Authoritative Source in a Hyperlinked Environment

JON M. KLEINBERG
Cornell University

Journal ACM 1998

CHEN Ang
Le 13 Jan, 2003

Étapes à suivre

- Introduction
- Construction d'un sous graphe www
- Un algorithme itérative (HITS)
- Applications
  - Requête sur similarité
  - Multiple ensembles de Hubs et Autorités
- Diffusion et généralisation
- Évaluation
- Conclusion

Introduction

- L'environnement
  - Complicité énorme et entrain d'augmenter
  - Hypertexte: texte, image, multimédia
  - Hyperliens (URL): liens entres les hypertextes
- Objective: Trouver les pages qui satisfait le requête d'utilisateur
- Les requêtes
  - Requête spécifique
    - « Est-ce que Netscape support JDK 1.1 code signing API ? »
    - ***Problème de disette***: peu de page contient l'information utile et il est difficile de les trouver.
  - **Requête général sur un terme**
    - « Je veut toute information du langage de programmation Java »
    - ***Problème d'abondance***: trop de page relevant.
    - Notion d'autorité: étant donné un page particulière, comment on décide si il est autoritaire ou non ?
  - Requête de similarité
    - « Trouvez les sites qui sont similaires à java.sun.com »
    - Dans Google: **related:java.sun.com/**

Introduction

- Statistique du texte
  - Une solution classique mais pas bonne
  - Problème de Spam
- Observation
  - En général, un autorité n'est pas le site qui utilise le terme le plus souvent. Ex. pour les mots comme « search engines », « car maker » .
- L'analyse de structure des liens
  - Hyperlien encode le jugement de humain, qui nous permet de trouver l'autorité.
  - Obstacle: une lien peut être crée en diverses raisons.(Ex. navigation , publicité)
  - Différence entre la relevance et la popularité.

Construction d'un sous graphe www

Requête général sur un terme

Obtenir les pages relevantes: un sous graphe du Web

- Idéalement, le sous graphe S:
  i. S est relativement petit
  ii. S est riche en pages pertinentes
  iii. S contient la plus part (ou beaucoup) des autorités fortes

Condition i. garanti le temps de calcul
Condition ii. facilite le recherche d'autorité

Une graphe qui satisfait ces deux conditions peuvent être obtenues par un moteur de recherche textuel. On le dénote R, la racine du graphe.

Il faut développer R afin de satisfaire la condition iii.

Construction d'un sous graphe www

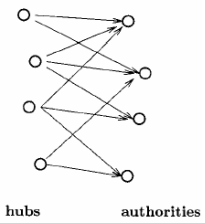
SubGraph(Str, Engine, T, D)
***Str:* le terme de requête**
***Engine*: moteur de recherche textuel**
***T,D*: nombre naturel**
R* dénote les premières *T* résultats de l'*Engine* recherche sur *Str.
Set S=R
Pour chaque page *p* dans *R*
  ***Out(p)* dénote l'ensemble de page *p* pointe vers**
  ***In(p)* dénote l'ensemble de page qui pointent au p**
  Ajoute toutes les page de *Out(p)* dans *S*
  Si |*in(p)*| <= *D* alors ajoute les pages de *In(p)* dans *S*
  sinon on choisit *D* pages arbitrairement et les ajoute dans *S*
Return *S*

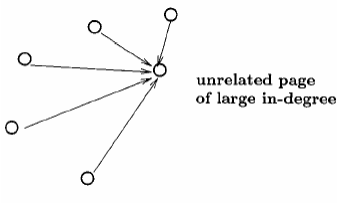
T=200, D=50
|S|=1000-3000

- Post-traitement du graphe: Heuristiques
  - Lien de navigation: enlever les liens intrinsèques.
  - Publicité ou autre type d'info non utile: beaucoup de page d'un seul domaine pointent vers une page. Limiter le nombre de liens entre les pages d'un domaine vers une seule page particulière.

Hubs et autorités

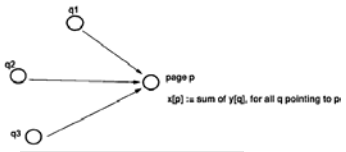
- Un bon Hub (par rapport à une requête donnée) est une page qui pointe vers de nombreuses bonnes autorités.
- Réciproquement, une bonne autorité est pointée par de nombreux (bon) Hub.
- Les deux notions se renforcent mutuellement.





