Fouille de structure web:

algorithmes PageRank et HITS, découverte de communauté web, modélisation par les graphes, classification par information des liens.

Introduction 1/2

- Les premiers moteurs de recherche, comme Voila, AltaVista, Excite, .., retrouvaient les documents pertinents seulement en fonction de la similarité du contenu avec la requête. Leur méthode est basée sur le nombre d'occurrences des termes de la requête dans les documents, proximité, l'emplacement dans le texte …

Problème

- **Spamming** Facile à falsifier pour placer les pages sur la première page de résultat (la seule lue dans 95% des cas)
- **répéter les mots importants**: soit dans l'en-tête, soit dans le texte (texte en blanc sur fond blanc).
- **Cloaking:** envoyer une page différente (spam) au moteur de recherche pour l'indexer.
- **Redirection:**

Definition

Web Structure Mining: découvrir des données utiles à partir de liens hypertextes.

- Il examine la topologie et la structure des liens entre les pages web. Cela inclut l'analyse des liens hypertexte, des structures de site web, des liens entre les différents sites, etc. Le but est de comprendre la manière dont les pages sont liées entre elles et d'extraire des connaissances à partir de ces relations.

Les objectifs du web structure mining:

- Analyser la topologie du Web
- Détecter les structures communautaires
- Classification des pages web
- Détection de liens importants
- Optimisation de la navigation
- Prévenir la fraude en ligne : anomalies
- Optimiser les moteurs de recherche

Web et reseaux sociaux

- Nous pouvons considérer le web comme un réseau social où les nœuds représentent des documents et les liens sont des citations d'un document vers d'autres documents.. Ainsi, les mesures de popularité, d'autorité et de prestige peuvent être utilisées pour classer les pages web.
- Au cours de la période 1997-1998, deux des algorithmes de recherche d'information basés sur les liens les plus influents ont été proposés : PageRank et HITS. Ces deux algorithmes ont été développés à l'origine pour l'étude des réseaux sociaux.
- Sergey Brin et Lawrence Page, étudiants à Stanford, ont trouvé une solution aussi originale que simple "PageRank" : utiliser l'information des liens entre les pages pour mesurer l'importance des sites. Après cela, ils ont créé la société Google.

modélisation par les graphes

les pages web sont reliées entre elles par des liens hypertextes formant un graphe web.

- Le graphe Web partage diverses propriétés graphiques avec d'autres types de réseaux complexes, par ex. un réseau de citations, un réseau électrique, etc.
- Plusieurs parties du graphe Web ont été signalées comme ayant une connectivité en loi de puissance et une structure macroscopique qui sont différentes des propriétés d'un graphe aléatoire.

Prestige 1/2

- Le prestige est une relation récursive, c'est-à-dire qu'il dépend de l'autorité (ou du prestige) des citations (liens entrants). La popularité d'une page est d'autant plus importante que de nombreuses pages populaires la référencent.
- Considérons la matrice d'adjacence A du graphe des citations de documents défini comme suit : A(u, v) = 1 si le document u cite le document v et A(u, v) = 0 sinon. Le prestige du nœud **u** est défini comme suit : $p(u) = \sum_{v} A(v,u)\, p(v)$ ou $\acute{P} = A^T P$
- Le recalcul du vecteur de prestige **P** un certain nombre de fois conduit à un point fixe, qui est la solution de l'équation

$$\lambda P = A^T P$$

il faut résoudre le système:

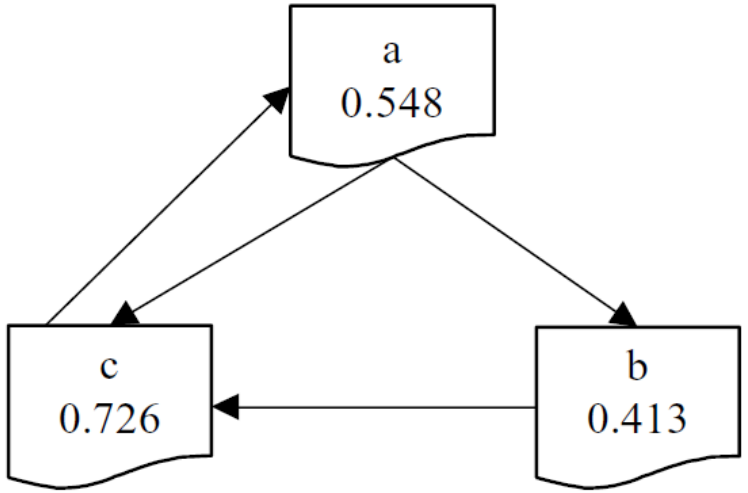
$$|A - \lambda I| = 0$$

Prestige 2/2

Mais, on s'intéresse seulement au vecteur propre dominant, le vecteur propre associé à la plus grande valeur propre.

