

utilisation Les graphes sont des objets très simple qui n'encode qu'une quantité limité d'information : un graphe représente les voisinages d'un système.

Cela est utile dans les contextes pour lesquels les notions de voisinage sont une partie intégrale du problème : les problèmes appliqués sur les réseau :

- structure : connectivité, centralité, distances
- dynamiques : communication, consensus, propagation, contagion
- réseaux informatiques : prédiction de l'effet d'un protocole, stabilité
- ...

mais aussi comme un moyen de découper un problème trop complexe ou trop grand en un grand nombre de petite composante, dont les complexités interactions sont bornées et solubles. Le cas le plus simple est la récurrence : on prouve une implication simple sur chaque valeur de paramètre, en renvoyant la résolution en aval jusqu'à arriver à un point suffisamment simple pour être résolu directement, ce qui peut être vu comme une marche infinie remontant un graphe implicite jusqu'à arriver en terrain connu.

On peut construire sur cet objet pour obtenir des modèle plus riche et utiliser des variantes de cet objet dans des contextes variés :

- Un graphe dont les arêtes sont étiquetées par des probabilités, et dont on considère principalement les marches aléatoire devient une Chaîne de Markov un outil puissant de modélisation de système, caractérisant entre autre les état et temps de convergence
- Un Automate est un graphe dont les arêtes sont étiqueté par un alphabet, reflétant parmi ses sommets le comportement de mots d'un langage dictant ses transitions
- Souvent en optimisation, on étiquète les sommets ou les arêtes d'un graphes par des poids, en cherchant à extraire un sous graphe vérifiant une propriété et minimisant le poids total : le voyageur de commerce est un tel problème, avec des poids sur les arêtes et la propriété d'être un cycle. d'autres exemples incluent une couverture par un arbre : un sous graphe connexe, la couverture de toutes les arêtes par des sommets, qui est une forme de coloriage, la recherche d'association via les couplages ...
- Si on associe aux arêtes des bornes, puis que l'on ajoute une loi de conservation, avant de maximiser un transfert entre deux sommets, on obtient les problème de flot
- dans les réseaux, l'identification de structure, d'éléments centraux, de sommet très connectés, de sous groupes isolés, d'exposition de page web... peut être modélisé par des propriétés de graphes : principalement les notions de centralité, α -connectivité.
- on peut parfois utiliser plusieurs niveaux de graphes : la théorie des file d'attente admet une structure de graphe entre les serveurs de part l'aspect de modélisation de réseau, mais aussi une structure de chaîne de Markov entre les différentes configuration. Certain modèle épidémiologiques modélise les transfert d'information et leur impact sur les comportement par un deuxième graphe, partiellement indépendant de celui des contact physique
- En théorie des jeux, les graphes apparaissent sur plusieurs niveaux : de

nombreux jeux utilisent une structure de graphe entre les joueurs, ou peuvent être simplifiés en forçant une telle structure, mais le principale apport vient du graphe des configuration, qui est souvent une chaîne de Markov et permet de traiter l'évolution et la dynamique des différentes situations

Les graphes sont un objet abstrait et simple, qui doivent souvent être adaptés et choisi pour être appliqué à un problème donné. Une fois cette étape faite, ils permettent de réduire et résoudre les instances en bornant le nombre de facteurs affectant chaque sous cas.

définitions

Définition 1. *Grappe* Un graphe est un ensemble de sommet, parfois appelé noeuds, et généralement noté V et un ensemble de couple de sommets, appelés arête ou parfois arcs, et noté $E \subset V \times V$.

Un graphe est dit non orienté si l'ensemble des arêtes est symétrique : $\forall (i, j) \in E, (j, i) \in E$, et orienté sinon.

Dans le case non orienté, une arête correspond aux deux directions, et n'est comptée qu'une seule fois.

Définition 2. *Sous graphe* Un sous graphe $G' = V', E'$ d'un graphe $G = V, E$ est un sous ensemble des sommets de G $V' \subset V$ et un sous ensemble des arêtes de G $E' \subset E$ compatible avec ses sommets $E' \subset V' \times V'$

Comme le graphe est une pair d'ensemble, on peut effectuer des opérations d'ensemble : l'intersection de deux graphes est un sous graphe commun aux deux graphes, l'union de deux graphes est un graphe, l'union des arêtes de deux graphes sur les même sommets est un graphe. On peut également parler de partitions d'un graphe, soit sur seulement les arêtes, soit sur les sommets, en associant les arêtes incluses dans une partie à celle ci, et mettant de côté celles entre deux parties. On peut associer à un graphe orienté son équivalent non orienté qui est l'union du graphe et du graphe obtenu en retournant les arêtes.

