Présentation projet de recherche

Théodore CHAPUIS-CHKAIBAN, Ghita CHEMIA, Paul LIEUTIER, Hugo SERVANT, Victor SANSAC

CentraleSupélec

15/06/2020

Page Rank

- **Classer les noeuds** d'un graphe par ordre d'importance
- Parcourir le graphe avec une **marche aléatoire** (chaîne de Markov) : selon le nombre de visites d'un noeud, on lui **attribue un score**
- Cela revient à **diagonaliser la matrice d'occurrences** A du graphe (si les sommets i et j sont connectés alors A[i][j] = 1, sinon 0)

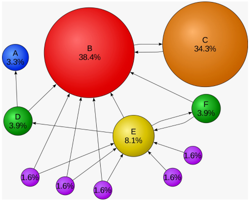


Figure 1 – Exemple de Graphe sur lequel fonctionne le Page Rank.

Page Rank

- Problème avec la diagonalisation : matrice de **dimension** $10^9 \times 10^9$: trop calculatoire
  $\rightarrow$ Projeter les valeurs propres sur un **espace de dimension inférieure** (SVD par ACP)
- Analogie avec la transformée de Fourier : que se passe-t-il si on applique la transformée de Fourier au PageRank ?
  $\rightarrow$ Objectif : **analogue quantique de la transformée de Fourier pour les graphes** ?

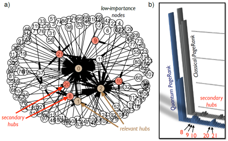


FIGURE 2 – Mise en évidence de la décomposition en fréquences pour un graphe

Applications actuelle du pageRank

Principale application directe à l'heure actuelle :

- La **classification** des pages internet.

Les vecteurs du PageRank sont aussi utilisés pour :

- Trouver des **clusters** dans un graphe
- Calcul de **partitions** de graphe
- Intelligence artificielle
- Quantifier / **classer les noeuds** selon leur importance

→ Dans tous les cas : **établir l'importance de certains noeuds** par rapport aux autres

Outline

Notion d'Importance

Importance des noeuds d'un graphe

- **L'importance d'un noeud** d'un graphe orienté G=(V,E) est définie par l'équation :

$$I(P_i) = \sum_{j \in B_i} \frac{I(P_j)}{deg(P_j)}$$

- Avec P_i la page i, $deg(P_i)$ le nombre de liens sortant de la page i et B_i l'ensemble des pages pointant sur la page i.
- Cette **équation est récursive** (I est définie en fonction des autres valeurs de l'importance)

Equation aux valeurs propres

- Soit alors I le vecteur de $\mathbb{R}^N$ (avec N le nombre de page du web) et H une matrice telle que $H_{i,j} = \frac{1}{deg(P_j)}\delta_{j \in B_i}$, il vient :

$$HI = I$$

- H est **typiquement de taille** $10^9 * 10^9$, il est impensable de diagonaliser directement H (en tout cas par des méthodes classiques)
- On travaille sur une **forme légèrement différente de H** (matrice primitive) qui puisse nous permettre d'appliquer le théorème de Perron Frobenius

Matrice de Google

- En posant : $G = \alpha H + \frac{1-\alpha}{N} F$ avec F une matrice qui ne comporte que des 1, on peut **appliquer le théorème de Perron-Frobenius** et obtenir le résultat par approximations successives.
- G est appelée "matrice de Google"

Quantum Page Rank

Google Page Rank

- Utilise la **Grover search**
- Cela revient à faire une **marche aléatoire quantique sur le graphe de Google**

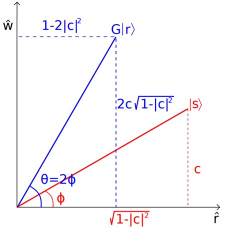


FIGURE 3 – La recherche de Grover est utilisée pour calculer le PageRank par réflexions successives.

Principe de la Grover Search

- On met le système dans un **état initial superposé** :

$$\psi_0 = \sum_{j=1}^{N}(j \otimes \sum_{k=1}^{N} \sqrt{G_{kj}}k)$$

- Que l'on fait **évoluer par réflexions successives** autour de l'espace engendré par les $\psi_j = j \otimes \sum_{k=1}^{N} \sqrt{G_{kj}}k$.
- L'opérateur d'évolution est défini de la manière suivante :

$$U = SR = S(2P - I)$$

$\rightarrow$ S l'opérateur de Swap ($\sum_{i,j} ijji$) et P la projection ($\sum_{i,j} \psi_j\psi_i$)

