PageRank thématique

Michel Chilowicz

8 mars 2006

1 Introduction

Nous discutons ici de l'article *Topic-Sensitive PageRank* [3] de Haveliwala concernant l'adaptation de l'algorithme d'attribution de score de réputation *PageRank* pour prendre en compte la dimension thématique des pages. Dans un premier temps nous rappellerons le fonctionnement de l'algorithme *PageRank* utilisé pour quantifier la réputation des pages pour le moteur de recherche Google puis ensuite nous nous intéresserons à l'adaptation réalisée par Haveliwala pour le calcul de *PageRank* thématique ainsi que les principaux résultats qui s'en dégagent.

2 L'algorithme PageRank

2.1 Principe

L'algorithme *PageRank*, proposé par L. Page et S. Brin [7] permet d'attribuer un score de réputation pour chaque page présente sur le Web. Cet algorithme quantifie la réputation d'une page par le nombre de liens hypertextes qui pointent vers celle-ci : ainsi une page comportant de nombreux liens entrants est considérée comme très populaire et donc jouissant d'une réputation élevée. Concrètement un lien hypertexte d'une page i vers une page j est considéré comme un vote de i pour la page j : ainsi pour chaque page i du Web référencée par Google est calculé un vecteur de note locale de réputation c_i avec $c_{i,j} = 0$ s'il n'existe aucun lien de i vers j et $c_{i,j} = 1/L_i$ si au moins un lien existe (L_i est le nombre de liens sur la page i). Le *PageRank* R_i d'une page i est calculé par la somme des *PageRank* (pondérés par l'inverse du nombre de liens) des pages B qui la pointent 1.

$$R_i = \sum_{j \in B} \frac{R_j}{|l_j|} \tag{1}$$

La formule de calcul du *PageRank* d'une page est récursive : le *PageRank* est approximable par une méthode itérative. On initialise l'algorithme de calcul par une valeur constante non-nulle de *PageRank* pour chaque page, puis lors de chaque itération, on recalcule le *PageRank* de chaque page en utilisant la formule. On répète les itérations pour aboutir sur une convergence des valeurs de *PageRank*.

2.2 Le problème de la convergence

La formule exprimée précédemment ne permet pas la convergence du *PageRank* dans le cas où des cliques de pages tisseraient des liens entre-elles, sans lien sortant vers l'extérieur. Ces cliques (formées accidentellement ou intentionnellement pour exploiter la faille de convergence) seraient des accumulateurs de *PageRank* dont elles ne feraient pas profiter l'extérieur. La convergence du *PageRank* n'est donc pas assurée.

La solution proposée consiste à intégrer à la formule une pré-notation pour chaque page. Comment établir cette pré-notation ? Il est envisageable d'accorder une pré-notation non-nulle uniquement à certaines pages extrêmement populaires. Notons $\overrightarrow{E}$ le vecteur de pré-notation des pages, on introduit alors le *PageRank* corrigé R' 2.

$$R'_i = cR_i + (1-c)E_i \tag{2}$$

En pratique, la valeur du coefficient c utilisée est $0,85$ [7].

Finalement, si nous notons A la matrice des notes locales., alors nous obtenons l'équation matricielle 3.

$$\overrightarrow{R'} = cA\overrightarrow{R'} + (1-c)\overrightarrow{E} \tag{3}$$

R' est donc le vecteur propre dominant de A (*eigenvector*) associé à la valeur propre c. Il est possible d'approximer ce vecteur en partant d'un vecteur source R'_0 puis par itération successive $R'_{k+1} = AR'k$. Page et Brin ont montré expérimentalement qu'un vecteur propre tolérable pouvait être trouvé, pour n pages, en $O(\log n)$ itérations. L'algorithme de calcul des *PageRank* pour n pages comportant m liens[1] peut donc être exécuté en $O(m \log n)$. Diverses techniques d'optimisation peuvent être utilisées afin d'accélérer en pratique le calcul des scores de réputation des pages : certaines méthodes telles que l'approximation quadratique [4] permettent le calcul de *PageRank* approché.

La convergence ne peut être assurée que si la matrice de notes A est irréductible (le graphe des pages est fortement connexe) et apériodique [6].

2.3 Liens vers des pages non-liantes

Certains liens pris en compte pour le calcul du *PageRank* pointent vers des pages qui ne comportent aucun lien. De telles pages reçoivent une contribution d'autre pages pour leur *PageRank* mais ne la redistribuent pas : de telles pages limitent la connexité du graphe du Web et empêchent la convergence du calcul du *PageRank*. Une solution apportée à ce problème consiste à simuler la redistribution de réputation égalitaire sur toutes les pages du Web : on considère ainsi qu'une page ne comportant aucun lien sortant comporte un lien vers chacune des pages indexées.

