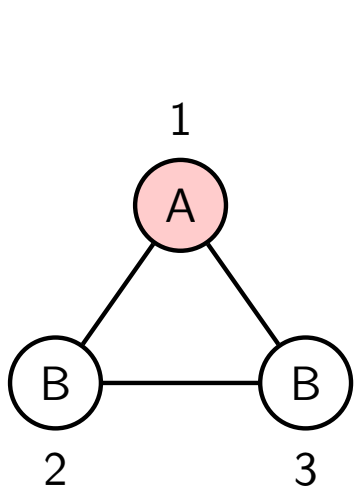


Graphe produit G_X

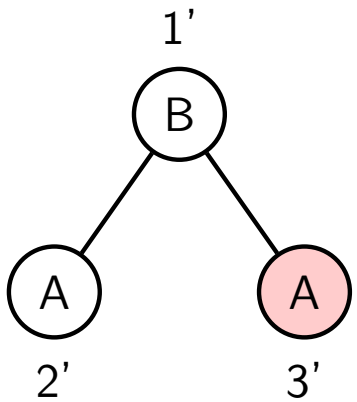
Graphe produit

Les sommets du graphe produit de G_1 et G_2 correspondent aux paires de sommets de G_1 et de G_2 qui ont la même étiquette :

$$V_X = \{(v_1, v_2) : v_1 \in G_1 \wedge v_2 \in G_2 \wedge \text{label}(v_1) = \text{label}(v_2)\}$$



G_1



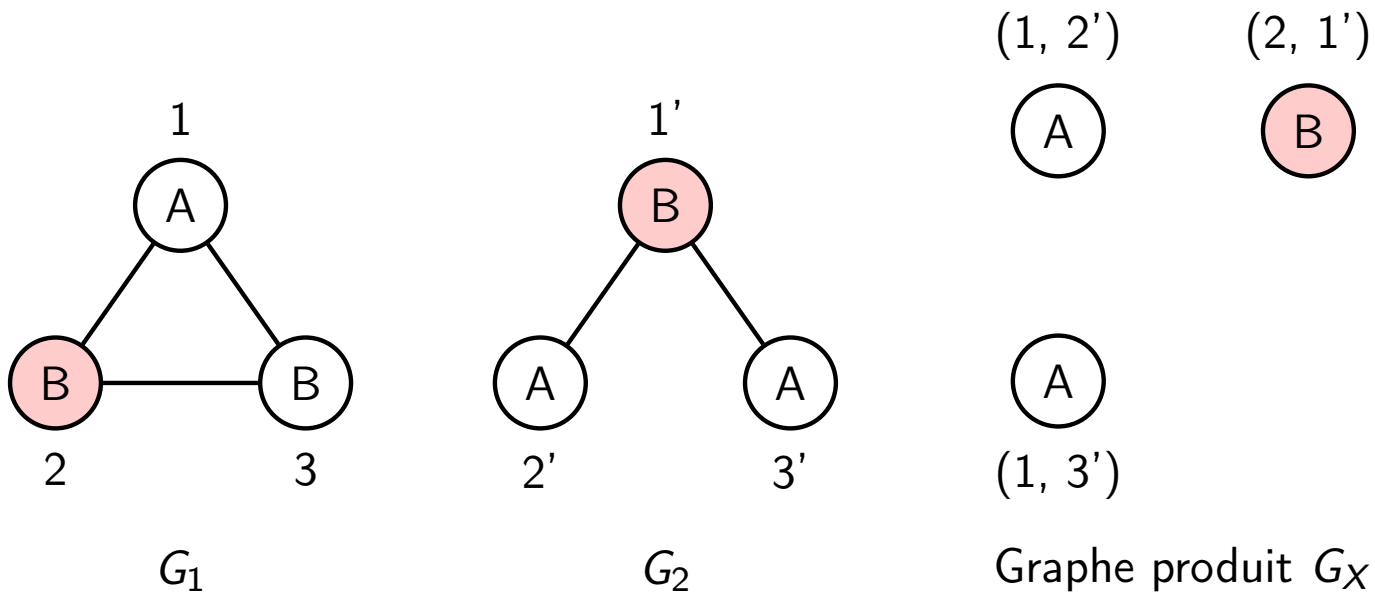
G_2

Graphe produit G_X

Graphe produit

Les sommets du graphe produit de G_1 et G_2 correspondent aux paires de sommets de G_1 et de G_2 qui ont la même étiquette :