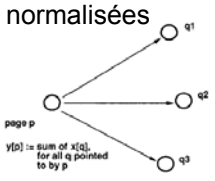
Un algorithme itérative

- On applique un algorithme itérative sur le sous graphe
- Opération F: calcule le poids d'autorité
- Opération O: calcule le poids de hub
- A chaque étape les vecteurs sont normalisées



$$x^{\langle p \rangle} \leftarrow \sum_{q:(q,p)\in E} y^{\langle q \rangle}.$$

$$\sum_{p \in S} (x(p))^2 = 1$$



$$y^{\langle p \rangle} \leftarrow \sum_{q:(p,q)\in E} x^{\langle q \rangle}.$$

$$\sum_{p \in S} (y(p))^2 = 1$$

Un algorithme itérative

Iterate(**G**,**k**)
G: le sous graphe, une collection de n page avec des liens
k: nombre naturel
x: Vecteur de poids d'autorité pour toutes les pages
y: Vecteur de poids de hub pour toutes les pages
Initialise **x** et **y** à {1,1…,1}
Applique **k** fois l'opération **F** et **O** itérativement et chaque fois normalise le vecteur
Return les deux vecteurs: autorité **x** et hub **y**.

Filter(**G**,**k**,**c**) Typiquement, c=5 ~10
k,**c**: nombre naturel, G: le sous graphe
(**x**,**y**)=Iterate(**G**,**k**)
Les **c** coordonnées ayant les plus grands valeurs dans **x** et **y** sont des autorité et des hubs

Un algorithme itérative

- A: matrice adjacent N*N, N: nombre de page

$$\mathbf{x} \leftarrow \mathbf{A}^T\mathbf{y} \qquad \mathbf{y} \leftarrow \mathbf{A}\mathbf{x}$$

$$\mathbf{x}_k \leftarrow \left(\mathbf{A}^T\mathbf{A}\right)^{k-1}\mathbf{A}^T\mathbf{z} \qquad \mathbf{y}_k \leftarrow \left(\mathbf{A}\mathbf{A}^T\right)^k\mathbf{z}$$

$\{\mathbf{x}_k\}$ converge à la limite $\mathbf{X}^*$
$\{\mathbf{y}_k\}$ converge à la limite $\mathbf{Y}^*$
$\mathbf{X}^*$: eigenvector principale de $\mathbf{A}^T\mathbf{A}$
$\mathbf{Y}^*$: eigenvector principale de $\mathbf{A}\mathbf{A}^T$

Vecteur z: {1,1,…1}.
Souvent, k=20 itérations sont suffisants pour que la valeur soit stable.

Expriment

- (java) Authorities
- .328 http://www.gamelan.com/ Gamelan
- .251 http://java.sun.com/ JavaSoft Home Page
- .190 http://www.digitalfocus.com/digitalfocus/faq/howdoi.html The Java Developer: How Do I . . .
- .190 http://lightyear.ncsa.uiuc.edu/;srp/java/javabooks.html The Java Book Pages
- .183 http://sunsite.unc.edu/javafaq/javafaq.html comp.lang.java FAQ

- ("search engines") Authorities
- .346 http://www.yahoo.com/ Yahoo!
- .291 http://www.excite.com/ Excite
- .239 http://www.mckinley.com/ Welcome to Magellan!
- .231 http://www.lycos.com/ Lycos Home Page
- .231 http://www.altavista.digital.com/ AltaVista: Main Page

- (Gates) Authorities
- .643 http://www.roadahead.com/ Bill Gates: The Road Ahead
- .458 http://www.microsoft.com/ Welcome to Microsoft
- .440 http://www.microsoft.com/corpinfo/bill-g.htm

Applications

Requête sur la similarité

- Trouver les sites similaires à une site donnée
  - Appliquer l'algorithme directement sans modification majeure.
  - Au lieu de demander « Donne-moi T pages qui contient le mot 'java' au moteur de recherche textuel,», on demande « Donne-moi T pages qui pointent vers la page java.sun.com». On aura un racine du graphe R différence.
  - Regarder le in-degree: pas la bonne solution
  - Appliquer la suite de l'algorithme HITS et retourne les autorité les plus forts.
- Avec google: « similaire pages »

Expriment: similarité

- (www.honda.com) Authorities
- .202 http://www.toyota.com/ Welcome to @Toyota
- .199 http://www.honda.com/ Honda
- .192 http://www.ford.com/ Ford Motor Company
- .173 http://www.bmwusa.com/ BMW of North America, Inc.
- .162 http://www.volvocars.com/ VOLVO
- .158 http://www.saturncars.com/ Welcome to the Saturn Web Site
- .155 http://www.nissanmotors.com/ NISSAN—ENJOY THE RIDE
- .145 http://www.audi.com/ Audi Homepage
- .139 http://www.4adodge.com/ 1997 Dodge Site
- .136 http://www.chryslercars.com/ Welcome to Chrysler

Applications

Multiple ensembles de Hubs et Autorités

- Le cas où le mot clé est ambiguë
  - Ex. « jaguar »
    - Mac OS X v10.2
    - L'équipe de football
    - Animal
    - Un ancien marque d'ordinateur
    - Automobile
  - Un terme général dont la signification dépend le contexte, utilisé dans différences domaines.
    - Ex. « Randomized algorithmes»
- Appliquer l'algorithme en utilisant les eigenvectors non principales, positive et négative.
- Intuitivement pas très claire, mais résultat intéressant.