2.4 Le modèle du surfeur aléatoire

La notion de *PageRank* permet d'introduire le modèle du surfeur aléatoire. Un tel individu parcourt le Web en suivant des liens aléatoirement : le *PageRank* d'une page reflète la probabilité qu'il atteigne sur cette page en parcourant au hasard le graphe des pages Web. L'introduction du vecteur de pré-notation $\overrightarrow{E}$ traduit la possibilité qu'il choisisse de sauter aléatoirement vers une autre page en s'y rendant directement, sans utiliser un lien hypertexte (par exemple lorsqu'il tourne en rond dans une collection de pages). Il serait possible d'envisager l'utilisation d'un vecteur $\overrightarrow{E}$ différent pour chaque utilisateur afin d'obtenir des scores personnalisés : il s'agit de l'idée utilisée par Haveliwala. Il propose ainsi d'utiliser des vecteurs $\overrightarrow{E}$ spécifiques à une thématique de recherche pour calculer différents scores de réputation.

3 PageRank thématique

Haveliwala propose une méthode afin de calculer pour chaque page du Web différentes notes de réputation, chacune spécifique à une thématique donnée.

[1] m est borné par n^2 mais $m << n^2$: la matrice A est creuse.

3.1 Intérêt de l'établissement de scores de réputation thématique

Levée de polysémie Dans certaines situations, l'utilisation de scores thématiques présentent des avantages certains. Le cas le plus intéressant est sans doute la levée d'ambiguïtés polysémiques sur des mots-clés de recherche. La connaissance de la thématique de recherche associée à l'utilisation de scores thématiques permettrait d'accorder une meilleure pertinence aux pages populaires au sein du thème envisagé. Par exemple, la recherche du terme *Java*, avec un système de réputation global, retournera probablement en premier résultat le site officiel du langage de programmation Java de Sun ; une recherche de ce terme utilisant un *PageRank* associé à la thématique du tourisme mènera plutôt à l'obtention de pages traitant de l'île indonésienne éponyme.

Catégorisation automatique Les scores de réputation thématiques permettent également de procéder à une catégorisation automatique de pages Web : on utile pour cela que la probabilité est élevée qu'un lien sortant d'une page vers une autre traite d'une thématique similaire.

Compartimentalisation thématique du PageRank Si la plupart des liens hypertextes sont réalisés entre pages de thématique similaire, ce n'est pas toujours le cas : certaines pages spécialisées sur un thème particulier — où paraissant ainsi car citées par de nombreuses pages de ce thème — peuvent contenir des liens vers des pages d'un domaine totalement différent. Il peut être utile de compartimentaliser la distribution de notes de réputation aux sites de même catégorie, cependant une telle affirmation peut entrer en contradiction avec le mécanisme de catégorisation automatique.

3.2 Principe du PageRank thématique

L'algorithme *PageRank* originel repose sur deux composantes essentielles :

- les notes locales attribuées par les pages (liens hypertextes),
- et un vecteur de pré-réputation $\overrightarrow{E}$ utilisé pour assurer la convergence du calcul du *PageRank* et définir les pages bénéficiant d'une réputation par défaut.

La thématisation du *PageRank* repose sur le calcul d'un score de réputation par thème avec l'utilisation d'un vecteur $\overrightarrow{E}$ de scores de pré-réputation personnalisés pour chaque thème : on attribue ainsi un score de pré-réputation non-nul à des sites portails importants au sein d'un certain thème ; typiquement des sites pouvant être accédés initialement par un *surfeur aléatoire* s'intéressant à un thème particulier.

3.3 Constitution de vecteurs de pré-réputation thématique

Utilisation de l'Open Directory Afin de créer un vecteur de pré-réputation thématique, il est nécessaire de déterminer des pages importantes correspondant au thème choisi. Une idée intéressante consiste à exploiter des annuaires de liens catégorisés disponibles sur le Web : afin de mener ces tests, Haveliwala décide d'utiliser le répertoire de liens *Open Directory* [1]. Cet annuaire thématique hiérarchique maintenu par des milliers d'internautes bénévoles[2] est subdivisé en 16 catégories principales[3] : Haveliwala entreprend ainsi de calculer un score de réputation *PageRank* pour chacune de ces catégories.

[2] 71 337 éditeurs le 6 février 2006

[3] Ces 16 catégories sont : *Arts*, *Games*, *Kids and Teens*, *Reference*, *Shopping*, *Business*, *Health*, *News*, *Regional*, *Society*, *Computers*, *Home*, *Recreation*, *Science*, *Sports*. On pourra noter que la catégorie *World* est spécifique : elle comprend tous les sites non-anglophones. Cette catégorie comprend des sous-catégories pour chaque langue et chaque langue reprend les catégories principales de l'annuaire. Il pourrait alors être judicieux d'intégrer les pages de la catégorie *World* dans les autres catégories principales si l'on envisage de réaliser des recherches dans d'autres langues que l'anglais.