$$V_X = \{(v_1, v_2) : v_1 \in G_1 \wedge v_2 \in G_2 \wedge \text{label}(v_1) = \text{label}(v_2)\}$$

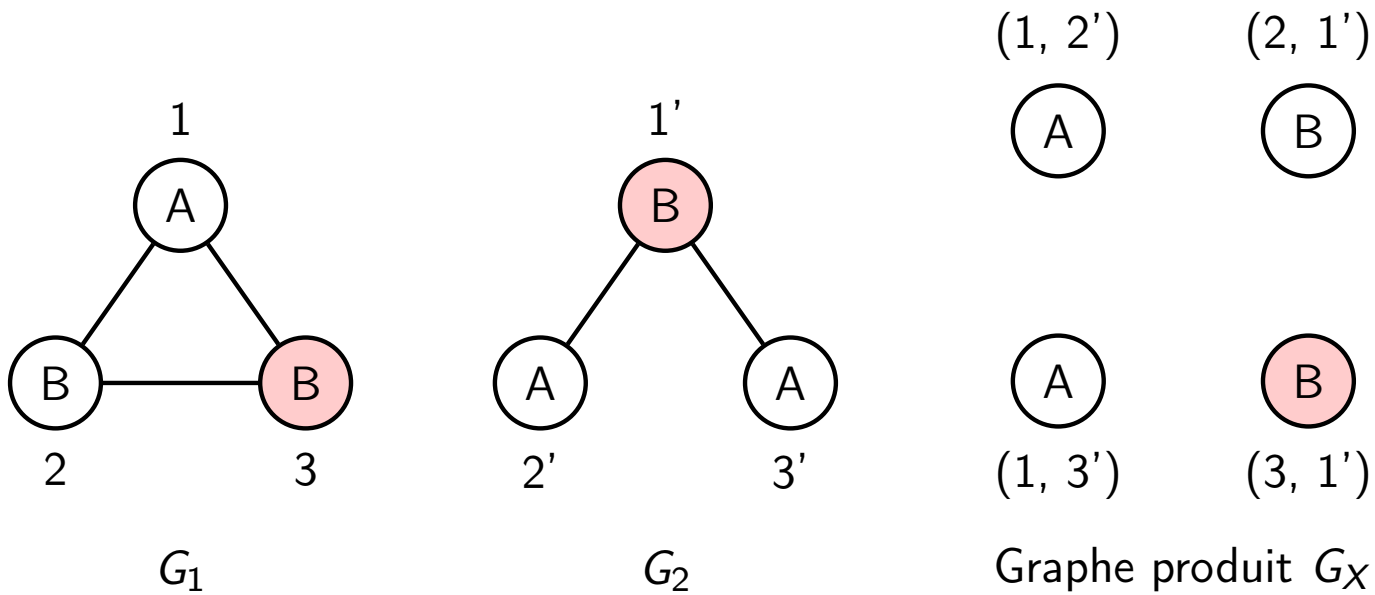


G_1 G_2 Graphe produit G_X

Graphe produit

Les sommets du graphe produit de G_1 et G_2 correspondent aux paires de sommets de G_1 et de G_2 qui ont la même étiquette :

$$V_X = \{(v_1, v_2) : v_1 \in G_1 \wedge v_2 \in G_2 \wedge \text{label}(v_1) = \text{label}(v_2)\}$$