Calcul de complexité

Théorème

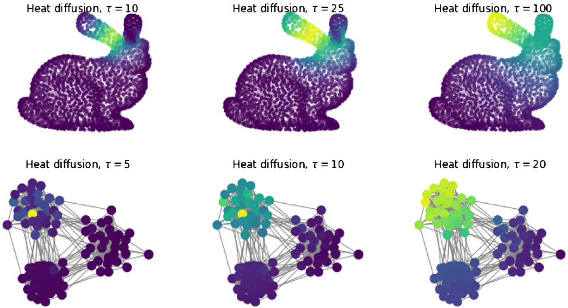
La **complexité moyenne de cette méthode est** $\sqrt{N}$ où N est la taille des données en entrée.

Origines de la Transformée de Fourier sur les Graphes

Théorie spectrale des graphes : très récent (2015 pour les premiers articles)

Comment **définir une transformée de Fourier sur des Graphes** ?

$\rightarrow$ structure irrégulière



Formalisation

- **Propagation de chaleur dans un graphe $\leftrightarrow$ marche aléatoire sur les sommets**
- $\frac{d\phi}{dt} = -k \underbrace{(D - A)}_{\doteq L} \phi$; équation de la chaleur
- D : matrice des degrés, $D \doteq diag(deg(0), \ldots, deg(n))$; A : matrice d'adjacence, ϕ : distribution de chaleur
- ∇ : gradient ; $\nabla f(i,j) \doteq \sqrt{A_{ij}}(f(j) - f(i))$
- $L = \nabla^T \nabla$: Laplacien

Définition

Transformée de Fourier sur les graphes

- Vecteurs propres du Laplacien u_k : **base de décomposition de Fourier**
- $\hat{f}(\lambda_k) \doteq \sum_{i \in V} f(i) u_k(i) = < f | u_k >$: **changement de base**
- $f(i) \doteq \sum_{k=0}^{N-1} \hat{f}(\lambda_k) u_k(i)$: **transformation inverse**

Modes de Fourier

- **Energie de Dirichlet** :

$$< f|Lf > \doteq f^T L f = \sum_{(i,j) \in E} A_{ij}(f(j) - f(i))^2 = \|\nabla f\|_2^2$$

- **Interprétation** : Mesure des oscillations sur le graphe.
- **Théorème MiniMax** : obtention des valeurs propres de L avec certaines valeurs de l'énergie de Dirichlet.
- Plus valeur propre est **grande**, plus énergie est **élevée**.

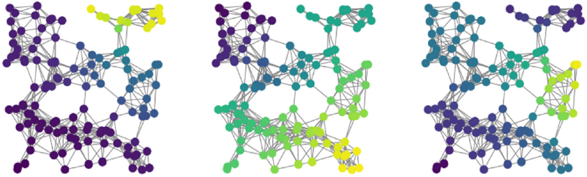


FIGURE 5 – Modes de fourier dans un graphe : u_1, u_2, u_3.

Application Graphes orientés

- **Particularité des graphes non-orientés à poids réels** : matrice d'adjacence est symétrique réelle
  $\rightarrow$ permet de toujours diagonaliser le laplacien dans $\mathbb{R}$ (théorème spectral).
- **Cela n'est plus le cas pour les graphes orientés** et il faut trouver un moyen de contourner cela : solution réside dans les marches aléatoires.

Opérateur de Marche aléatoire

On considère une **marche aléatoire sur un graphe dirigé fortement connexe** G = (V, E, w) (chaîne de Markov homogène à espace d'états fini et dont les probabilités de transitions sont proportionnelles aux poids des arêtes).

Matrice de Transition

Pour un graphe dirigé la **matrice de probabilité de transition** est définie par :

$$P_{i,j} = \mathbb{P}(X_{n+1} = i | X_n = j) = (D^{-1}A)_{i,j}$$

Avec D la matrice diagonale des degrés sortants des noeuds et A la matrice d'adjacence du graphe.

Ergodicité d'une marche aléatoire

‘

Ergodicité

Une marche aléatoire est dite **ergodique** si elle est **irréductible et apériodique**.

Irréductibilité d'une marche aléatoire

- Une marche aléatoire est **irréductible** si la **probabilité d'atteindre un noeud x depuis un noeud y en un temps fini est strictement positive**.
- On a vu que cela était **équivalent à avoir un graphe fortement connexe**.
- La **matrice de google est irréductible**.