Vecteurs de pré-réputation On créé pour chacune des catégories principale c_j de l'*Open Directory* un vecteur de pré-réputation v_j que l'on définit ainsi (avec T_j l'ensemble des pages citées dans l'annuaire sous la catégorie c_j) :

$$v_{j,i} = \begin{cases} \frac{1}{|T_j|} & i \in T_j \\ 0 & i \notin T_j \end{cases} \tag{4}$$

Un vecteur de pré-réputation non-biaisé est également calculé pour les tests : avec T l'ensemble des pages citées dans l'annuaire, $v_i = 1/|T|$ si la page i est présente dans le répertoire, sinon v_i est nul.

Résistance à des attaques des mainteneurs de l'Open Directory

Tout mainteneur humain d'un annuaire possède une part de subjectivité : le classement d'un site dans une catégorie spécifique n'est pas toujours évident. De plus, certains mainteneurs malhonnêtes peuvent volontairement insérer des pages dans une mauvaise catégorie, voire tout simplement incorporer des sites de mauvaise qualité dans l'annuaire. Il serait donc intéressant de rendre les vecteurs de pré-réputation thématiques moins sensibles au contenu de l'annuaire.

Une solution proposée consiste à répartir les notes de pré-réputation sur toutes les pages du Web : les pages référencés dans l'*ODP* servent alors de jeu d'apprentissage afin d'entraîner un système de catégorisation de pages (par exemple un filtre baysien utilisant la fréquence des mots pour chacune des catégories). Une page i se voit alors attribuée un vecteur w_i associant pour chacune des catégories un poids (avec normalisation unitaire de la somme des poids) : la note de pré-réputation de la page i pour la catégorie j est alors $p_{i,j} = w_{i,j}$. Cette solution permet d'accorder un score de pré-réputation non nul à des pages non-listées dans l'*ODP* et diminue l'importance relative des pages de l'*ODP*.

4 Contextualité de recherche

Afin d'utiliser de manière appropriée les *PageRank* calculés pour chacune des catégories définies, il est nécessaire de comprendre la contextualité de chacune des requêtes de recherche et de la traduire en une combinaison linéaire de thèmes.

4.1 Catégorisation de la recherche

On propose de définir un contexte de recherche par un ensemble de mots-clés[4] Q : il est nécessaire de déduire les probabilités $p(c_j|Q)$ d'appartenance des mots-clés du contexte à la catégorie c_j (catégorie pour laquelle on aura préalablement calculé un score de réputation pour chacune des pages). En connaissant les probabilités individuelles de chacun des mots-clés d'appartenir à une catégorie, il est possible de calculer $p(c_j|Q)$ ainsi en s'inspirant du théorème de Bayes :

$$P(c_j|Q) = \alpha \prod_i \frac{p(Q_i|c_j)}{p(Q_i)} \tag{5}$$

La méthode la plus simple afin de déterminer $p(Q_i|c_j)$ et $p(Q_i)$ consiste à indexer l'ensemble des mots-clés présents dans les pages pré-réputées de chacune des catégories : dans le cas présent, il s'agit des pages listées directement dans l'*Open Directory*. On définit alors $pQ_i|c_j$ comme la fréquence d'apparition du terme Q_i au sein des pages de la catégorie c_j : cela nécessite de maintenir, pour chaque catégorie, un index de mots associés à leur effectif. Un coefficient de normalisation α est introduit tel que $\sum_j P(c_j|Q) = 1$.

[4]Cet ensemble de mots-clés pouvant être les termes de la recherche et/ou les termes du contexte de celle-ci.

Dans le cas présent, nous calculons la probabilité d'appartenance de termes à une catégorie en considérant individuellement chacun des termes de la recherche : il pourrait s'avérer intéressant d'étudier les termes dans leur ensemble sur des pages de chacune des catégories ; une telle opération est cependant beaucoup plus coûteuse.

4.2 Contextes envisageables

Afin de calculer la probabilité qu'une requête soit corrélée à une catégorie, il est possible de se contenter uniquement des termes de recherche : cette méthode permet de privilégier les catégories dans lesquelles les termes proposés sont les plus fréquents. Par exemple, 71 pages sur 120 359 pages[5] traitent de Java dans la catégorie *Recreation* de l'*ODP* (en rapport avec le tourisme sur l'île indonésienne de Java), soit $0,06\%$, tandis que sur les 143 488 pages de la catégorie *Computers*, 4567 traitent de Java (ici le langage de programmation), soit $3,18\%$. Ainsi une recherche sur le terme *Java* attrbuerait un poids plus de 50 fois plus important à la catégorie *Computers* plutôt qu'à la catégorie *Recreation*. Il est donc nécessaire de prendre en compte également le contexte de la recherche : on peut le définir par un ensemble de mots-clé qui seront utilisés pour le calcul des probabilité d'appartenance de la requête à une catégorie.