Définition 3. *Degré* Le degré sortant d_i^+ d'un sommet i est le nombre d'arêtes dont le premier élément est ce sommet.

Le degré entrant d_i^- d'un sommet i est le nombre d'arêtes dont le second élément est le sommet.

Le degré total d_i est le nombre d'arêtes dont l'un des éléments est le sommet.

Dans le contexte d'un graphe non orienté, ces trois valeurs sont les mêmes et simplement appelées le degré du sommet.

Dans le cadre orienté, en l'absence de boucle (arête d'un sommet vers lui même), le degré total est la somme des degrés entrants et sortants

Propriété 1. Dans un graphe orienté, $\sum_i d_i^+ = \sum_i d_i^- = |E|$
 Dans un graphe sans boucles $\sum d_i = 2|E|$

Démonstration. Une manière simple de prouver ce résultat est de le montrer sur un graphe sans arêtes (toutes ces valeurs sont 0) et de regarder la contribution de l'ajout d'une arête à un graphe quelconque (une arête i, j augmente $|E|$, d_i^+ , d_j^- , d_i et d_j de 1 exactement) \square

Le degré d'un sommet dans une union disjointe de graphes sur les mêmes sommets est la somme des degrés dans les différents graphes.

Définition 4. *Chemin* Un chemin dans un graphe est une suite d'arête consécutives :

$$i \rightarrow^* j = ((ix_1)(x_1x_2)(x_2x_3)(x_3x_4) \cdots (x_mx_j)), \forall k, (x_k, x_{k+1}) \in E$$

Un chemin est élémentaire s'il ne passe pas deux fois par la même arête. (ie $\forall k \neq \ell, x_k \neq x_\ell \vee x_{k+1} \neq x_{\ell+1}$)

Un chemin est simple s'il ne passe pas deux fois par le même sommet (ie $\forall k \neq \ell, x_k \neq x_\ell$)

Remarquez que dans un graphe orienté, un chemin simple est toujours élémentaire, tandis qu'un graphe non orienté admet les cycle de longueur 2 comme contre exemples qui empruntent une arête dans les deux sens, mais la propriété est vrai pour tout autre chemin.

Propriété 2. *S'il existe un chemin entre un sommet i et un sommet j , alors il existe un chemin simple entre i et j .*

Démonstration. Soit un graphe, et deux sommets du graphe tels qu'il existe un chemin depuis le premier vers le second. On considère la suite des sommets visités par le chemin.

Dans cette suite, on considère l'ensemble des sommets apparaissant plus d'une fois, que l'on ordonne par l'indice de première apparition. Nous allons effectuer une récurrence sur la taille de cet ensemble.

Si l'ensemble est vide, le chemin est simple. Si l'ensemble n'est pas vide, nous allons exhiber un nouveau chemin entre i et j dont l'ensemble des sommets visités plus d'une fois est un sous ensemble stricte de l'ensemble actuel. Pour cela, on considère le sépare la suite aux indices de la première et de la dernière instance du premier sommet apparaissant plusieurs fois : $i \rightarrow^* j = ix_1 \cdots x_k Y x_{k+2} \cdots x_\ell Y x_{\ell+2} \cdots x_m j$, et on retire tous les sommets entre ces deux instance : $ix_1 \cdots x_k Y x_{\ell+2} \cdots x_m j$.

Toutes les arêtes empruntées dans ce chemin étaient présentes dans le chemin précédent, donc dans le graphe : ce chemin est valide. L'ensemble avec multiplicité des sommets emprunté par ce nouveau chemin est un sous ensemble de celui du chemin précédent, donc les sommets multiples du nouveau chemins étaient des sommets multiples du précédent. De plus, Y était un sommet traversé plus d'une fois et n'est traversé qu'une seule fois : l'opération a réduit l'ensemble et nous a ramené à l'hypothèse de récurrence sur un ensemble plus petit.

Par conséquent (en répétant l'opération un nombre fini de fois) on peut construire un chemin simple de i vers j \square

Pour illustrer la parenthèse sur l'équivalence entre récurrence et absurde, la même preuve, par l'absurde :

Démonstration. Soit i et j deux sommets d'un graphe tels qu'il existe un chemin de i vers j dans le graphe. On considère le plus court chemin de i vers j . Si le chemin n'est pas simple, on considère un sommet traversé plus d'une fois, et on découpe le chemin sur la première et la dernière traversée : on a un chemin de i vers y , un chemin non vide de y vers y et un chemin de y vers j . En concaténant le premier et le dernier, on obtient un chemin de i vers j strictement plus petit que celui supposé le plus court chemin, contredisant l'hypothèse de l'existence d'un sommet traversé plus d'une fois. \square

Définition 5. *Connexité* Un graphe est connexe si pour toute paire de sommets, il existe un chemin dans le graphe sans orientation.