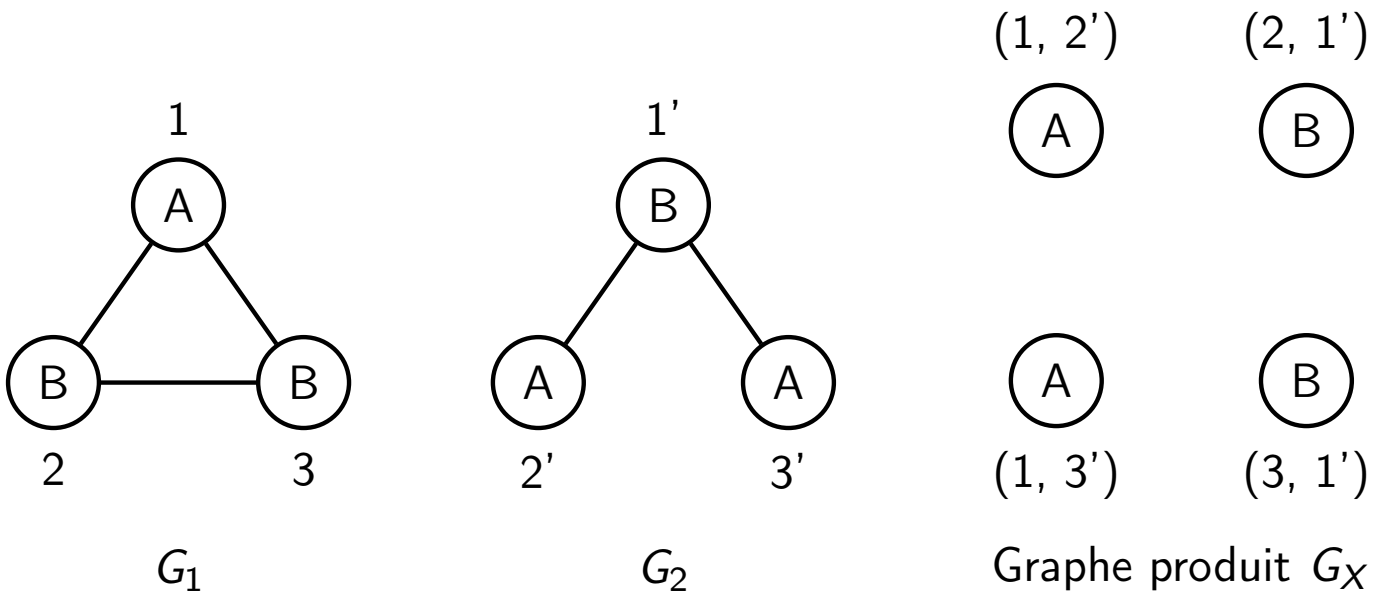


G_1 G_2 Graphe produit G_X

Graphe produit

Les arêtes du graphe produit correspondent aux arêtes communes à G_1 et à G_2 : arretes à la fois dans le graphe 1 et dans le graphe 2

$$E_X = \{((u_1, u_2), (v_1, v_2)) : (u_1, v_1) \in G_1 \wedge (u_2, v_2) \in G_2\}$$

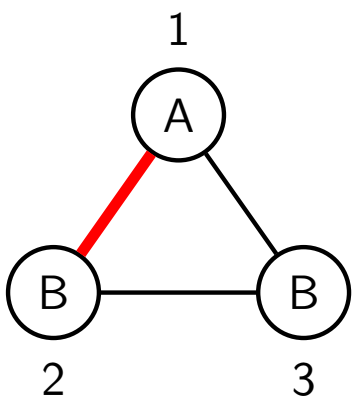


Graphe produit

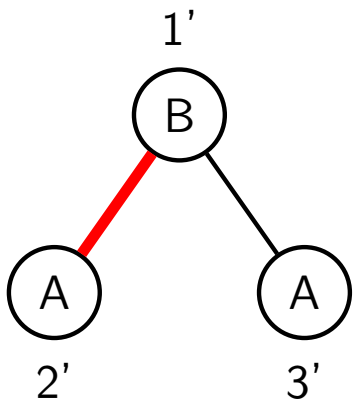
Les arêtes du graphe produit correspondent aux arêtes communes à G_1 et à G_2 :

$$E_X = \{((u_1, u_2), (v_1, v_2)) : (u_1, v_1) \in G_1 \wedge (u_2, v_2) \in G_2\}$$