Contextes immédiats Nous proposons ici quelques contextes pouvant définir une recherche :

- Catégorie choisie manuellement : l'utilisateur peut spécifier manuellement la catégorie dans laquelle il souhaite réaliser une recherche. Ce mode de recherche est particulièrement adapté aux requêtes formulées sur une page de catégorie d'un annuaire.
- Recherche de termes depuis une page[6] : si la page depuis laquelle la recherche est réalisée a été indexée et catégorisée, il est possible d'accorder un poids plus important à la thématique de la page. Pour déterminer celle-ci, plusieurs solutions peuvent être envisagées :
  - *La catégorisation hors-ligne de la page.* Chaque page se voit alors attribuer un vecteur de catégorie v avec v_j la probabilité d'appartenance de la page à la catégorie j : cette probabilité peut être calculée avec la méthode baysienne exposée précédemment en considérant l'ensemble des mots présents sur la page
  - *La catégorisation en ligne sur fenêtre contextuelle.* La considération d'une fenêtre autour des termes sélectionnés sur une page pour une recherche peut s'avérer généralement plus avantageuse que la prise en compte de l'ensemble des mots présents sur celles-ci. En effet, certaines pages éclectiques peuvent aborder différentes thématiques dans plusieurs parties et n'être que difficilement catégorisable globalement (c'est le cas par exemple de certaines pages personnelles). L'utilisation d'une fenêtre réduite autour des termes sélectionnées pour la détermination du contexte apparaît alors plus appropriée.

Utilisation de pré-contextes Il est possible de considérer un pré-contexte pour la recherche en étudiant les centres d'intérêt de l'internaute : cette étude permet d'accorder un poids plus ou moins important à chacune des catégories. Ce profil de l'internaute peut être réalisé en récoltant l'historique des pages visitées par celui-ci ou alors simplement par la connaissance des requêtes antérieures. Ces deux solutions peuvent être facilement mises en œuvre dans un moteur de recherche par l'usage d'un identifiant de connexion (cookie) ou alors en imposant l'enregistrement et l'identification de l'internaute. Un moteur de recherche ne peut connaître l'historique complet des pages visitées par l'internaute, mais il lui est néanmoins possible d'introduire dans ses pages de résultats des liens indirects vers les sites, avec enregistrement des liens suivis. L'usage de telles données induit une problématique de protection des données personnelle et de respect de la vie privée. Il serait préférable de déléguer l'étape de calcul des poids préliminaires de chacune des catégories à une application locale, ces poids étant ensuite communiqués au moteur qui n'aurait pas connaissance des recherches antérieures ainsi que des sites visités.

[5] Les données exprimées pour notre exemple ont été collectées le 15/02/2006 sur `http://dmoz.org` [1]

[6] Par exemple, par sélection de termes présents sur une page.

4.3 Score de réputation personnalisé

Lorsque l'internaute soumet sa requête Q auprès du moteur de recherche, celui-ci calcule les poids $P(Q, c_j)$ pour chacune des catégories c_j en considérant la contextualité de la recherche. Plutôt que d'utiliser le score de réputation général, le moteur considère un score de réputation personnalisé, basés sur les scores de réputation thématiques préalablement calculés en hors-ligne pondérés par les poids catégoriels liés à la requête et son contexte. On exprime donc ainsi le score $s(i|Q)$ le score de réputation de i pour les requêtes et contexte Q :

$$s(i|Q) = \sum js_{c_j}(i) \tag{6}$$

Les scores catégoriels pour chacune des pages étant préalablement calculés hors-ligne, le temps de calcul pour chacun des pages lors du traitement de la requête est faible. Les scores personnalisés de chacune des pages répondant à la requête étant obtenus, il peuvent être utilisés pour trier les résultats.

5 Résultats pratiques

Nous présentons ici les résultats obtenus par Haveliwala sur quelques requêtes de tests avec une base de 120 millions de pages[7] (parmi lesquels 280 000 des environ 3 millions de pages de l'*ODP*). Quelques requêtes de test[8] ont été exécutées sur cette base en utilisant des scores *PageRank* thématiques.

5.1 Mesures de similarités

Afin de comparer les différents classements induits par l'usage de *PageRank* thématiques, on introduit deux mesures de similarité sur les classements.