Methode

- $P \leftarrow P0$
- loop:
  - $Q \leftarrow P$
  - $P \leftarrow AP$
  - $P \leftarrow P / \|P\|$ // normaliser P
- while $\| P - Q \| > \varepsilon$



Exemple:

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

PageRank 1/3

Une recherche se décompose en deux étapes :

- la sélection des pages contenant les mots-clés de la requête.
- Le classement, par ordre décroissant, des pages concernées selon leur valeur de PageRank. Cette valeur aura bien entendu été calculée précédemment et n'est pas recalculée à chaque requête.

Supposant qu’on un vecteur **v** qui représente la probabilité de présence d'un internaute sur tous les page. Si on prend l’exemple précèdent et disant que l’internaute est sur la page **b** alors V=[0 1 0]

Si on multiplie V par A : V' = V*A alors on obtient un vecteur V’ = [0 0 1], ce qui représente la probabilité de présence de l'internaute sur les pages après un clic. Si on suppose que l’internaute est sur la page **a** alors v=[1 0 0] si on multiplie par A alors v=[0 1 1] pour obtenir une probabilité alors nous devons normaliser la matrice A en divisant chaque composante par la somme des valeurs de la ligne correspondante,

PageRank 2/3

Pour obtenir une vision globale de la popularité des pages, il suffit maintenant de supposer que les internautes sont initialement répartis uniformément sur le réseau et on les laisse se déplacer jusqu'à leur répartition (le vecteur V) se stabilise au fur et à mesure des itérations de multiplication avec A.

$$R(u) = \lambda \left[\frac{A(v,u)}{N_v} R(u) + E(u) \right]$$

$$A = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

R=[0.39 0.21 0.40]

- L'utilisation de la source de classement E résout le problème avec le rank sink et permet à l'algorithme de fonctionner avec un graphe avec des parties déconnectées

PageRank 3/3

$$r = P*(A'*(r./d)) +(1-P)/n$$

- *r est un vecteur de scores de PageRank.*
- *P est un facteur d'amortissement scalaire (généralement 0,85), qui est la probabilité qu'un internaute au hasard clique sur un lien de la page en cours, au lieu de continuer sur une autre page au hasard.*
- *d est un vecteur contenant le degré extérieur de chaque nœud du graphe. d est mis à 1 pour les nœuds sans liens sortants.*
- *n est le nombre de nœuds du graphe.*

Positives:

- Sa capacité à lutter contre le **spam** : il n'est pas facile pour le propriétaire d'une page Web d'ajouter des liens internes depuis d'autres pages importantes sur sa page.
- C'est une mesure **globale** et indépendante de la requête.

Negatives:

- Il ne peut pas faire la **distinction** entre les pages qui sont généralement **autoritaires** et celles qui le sont sur le sujet de la requête.
- Il ne prend pas en compte le facteur **temps**.

HITS 1/3

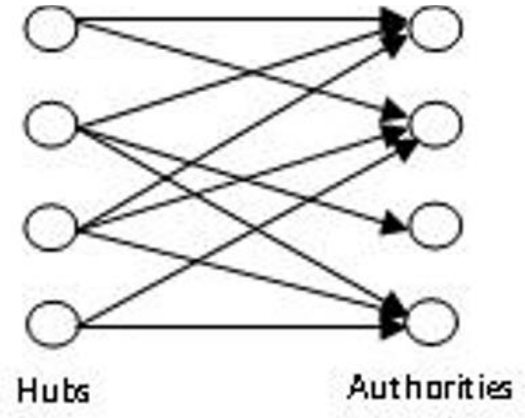
HITS stands for Hypertext Induced Topic Search.

HITS est un algorithme de classement dépendant de la requête de recherche.

Lorsque l'utilisateur lance une requête de recherche, HITS élargit d'abord la liste des pages pertinentes renvoyées par un moteur de recherche, puis produit deux classements de l'ensemble élargi de pages, le classement d'autorité et le classement du hub.

- Une **autorité** est une page avec de nombreux liens entrants.
  - la page peut avoir un bon contenu pour un certain sujet et donc beaucoup de gens lui font confiance et créent des liens vers elle.
- Un **hub** est une page comportant de nombreux liens sortants.
  - Il sert d'organisateur des informations sur un sujet particulier.

L'idée clé de HITS est qu'un bon hub pointe vers plusieurs bonnes autorités et qu'une bonne autorité est pointée par plusieurs bons hubs.



HITS 2/3

- Il comporte deux parties :

1. Construire un sous-graphe WWW ciblé appelé S,
   - Tout d'abord, om utilise un moteur de recherche pour obtenir l'ensemble de base des pages pertinentes W avec un score de classement élevé.
   - Ensuite, l'ensemble racine W est étendu en incluant toute page pointée par une page dans W et toute page qui pointe vers une page dans W.
2. Calculer les hubs et les autorités.

$$a = A^T h$$
$$h = A\ a$$

Les scores d'autorité et de hub peuvent être calculés en utilisant la Méthode de la puissance itérée

les solutions finales des processus d'itération obtenues par substitutions sont

$$a_k = A^T A\ a_{k-1}$$
$$h_k = A\ A^T\ h_{k-1}$$

HITS 3/3

L'algorithme d'itération de puissance pour HITS est le suivant

$a_0 = h_0 = \{1,1,\ldots,1\}$

$k=1$

repeat

$a_k = A^T A\, a_{k-1}$

$h_k = A\, A^T\, h_{k-1}$

$a_k = a_k / ||a_k||$

$h_k = h_k / ||h_k||$

$k \leftarrow k+1$

until $|| a_k - a_{k-1}|| < \varepsilon$ **and** $|| h_k - h_{k-1}|| < \varepsilon$