Expriment: multiple ensemble d'autorité et hub

- (jaguar*) Authorities: principal eigenvector
- .370 http://www2.ecst.csuchico.edu/;jschlich/Jaguar/jaguar.html
- .347 http://www-und.ida.liu.se/;t94patsa/jserver.html
- .292 http://tangram.informatik.uni-kl.de:8001/;rgehm/jaguar.html
- .287 http://www.mcc.ac.uk/dlms/Consoles/jaguar.html *Jaguar Page*

- (jaguar*) Authorities: 2nd nonprincipal vector, positive end
- .255 http://www.jaguarsnfl.com/ *Official Jacksonville Jaguars NFL Website*
- .137 http://www.nando.net/SportServer/football/nfl/jax.html *Jacksonville Jaguars Home Page*
- .133 http://www.ao.net/;brett/jaguar/index.html *Brett's Jaguar Page*
- .110 http://www.usatoday.com/sports/football/sfn/sfn30.htm *Jacksonville Jaguars*

- (jaguar*) Authorities: 3rd nonprincipal vector, positive end
- .227 http://www.jaguarvehicles.com/ *Jaguar Cars Global Home Page*
- .227 http://www.collection.co.uk/ *The Jaguar Collection—Official Web site*
- .211 http://www.moran.com/sterling/sterling.html
- .211 http://www.coys.co.uk/

Diffusion et généralisation

- Diffusion
  - Le calcul est sur le sous graphe G sans regarder le contenu de la page.
  - G contient beaucoup de termes différentes.
  - Lorsque le terme de recherche est étroit, le résultat obtenue est dominé par un terme plus large et populaire.
  - Ex. www conférence
    - www est plus populaire que www conférence dans le sous graphe
    - www est le terme plus compact, donc le résultat obtenu dépend le terme www.
    - Solution: utilise eigenvecteur non principale + term-matching
- Généralisation
  - Malgré l'inconvénient de diffusion, on a trouvé un moyenne de généraliser un requête spécifique.

Évaluation

- Avec CLEVER système d'IBM, qui utilise une extension de HITS.
  - L'objective du CLEVER: automatic resource compilation, construire une liste de web page de haut qualité comme les catégories de Yahoo mais automatiquement.
  - Mélange les résultats sur 26 termes
    - CLEVER: les première 5 autorités et 5 hubs.
    - Les pages dans les catégories de Yahoo.
    - Regard aussi les 10 pages retournés par AltaVista.
  - Assemblé 37 utilisateurs: Il ne sont pas d'experts d'informatique ni dans les domaine de 26 termes. Totalement 1369 réponses.
  - Juger les résultats comme: *mauvais, ca va, bien, fantastique*
  - Équivalant sur 31% de réponse
  - CLEVER évalue plus haut sur 50% de réponse
  - Yahoo évalue plus haut sur autre 19% de réponse.

  Mais:
  - Il n'existe pas une fonction de la qualité de jugement bien définie, donc on peut pas tirer une conclusion irrévocable.
  - La soumission d'un page Web est faite souvent à l'extérieur de Yahoo

Conclusion

- HITS: un algorithme général dans l'environnement hypertexte.
  - Le principe: par rapport à une requête donnée, on construit à l'aide d'un moteur de recherche classique un ensemble de pages qui servira de base à l'analyse des liens. Puis, grâce à un algorithme de point fixe traduisant la réciprocité entre bonnes autorités et bon hubs, on extrait les meilleurs sites pour la requête.
  - Hypothèse: les bonnes autorités sont toujours pointés par les bons hubs, ils sont deux notions mutuellement renforcés.
  - Algorithme itérative ne regard pas le contenu de page, problème de diffusion.
- Beaucoup d'utilisateur indique que la liste de page retourné sont des points de départ d'exploration, mais ils visitent beaucoup de pages qui ne sont pas dans la liste.
- C'est un processus naturel pendant l'exploration sur le Web. On va pas le remplacer mais le faciliter.