5.1.1 Mesure de similarité OSim

La mesure *OSim* quantifie le recouvrement parmi les k pages les mieux classées de deux classements : soit deux ensembles A et B, le recouvrement de A et B est défini par $\frac{|A \cap B|}{k}$. Cette mesure n'est pas suffisante pour quantifier les effets de l'usage de *PageRank* thématique : il est également nécessaire de prendre en compte l'ordre relatif des pages dans le classement, d'où l'introduction d'une autre mesure, *KSim*.

5.1.2 Mesure de similarité KSim

La mesure *KSim* entre deux classements reflète la probabilité qu'une paire de pages sélectionnée aléatoirement présente un ordre relatif identique dans les deux classements. Elle se définit formellement ainsi :

$$KSim(\tau_1, \tau_2) = \frac{|(u,v)/\tau_1', \tau_2' \text{avec accord sur l'ordre de } (u,v), u \neq v|}{(|U|)(|U|-1)} \tag{7}$$

avec U l'union des pages de τ_1 et τ_2, $\lambda_1 = U - \tau_1$ et $\lambda_2 = U - \tau_2$. On définit ensuite τ_i' comme l'extension de τ_i avec ajout des pages de δ_i (après les pages de τ_i).

Cette mesure permet de quantifier les variations d'ordre des résultats entre deux méthodes de classement. Si l'ordre des résultats est le même pour deux méthodes de classement, la valeur de $KSim$ atteint 1 ; si par exemple, les résultats sont classés en ordre inverse, $KSim$ est nul.

[7] Ces pages constituent la base *Stanford WebBase* utilisée par les chercheurs de cette université travaillant sur la thématique des moteurs de recherche

[8] Parmi ces requêtes de test : *affirmative action*, *alcoholism*, *lyme disease*, *vintage cars*, *java*, ...

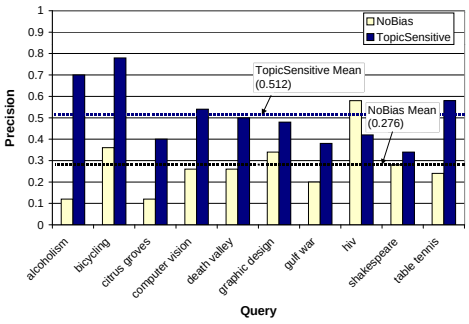

Figure 1: Précision quantifiée par des utilisateurs pour les 10 premiers résultats retournés par des requêtes-tests

5.2 Effet du PageRank thématique

Grâce aux mesures de similarités définies, on cherche à évaluer l'effet de l'utilisation d'un *PageRank* thématique sur des requêtes de test.

5.2.1 Choix du coefficient α

Le choix du coefficient α de prise en compte du vecteur de pré-réputation pour le calcul du *PageRank* influe sur le calcul du score de réputation thématique : en analysant les similarités *OSim* et *KSim* obtenues entre deux classements pour des requêtes test, l'un avec l'usage de *PageRank* avec $\alpha = 0,05$ et l'autre avec $\alpha = 0,25$, on constate que $(OSim; KSim) = (0,72; 0,64)$ pour le *PageRank* classique (avec notes de pré-réputations égalitaires pour toutes les pages de l'ODP). La similarité est généralement moins importante pour les *PageRank* avec biais thématique mains néanmoins assez variable : ainsi $(OSim; KSim) = (0,57; 0,50)$ pour la catégorie *Society* alors que pour la catégorie *Games*, $(OSim; KSim) = (0,78; 0,67)$. Nous pouvons donc en déduire que l'influence du choix du coefficient α est variable selon les catégories : il pourrait être judicieux d'opérer un choix de coefficient α spécifique pour chaque catégorie : il reste néanmoins à approfondir cette idée déjà abordée par Chakrabarti et al. [2]. Pour la suite des tests, la valeur $\alpha = 0,25$ est retenue.

5.2.2 Catégorisation liée à la requête

Une catégorisation bayesienne de la requête est d'abord envisagée sans tenir compte du contexte (comme décrit en 4.1) : il s'agit donc de calculer la probabilité d'appartenance de chacun des termes de la requête à chacune des catégories définie et à définir un score de réputation personnalisé issue de la moyenne pondérée des scores *PageRank* thématiques.

Afin de mesurer les bénéficies de l'introduction de scores catégorisés par les termes de la requêtes, on demande à un groupe d'utilisateurs de spécifier la qualité des 10 premiers résultats retournés par des requêtes tests avec ou sans l'utilisation de scores catégorisés : ils doivent spécifier pour chacune des pages listées si elle est ou non pertinente, puis indiquer leur préférence, en aveugle, pour l'une ou l'autre des techniques de classements. Les résultats de cette étude sont résumés sur la figure 5.2.2 : la précision des résultats retournés est presque toujours supérieure avec l'usage de scores catégorisés. La précision moyenne avec *PageRank* classique est de 0,276 (2,76 pages sur 10) tandis qu'elle atteint 0,512 avec des scores catégorisés.