Apériodicité

Apériodicité

- Une marche aléatoire est apériodique si :

$$\forall x \in V, \rho(x) = pgcd\{n \in \mathbb{N}^* : P^n(x,x) > 0\} = 1.$$

- Cela correspond à une marche aléatoire où l'**on ne peut pas revenir sur un noeud pour un multiple d'une période**.

$\rightarrow$ Comment faire pour rendre une marche aléatoire apériodique ?

Apériodicité

Lazy random Walk

- On définit la marche aléatoire apériodique suivante :

$$\tilde{P} = \frac{Id + P}{2}$$

- Il est facilement vérifiable que cette **marche aléatoire est apériodique** : modifier la matrice de transition de la sorte revient à ajouter des arcs bouclés au graphe de départ.

Intérêt de l'ergodicité

- Si la **matrice de transition $\mathbb{P}$ est diagonalisable, le théorème de Perron Frobenius s'applique**.
- Une marche aléatoire ergodique possède vers une distribution stationnaire de probabilité (ie (P^n) converge).

Manque une notion de réversibilité pour définir correctement le laplacien

Réversibilité

Marche aléatoire renversée

- La **marche aléatoire renversée pour une marche aléatoire ergodique** est définie par la matrice de transition suivante :

$$P^*_{i,j} = \frac{\pi_i}{\pi_j} \times P_{j,i}$$
$$P^* = \Pi^{-1} P^T \Pi$$

- π est la distribution stationnaire de probabilité.
- **Une marche aléatoire est réversible si** $P = P^*$

Marches aléatoire sur graphes non-dirigés sont réversibles, l'inverse n'est pas vrai.

Reversibilisation

Additive reversibilisation

- On rend la matrice de transition d'une marche aléatoire ergodique réversible :

$$\bar{P} = \frac{P + P^*}{2}$$

Comment définir le Laplacien ?

Laplacien sur Graphe orienté

- Le **Laplacien du graphe est :**

$$L = \Pi L_{RW}$$

On a noté L_{RW} le laplacien de la marche aléatoire.

$$L_{RW} = I - \bar{P}$$

- Généralisations de propriétés obtenues pour les graphes non-orientés
- L'on retrouve les notions d'Energie de Dirichlet définies ci-dessus.
- On peut définir le gradient d'une façon similaire et on retrouve l'idée d'oscillation

Algorithmes

- **Classique** : L'algorithme de pagerank est en open source donc nous avons pu le trouver facilement
- **Quantique** : Pas de documentation précise sur le code. Travail basé sur le projet de l'année dernière et utlisation d'IBM pour mieux implémenter le code.
- En parallèle nous essayons d'utiliser le créateur de circuit d'IBM pour créer des circuits comme celui de groover :

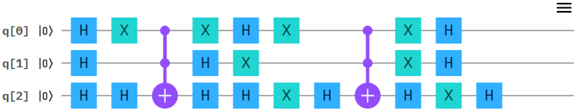


FIGURE 6 – Circuit de Groover

Jalons principaux

- Effectuer une **approximation de faible rang** de la matrice du Laplacien de la marche aléatoire. (théorème d'Eckart-Young)
- Cette approximation s'effectue par SVD ou PCA.
  $\rightarrow$ Il existe **plusieurs algorithmes quantiques qui réalisent ces opérations en O(log(N))** avec N la taille de la matrice de départ (voir par exemple Quantum Higher Order Singular Value Decomposition de Lejia Gu, Xiaoqiang Wang, H. W. Joseph Lee, and Guofeng Zhang.)
- L'importance résultante sera une **somme des vecteurs singuliers obtenus par PCA**.
  $\rightarrow$ Pondérer les vecteurs de fréquence différentes par un coefficient relatif à leurs participation au signal de départ.

Difficultés

- Réussir à **mettre le système dans un état superposé** pour **obtenir le résultat souhaité**.
  $\rightarrow$ Chaque algorithme quantique de SVD a sa propre manière de procéder : trouver la manière qui **correspond le plus à nos données initiales** et qui soit **réalisable avec un nombre de Qbits raisonnable** (le laplacien est une matrice de taille de l'ordre de 10^9...)
- **Quantifier les résultats** de notre algorithme par rapport à l'algorithme classique de PageRank : ici l'idée principale de l'algorithme change et il convient de quantifier les modifications que cela induit.
- **Choix des coefficients de pondération** qui ne peut être arbitraire. Quel jeu de coefficients maximise la pertinence des résultats sans nuire aux hubs secondaires ?

Remerciements

Pour finir, nous souhaitons remercier tout particulièrement Zeno Toffano et Benoit Valiron, chargés de l'encadrement de notre projet, pour leur présence et implication tout au long de cette étude.