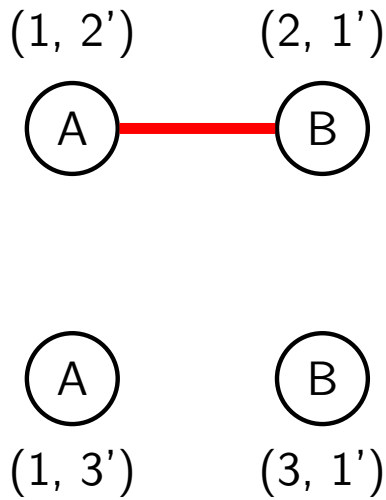
Est-ce qu'on peut aller de1 à 2 et de 2' à 1'? Oui, donc faire un trait



G_1



G_2

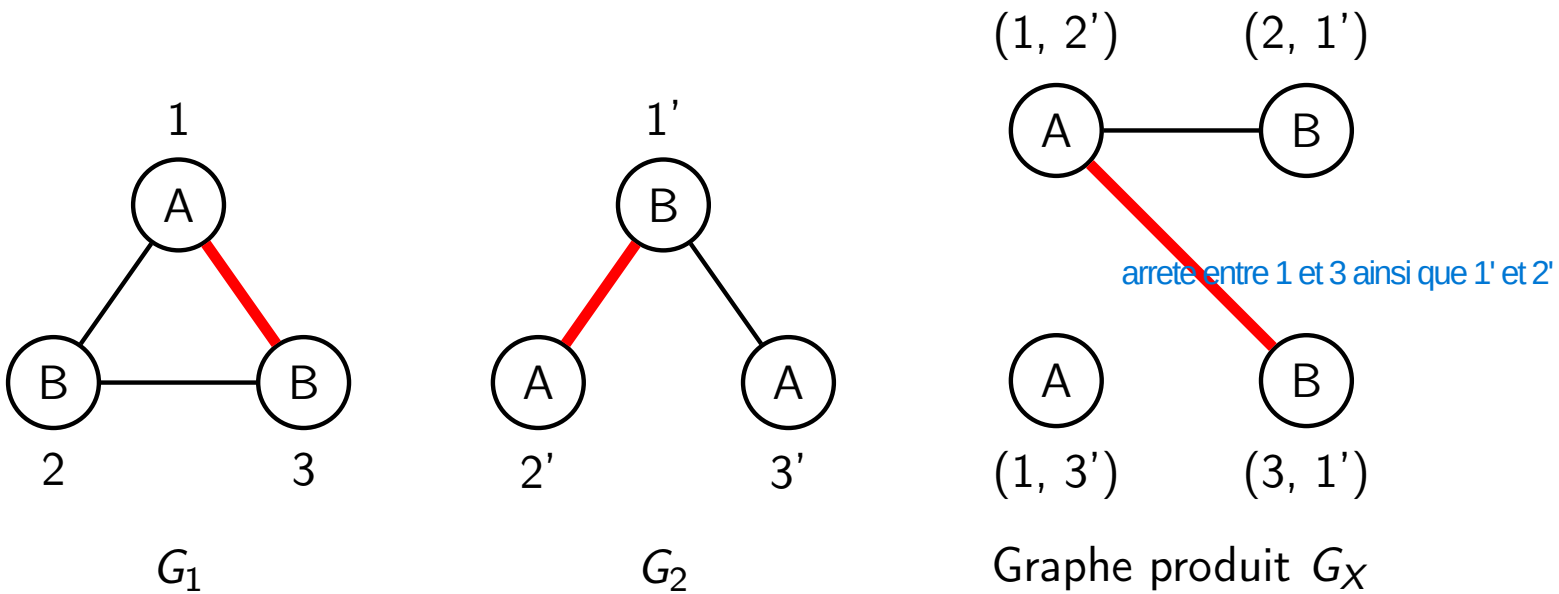


Graphe produit G_X

Graphe produit

Les arêtes du graphe produit correspondent aux arêtes communes à G_1 et à G_2 :

$$E_X = \{((u_1, u_2), (v_1, v_2)) : (u_1, v_1) \in G_1 \wedge (u_2, v_2) \in G_2\}$$