5.2.3 Catégorisation liée au contexte

Plutôt que de considérer la catégorisation des scores par rapport à la seule requête, on considère le contexte. L'auteur de l'article étudié prend pour exemple le mot *blues* dans deux contextes différents : un contexte médical (mélancolie) et un contexte artistique (genre musical). La prise en compte du contexte permet d'attribuer des poids différents aux scores thématiques des catégories *Health* et *Arts* et ainsi de lever la polysémie. On pourrait également reprendre l'exemple du terme *Java* cité précédemment.

6 Optimisation

6.1 Calcul du PageRank thématique

6.1.1 Algorithme classique

Afin de calculer le *PageRank* thématique des pages indexées, plusieurs méthodes peuvent être mises en oeuvre. La plus évidente consiste à utiliser l'algorithme traditionnel de calcul conduisant à l'obtention du vecteur propre de la matrice des notes locales après plusieurs itérations. Le calcul doit être répété pour le calcul du *PageRank* des pages pour chaque thème. Haveliwala expérimente ce calcul avec une machine disposant d'un processeur AMD Athlon 1533 MHz et 2,5 Go de mémoire centrale en environ 5 heures avec 25 itérations, sur un jeu de test de 360 millions de pages [9] nécessitant 4,3 Go de stockage.

6.1.2 Extrapolation quadratique

Une valeur approchée du *PageRank* peut être calculée par la méthode d'extrapolation quadratique [4] qui consiste à approximer et soustraire les vecteurs propres secondaires de l'itération courante. Cette méthode permet d'accélérer le calcul de 25 à 300% en fonction du coefficient α de prise en compte de la pré-réputation.

6.1.3 Programmation dynamique

La méthode naïve de calcul des *PageRank* thématiques nécessite un calcul indépendant des *PageRank* pour chacune des catégories. Ne serait-il pas possible de factoriser des étapes de calcul pour plusieurs *PageRank* thématiques ? Jeh et Widon proposent une méthode [5] par programmation dynamique permettant de calculer des vecteurs partiels de réputation pouvant être ensuite utilisées pour former des vues personnalisées de scores de réputation calculables lors de chaque requête.

6.2 Codage efficace des scores PageRank

6.2.1 Nécessité d'un codage efficace des scores de réputation

Un moteur de recherche utilisant des scores de réputation tels que le *PageRank* nécessite d'accéder à deux bases d'informations distinctes afin de procéder à la sélection des documents pertinents et à leur classement :

1. *Un index inverse associant pour chaque mot la liste de ces occurrences dans les pages Web indexées.* Un tel index est extrêmement volumineux puisqu'il est nécessaire de stocker l'ensemble des occurrences d'un mot[10]. Lorsqu'une recherche est menée sur les termes $Q_1, \ldots, Q_n$, le moteur recherche dans l'index inverse l'ensemble des documents contenant chacun des termes puis en réalise l'intersection (recherche ET) : cela nécessite $O(n)$ accès disques. Toutefois on notera que les requêtes comportent peu de mots en règle générale.

[9] Pour les 23 premières itérations, deux niveaux de pages non-liantes sont supprimées conduisant à l'obtention d'un sous-graphe de 80 millions de pages.

[10] Si l'on souhaite avoir un ordre de grandeur de la taille d'un tel index, posons pour hypothèse la présence de 10 milliards de pages indexées, chacune comportant en moyenne 1000 mots avec 500 mots distincts. Si un identificateur de document est stocké sur 5 octets et une position sur 2 octets, alors l'espace nécessaire à 45 To.

2. *Une table associant à chaque page son score de réputation (voire son vecteur de réputation dans le cas de* PageRank *thématique).* Les documents sélectionnés, il est nécessaire de leur attribuer une note de pertinence. Celle-ci est attribuée suivant plusieurs composantes : une composante liée uniquement aux termes recherchés et à leur localisation dans les pages sélectionnées (le score de pertinence est, par exemple, plus important si les termes recherchés sont localement proches dans la page ou sont mis en valeur — titre —). L'autre composante utilise le score de réputation (*PageRank* pour Google) de la page : il est alors nécessaire de récupérer ce score pour chacune des m pages de la sélection : si la table de scores est stockée sur mémoire de masse, cela nécessite m accès disque ; l'étape de récupération de scores s'avère alors le facteur limitant la vitesse de réponse à la requête. Il est donc nécessaire de conserver la table des scores en mémoire centrale pour un accès rapide, d'où la problématique du codage efficace des scores de réputation, d'autant plus inévitable lorsque l'on manipule de multiples scores thématiques.