Un graphe est fortement connexe si pour toute paire de sommets (i,j) , il existe un chemin de i vers j dans le graphe.

Définition 6. *Cycle* Un cycle est un chemin élémentaire non vide d'un sommet vers lui même

Une boucle est un cycle de taille 1, mais un chemin vide n'est pas un cycle. Une arête à double sens est un cycle dans un graphe orienté mais pas dans un graphe non orienté.

Définition 7. *Graphe acyclique* Un graphe est acyclique s'il ne contient aucun cycle

On peut remarquer que ces propriétés sont monotones vis à vis de l'inclusion sur l'ensemble des arêtes :

- Un sous graphe d'un graphe acyclique est acyclique
- Un graphe qui admet un sous graphe connexe sur les même sommets est connexe

Arbres

Définition 8. *Arbres* Un arbre enraciné en un sommet r est l'union disjointe du sommet r , d'une liste (possiblement vide) de graphes enracinés en $r_1 \cdots r_m$, et des arêtes de chacun des racines r_i des sous arbres vers r .

Les sommets de l'arbre sans descendant sont appelés feuilles.

Propriété 3. *Un arbre est toujours connexe et acyclique.*

Un graphe non orienté connexe acyclique est un arbre

Démonstration. Un arbre est connexe : tout sommet de l'arbre admet un chemin vers la racine, donc pour toute pair de sommet, il existe un chemin dans le graphe non orienté du premier sommet vers la racine puis de la racine vers le second.

Un arbre est acyclique : chaque arête ajoutée lie une racine d'un sous arbre à un unique parent, le degré de chaque sommet hors du sous arbre dont il est la racine (ie le degré sortant dans le cas orienté) est soit 0 pour la racine globale,

soit 1. Si l'on considère un chemin élémentaire dans le graphe, le degré de chaque sommet intermédiaire restreint au chemin est 2, donc au moins un voisin dans le sous arbre stricte, et l'autre voisin de ce voisin dans le sous arbre suivant : un chemin élémentaire s'éloigne strictement de la racine à chaque pas indéfiniment, ce qui ne permet pas de construire un cycle.

Soit un graphe non orienté connexe et acyclique, et un de ses sommets sélectionné comme candidat racine. On considère un autre sommet du graphe. Comme le graphe est connexe, il existe un chemin simple de cet autre sommet vers le candidat racine. Comme le graphe est acyclique, ce chemin est unique. On groupe l'ensemble des sommets du graphe selon le dernier élément du chemin vers la racine. Par existence et unicité de ce chemin simple, ce groupement est une partition des sommets. De plus, par construction, une arête entre deux sommet de deux parties différente créé un cycle, donc toutes les arêtes non adjacente à la racine sont interne à l'une des partie, et cette partition est aussi une partition des arête. Finalement, chaque partie est connexe, acyclique, et a comme candidat racine le voisin de la racine précédente, ce qui permet de répéter l'opération : un graphe non orienté connexe acyclique est un arbre. \square

Un parcours est une exploration d'un graphe passant par tous les sommets co-accessibles depuis le point de départ en suivant les arêtes. Implicitement, un parcours construit un arbre comme sous graphe de la composante connexe.

Un parcours en largeur visite les sommets par distance croissante à la racine. Un tel parcours permet d'évaluer des informations centrées sur la racine choisi : ensemble des distances, distance maximale, centralité dans le graphe, niveau de voisinage, exploration centrée.

Un parcours en profondeur évite les sauts, ie les visite à des sommets non adjacent au sommet précédent. en conséquence, ce type de parcours est beaucoup plus local, bornant la quantité de mémoire globale requise et permettant une plus grande parallélisation. L'heuristique "main droite" dans un labyrinthe est une forme de parcours en profondeur, choisissant toujours le premier descendant et n'annotant pas les bifurcations déjà visitées.

Il existe de nombreux autre types de parcours. Un équivalent du recuit simulé apparait parfois pour obtenir des optimum globaux dans des problèmes définis localement, en associant aux arêtes des probabilités de transition exponentielle sur une mesure d'énergie, dont la base de l'exponentielle augmente avec le temps