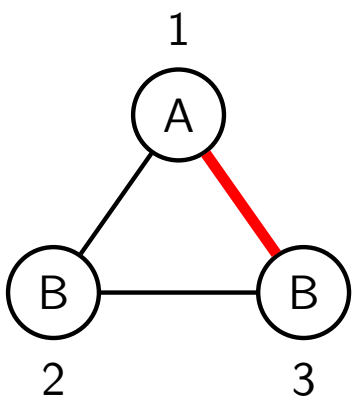


G_1 G_2 Graphe produit G_X

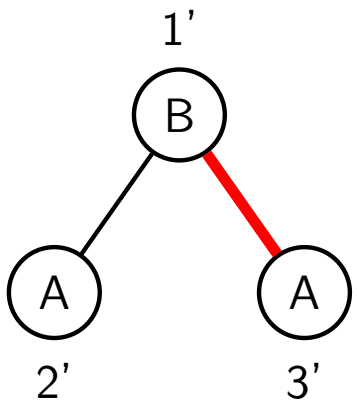
Graphe produit

Les arêtes du graphe produit correspondent aux arêtes communes à G_1 et à G_2 :

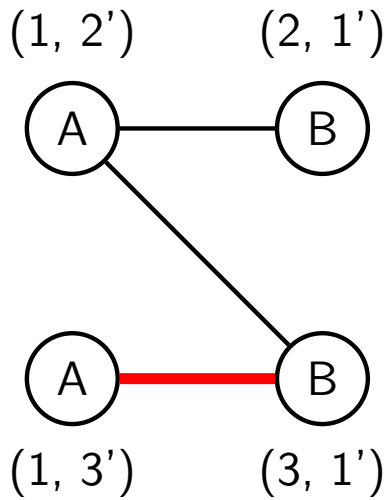
$$E_X = \{\big((u_1, u_2), (v_1, v_2)\big) : (u_1, v_1) \in G_1 \wedge (u_2, v_2) \in G_2\}$$



G_1



G_2

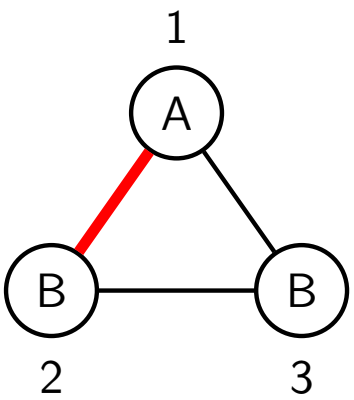


Graphe produit G_X

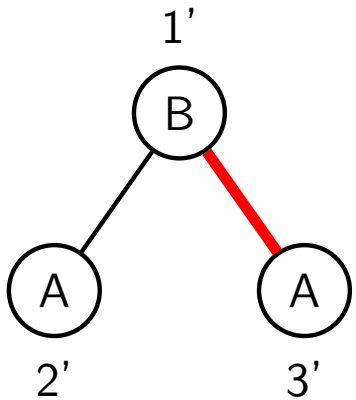
Graphe produit

Les arêtes du graphe produit correspondent aux arêtes communes à G_1 et à G_2 :