Estimation de la taille de la table de scores La taille $t(m,k)$ de la table de scores peut être ainsi calculée pour m documents et k catégories :

$$t(m,k) = m(taille(doc\_id) + ktaille(score))$$

soit pour 10 milliards de pages, 16 catégories et $taille(doc\_id) = 5$, $taille(score) = 4$, 690 Go. Un tel volume de données rend difficile la conservation en mémoire centrale de la table : il est donc nécessaire d'envisager des techniques spécifiques de codage. Une autre alternative serait de joindre à chaque identificateur de document, dans l'index inverse, son vecteur de scores, pour éviter une indirection dans la table de scores. Une telle méthode entraînerait cependant une augmentation très importante de la mémoire de masse nécessaire à l'index inverse.

6.2.2 Codage compressif

On choisit de réaliser un codage destructif pour les scores de réputation : on se contente d'une approximation de score. L'idée est de réaliser un découpage de l'espace des valeurs pouvant être prises par le score de réputation en plusieurs intervalles et de n'indiquer que l'identifiant de l'intervalle dans lequel se trouve le score. Si l'espace est découpé en n intervalles, $\lceil \log_2(n) \rceil$ bits sont nécessaires pour coder chaque score. Mais comment déterminer les tailles des intervalles ? Diverses stratégies de découpage de l'espace des valeurs des scores[11] sont envisageables en utilisant des fonctions de compression. Si f est la fonction de compression utilisée, on définit alors les n intervalles de découpage de l'espace ainsi : $[0, f^{-1}(1/n)[, [0, f^{-1}(2/n)[, ..., [0, f^{-1}(1)]$.

Mesure quantitative sur un découpage Après utilisation d'une méthode de compression de la valeur de score de réputation par découpage en intervalles, plusieurs pages de scores différents peuvent être représentés par le même identificateur d'intervalle : une telle destruction d'information sur le score peut-elle conduire à un classement sensiblement différent des pages retournés pour une requête ? Afin de mesurer la qualité d'un découpage, il est possible de calculer une mesure de distorsion entre le classement engendré par l'utilisation d'un score non-compressé – séquence de pages τ_1 – (représenté par un flottant de 32 bits) et le classement obtenu par l'utilisation du score compressé – séquence de pages τ_2 – (représenté par l'indication de son intervalle d'appartenance). On utilise pour cela la distance *KDist* définie par $KDist(\tau_1, \tau_2) = 1 - KSim(\tau_1, \tau_2)$ où *KSim* est la mesure de similarité définie précédemment quantifiant la variabilité sur le classement des résultats. On cherchera alors à améliorer l'efficacité d'un découpage compressif en diminuant la distorsion $KDist(\tau_1, \tau_2)$.

Quelques stratégies de découpages On propose alors différentes stratégies de découpage. Pour chacune de ces stratégies, la distorsion moyenne sur le classement des résultats est calculée

[11] On considère les valeurs des scores de réputation normalisées entre 0 et 1.

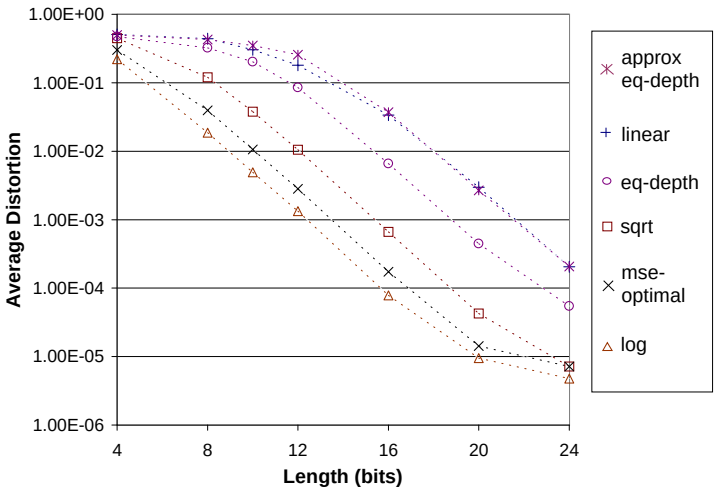

Figure 2: Distorsion moyenne $KDist(\tau_1, \tau_2)$ pour des stratégies de découpage : comparaison des classements sur les 100 premiers résultats. Score de pertinence : $s_i \cos(Q, d_i)$

par Haveliwla pour 86 requêtes de test : le score *pi* utilisé pour la quantification de pertinence de la pages i pour la requête Q est $p_i = s_i \cos(Q, d_i)$ (avec d_i le vecteur de mots du document i et s_i son score de réputation, soit exact, soit récupéré après compression – le score est approximé par le centre de l'intervalle le définissant –). On procède alors au calcul de distorsion moyenne pour plusieurs cardinaux d'intervalles (voir figure 6.2.2).