Les principales **faiblesses** de l'algorithme HITS sont

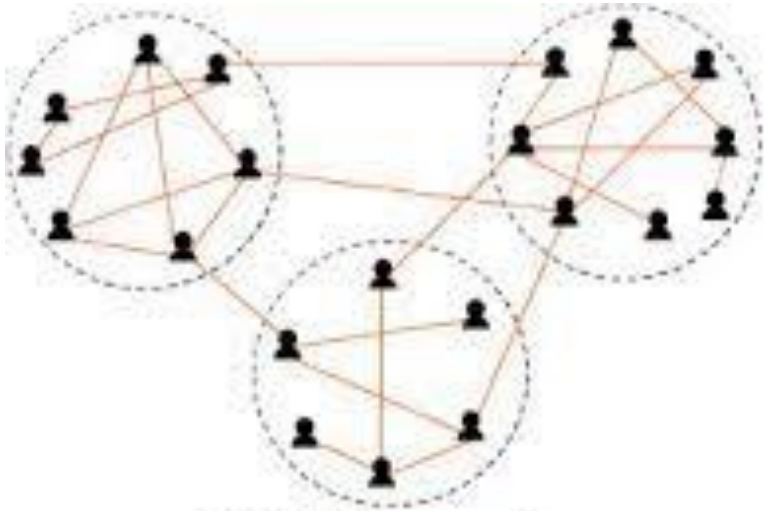
- Absence de la fonction anti-spam.
- L'expansion de l'ensemble racine peut inclure de nombreuses pages de bruit et non pertinentes
- La complexité.

Découverte de Communauté web

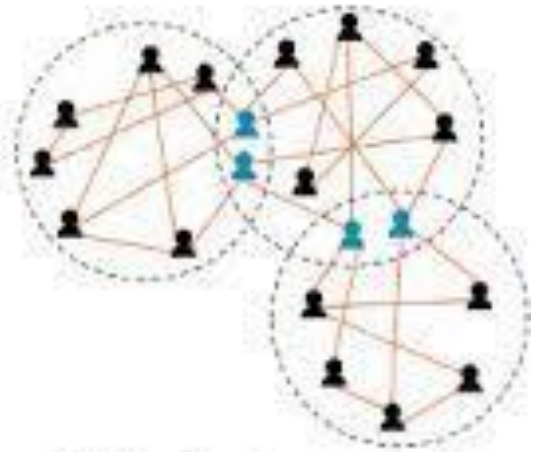
Outre le classement des résultats de recherche, les hyperliens sont également utiles pour découvrir les communautés du Web.

- une **communauté** est simplement un groupe d'entités (par exemple, des personnes ou des organisations) qui partagent un intérêt commun ou participent à une activité ou un événement.
- une **communauté de pages web** est un groupe ou ensemble de pages web qui présentent des liens étroits ou des relations significatives entre elles. Ces communautés peuvent se former en raison de similitudes de contenu, de thèmes communs, ou de liens réciproques fréquents.

- on appelle **communautés** des **sous-graphe**s dont la densité interne (liens entre leurs nœuds) est très supérieure à la densité externe (liens avec le reste du graphe).



(a) Disjoint communities



(b) Overlapping communities

pourquoi on s’intéresse par la découverte de ces communautés.

Dans le contexte du web

1. Les communautés fournissent des ressources d'information précieuses et actualisées à un utilisateur.
2. Analyse thématique
3. Détection d'anomalie
4. Amélioration des moteurs de recherche
5. Comprendre l'évolution du Web
6. Personnalisation de contenu
7. Les communautés permettent de cibler la publicité à un niveau très précis.

Méthodes pour la détection de communauté

Plusieurs méthodes ont été proposé :

1. Les méthodes de **partitionnement** de graphes: La plupart de ces méthodes sont basées sur la division itérative des graphes en deux groupes distincts, tout en essayant de minimiser le nombre des arrêtes à couper entre ces groupes. Généralement, il faut fournir à l'algorithme le nombre de clusters et leurs tailles.
2. Les algorithmes de classification **hiérarchique:** on utilise une fonction de similarité pour regrouper (agglomératives) ou séparer (divisives) les nœuds.
3. Les méthodes basées sur **l'optimisation**: voient la détection de communauté comme un problème de maximisation d'une fonction de qualité comme GA. SA, PSO.
4. Autre méthodes comme LPA RW …

The label propagation algorithm LPA

1. Initialize the unique label for each node in the network.
2. Arrange the nodes of the network in random order.
3. For each node $x \in X$, iteratively update the node label so that each node takes the label that is carried by the largest number of its adjacent nodes.
4. If the label of each node is the same as that of most of its neighboring nodes, then the nodes with the same label are placed in the same community, and the algorithm ends; otherwise, go to step (2).