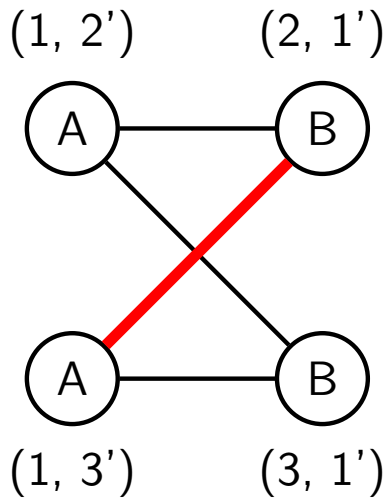
$$E_X = \{\big((u_1, u_2), (v_1, v_2)\big) : (u_1, v_1) \in G_1 \wedge (u_2, v_2) \in G_2\}$$



G_1



G_2



Graphe produit G_X

Marches aléatoires

- Calcul :
  - Construction du graphe produit de G_1 et G_2.
  - Le nombre de marches de longueur k peut être calculé en regardant la k-ième puissance de la matrice d'adjacence A du graphe produit.
  - Ainsi :

$$\mathcal{K}(G_1, G_2) = \sum_{ij} \sum_{k=1}^{\infty} \lambda^k [A^k]_{ij}$$

où $\lambda \in (0, 1)$ est choisi de façon à garantir la convergence de la série.

Chemins plus courts

- Les marches aléatoires permettent de répéter les sommets des séquences.
- Une marche peut visiter le même cycle de sommets plusieurs fois.
- Le noyau basé sur les marches aléatoires mesure la similarité en termes de marches communes.
- Par conséquent, une petite similarité structurelle peut provoquer une énorme valeur de noyau.
- Solution : noyau basé sur les plus courts chemins.

Chemins plus court

- La fonction $k(i_1, j_1, i_2, j_2)$ est définie par paires de sommets avec $i_1, j_1 \in G_1$ et $i_2, j_2 \in G_2$.
- $k(i_1, j_1, i_2, j_2) = 1$ si le plus court chemin entre i_1 et j_1 dans G_1 est de la même taille que le plus court chemin entre i_2 et j_2 dans G_2.
- Ainsi, la fonction noyau est définie comme :

$$\mathcal{K}(G_1, G_2) = \sum_{i_1, j_1, i_2, j_2} k(i_1, j_1, i_2, j_2)$$

Chemins plus court

- La fonction $k(i_1, j_1, i_2, j_2)$ est définie par paires de sommets avec $i_1, j_1 \in G_1$ et $i_2, j_2 \in G_2$.
- $k(i_1, j_1, i_2, j_2) = 1$ si le plus court chemin entre i_1 et j_1 dans G_1 est de la même taille que le plus court chemin entre i_2 et j_2 dans G_2.
- Ainsi, la fonction noyau est définie comme :

$$\mathcal{K}(G_1, G_2) = \sum_{i_1, j_1, i_2, j_2} k(i_1, j_1, i_2, j_2)$$

- ça coûte quand même cher...

Clustering de graphes

- Partitionne la base de données de n graphes $G_1, \ldots, G_n$ en k *clusters*.
- *out-of-the-box* : méthodes basées sur des dissimilarités.

Clustering de graphes

- Une deuxième méthodologie utilisée est celle des méthodes spectrales.
- Les graphes de données $G_1, \ldots, G_n$ sont utilisés pour construire un seul graphe global $\overline{G}$.
- Chaque graphe G_i correspond à un sommet dans $\overline{G}$.
- Chaque sommet de $\overline{G}$ est lié à ces plus proches voisins selon les distances calculées.
- Donc, le problème de regrouper $G_1, \ldots, G_n$ devient le problème de regrouper les sommets d'un seul graphe $\overline{G}$.
- Possible algorithme : transformation spectrale sur $\overline{G}$ + *k-means*.

Classification de graphes

- On suppose qu'un ensemble de n graphes $G_1, \ldots, G_n$ est disponible, mais seul un sous-ensemble de ces graphes est étiqueté (avec des étiquettes $1, \ldots, k$).

- *out-of-the-box* : KNN, chaque nouveau graphe non-étiqueté prend l'étiquette de la classe majoritaire parmi ses k-plus proches voisins .