- *Découpage linéaire (*linear*).* Pour ce découpage (fonction de compression utilisée : $x \to x$), la taille de chaque intervalle est constante : pour un espace découpé en n intervalles (cellules), chaque cellule possède une taille de $1/n$. La distorsion moyenne des classement est particulièrement élevée : 24 bits (2^24 intervalles) sont nécessaires afin qu'elle soit inférieure à 1%.

- *Découpage racine carrée (*sqrt*).* On utilise la fonction de compression $x \to \alpha sqrt(x)$ pour le découpage des intervalles.

- *Découpage logarithmique (*log*).* On utilise la fonction de compression $x \to \alpha \log x$. Intuitivement considérer une échelle logarithmique pour la distribution des scores de réputation parait intéressant dans la mesure, où par loi de puissance, il existe peu de sites possédant un score élevé et un grand nombre de sites avec un score faible : il est donc intéressant de discriminer les scores sur des petites valeurs plutôt que sur les grandes[12] L'intuition est ici confirmée par l'expérience : on constate que l'échelle logarithmique permet d'obtenir la distorsion moyenne la plus faible (inférieure à 1% avec 8 bits).

- *Découpage $x \to \alpha x^{-2,17}$ (*mse_optimal*).* Cette fonction de compression minimise le carré de la distorsion moyenne.

- *Fonction avec approximation moyenne d'intervalles d'effectifs égaux.* Cette fonction approxime des intervalles avec effectifs égaux.

[12]On peut d'ailleurs noter que Google communique les scores *PageRank* des pages indexées au public sous la forme d'une échelle logarithmique s'étalant de 0 à 10 : il n'est donc pas improbable que les *PageRank* soit conservés en interne avec l'usage d'une échelle logarithmique.

- *Partition en intervalles d'effectifs égaux.* On choisit ici des intervalles afin que chacun d'entre-eux comporte un nombre égal de scores de réputation de pages. L'efficacité d'une telle méthode, bien que supérieure à la fonction d'approximation précédente permet d'obtenir une distorsion d'environ 10% pour 8 bits (256 intervalles).

Utilisation du découpage logarithmique Au vu des résultats expérimentaux, il est préférable d'utiliser un découpage logarithmique : l'usage de 8 bits pour le codage du score est amplement suffisant (distorsion inférieure à 1%), l'utilisation de 4 bits pouvant même être envisagé (distortion inférieure à 10%). En partant de nos hypothèses formulées en 6.2.1, 10 milliards de pages nécessitent une table de 210 Go avec un score logarithmique sur 8 bits et 130 Go sur 4 bits.

7 Conclusion

L'utilisation de *PageRank* thématique offre d'intéressantes opportunités pour la personnalisation de recherches contextuelles. Ainsi le moteur de recherche Google a récemment mis en place un système de personnalisation de recherche : sa première version permettait d'attribuer manuellement des poids à chacune des catégories alors que la version actuelle utilise un compte Google (obtenu par enregistrement sur un des services Google tel que GMail) pour stocker les liens cliqués sur les pages de résultats et l'historique des recherches réalisées. Il est probable que Google utilise un système de *PageRank* thématique pour réaliser cette personnalisation.

D'autre part, certaines voies concernant le *PageRank* méritent d'être approfondies. Ainsi la démarche naïve du calcul individuel des *PageRank* thématique demande à être améliorée afin de pouvoir réaliser des calculs de *PageRank* sur des catégories de granularité plus fine avec une complexité plus avantageuse.

References

[1] Open directory project. URL `http://dmoz.org/`.

[2] S. Chakrabarti, M. M. Joshi, K. Punera, and D. M. Pennock. The structure of broad topics on the web. In *International World Wide Web Conference*, 2002. URL `http://www.lans.ece.utexas.edu/~kunal/papers/broadtopics.pdf`.

[3] Taher H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the Eleventh International World Wide Web Conference*, Honolulu, Hawaii, May 2002. URL `http://citeseer.ist.psu.edu/haveliwala02topicsensitive.html`.

[4] Taher H. Haveliwala, Christopher D. Manning, Gene H. Golub, and Sepandar D. Kamvar. Extrapolation methods for accelerating pagerank computations. Technical report, Stanford University, 2003. URL `http://www-db.stanford.edu/~taherh/papers/extrapolation.ps`.

[5] G. Jeh and J. Widom. Scaling personalized web search, 2002. URL `http://citeseer.ist.psu.edu/jeh02scaling.html`.

[6] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge University Press, 1995. ISBN 0521474655.

[7] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998. URL `http://citeseer.ist.psu.edu/page98pagerank.html`.