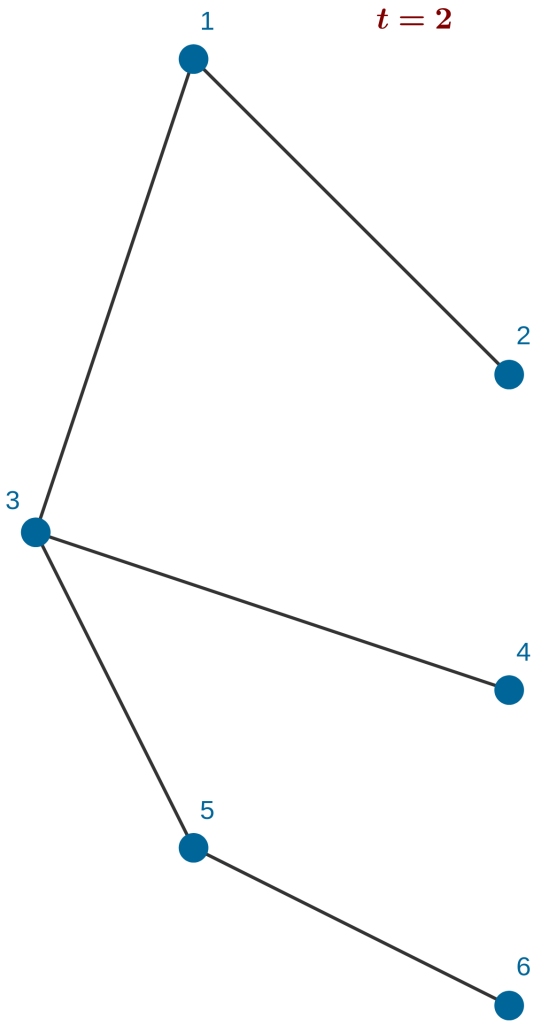
Descripteurs topologiques

- Les descripteurs topologiques convertissent les graphes en données multidimensionnelles où chaque attribut mesure une caractéristique structurelle importante.
- Une fois la conversion effectuée, des algorithmes d'exploration de données multidimensionnels peuvent être utilisés sur la représentation transformée.
- L'inconvénient de cette approche est qu'elle implique une perte information.

Descripteurs topologiques

- Quelques exemples de descripteurs topologiques sont :

  Morgan Index$_t$ égal à un vecteur de taille $|G|$ où chaque composante i est égale au nombre de sommets accessibles depuis le sommet i à une distance d'au plus t.



Morgan Index

4	2	5	3	4	2
1	2	3	4	5	6

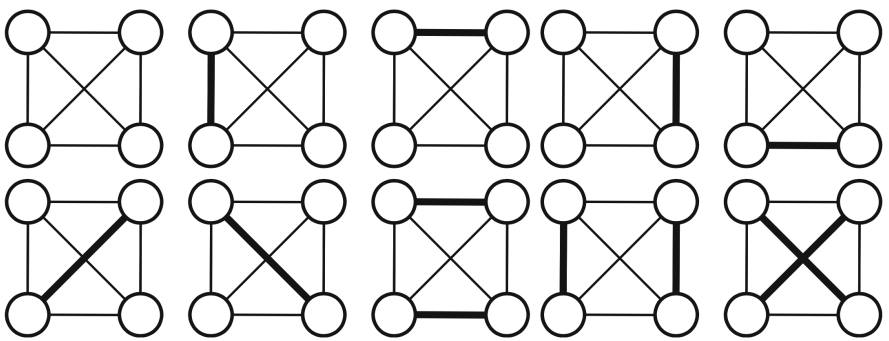
à partir de 1: avec une distance de deux on peut assister au sommet 3 et 4

Descripteurs topologiques

- Quelques exemples de descripteurs topologiques sont :

  Wiener Index égal à la somme des distances les plus courtes entre toutes les paires de sommets du graphe.

  Hosoya index égal au nombre de *matchings* valides dans le graphe.



Aggarwal, 2015