2
Modelo de Hubs e Authorities

Jon Kleinberg [1,2] propôs um modelo baseado em *hyperlinks* que permitisse a inferência de autoridade e um conjunto de algoritmos que identificasse páginas relevantes para tópicos de busca de caráter geral. Esse modelo é baseado na relação entre páginas que são autoridades sobre um tópico e páginas que interligam essas autoridades (*hubs*). *Jon Kleinberg* observou um equilíbrio natural entre autoridades e *hubs* num grafo definido pela estrutura de *hyperlinks* e desenvolveu um algoritmo, conhecido como *HITS* (*Hyperlink Induced Topic Search*), para identificar, simultaneamente, esses tipos de páginas. O algoritmo opera em um sub-grafo focado da *WWW*, construído a partir do resultado de uma máquina de busca baseada somente em texto.

Apresentamos, a seguir, uma descrição mais detalhada do algoritmo *HITS*.

2.1.
Algoritmo *HITS*

A partir de uma consulta de tópico geral especificada pela cadeia σ, é necessário, para se analisar a estrutura de *hyperlinks* e se extrair as páginas que são autoridades, definir qual sub-grafo da *WWW* o algoritmo vai utilizar. Tal sub-grafo (S_σ) deve conjugar três características, quais sejam:

1. Relativamente pequeno;
2. Rico em páginas relevantes;
3. Possuir a maior parte das grandes autoridades;

Para satisfazer às características 1 e 2 supracitadas, basta coletar as t primeiras páginas classificadas por uma máquina de busca baseada somente em texto – empiricamente, são escolhidas em torno de duzentas páginas. Esse grafo inicial (R_σ) geralmente está longe de satisfazer a terceira condição, pois é notório que páginas que são autoridades normalmente não possuem em seu

texto muitas repetições da palavra da consulta. Um exemplo simples consiste na procura de páginas que tratam sobre "*Harvard*". A página www.harvard.edu, considerada a maior autoridade sobre o assunto, não possui em seu texto repetidas vezes a palavra "*Harvard*".

Entretanto, R_σ pode ser usado como base para construir o grafo S_σ desejado. Como antes aludido, as páginas autoridades podem não pertencer a R_σ. Porém, considerando que essas podem ser apontadas por pelo menos uma página de R_σ, é razoável inferir que expandindo o conjunto R_σ através dos *hyperlinks* das páginas pertencentes a R_σ, o grafo resultante passe a satisfazer a condição 3 e, por conseqüência, gere S_σ(figura 1).

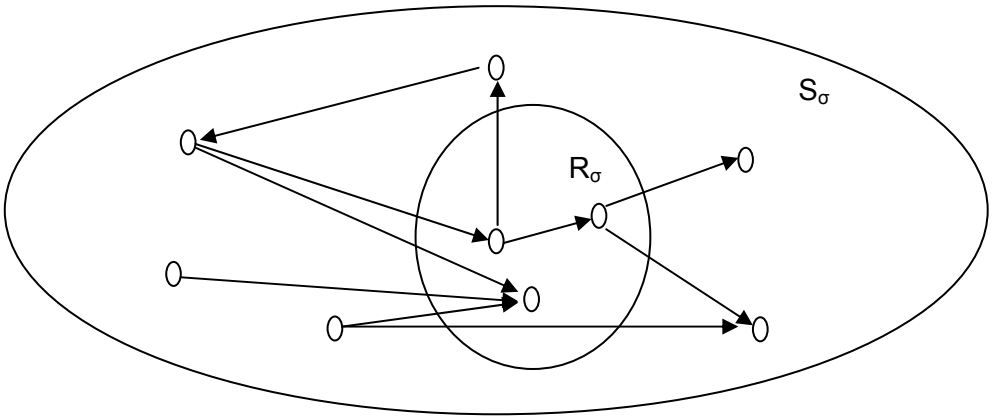

Figura 1: Expansão de R_σ.

A expansão de R_σ consiste em acrescer as páginas que são apontadas por cada página pertencente a R_σ, e também um número d de páginas que apontam para as páginas de R_σ. Os *hyperlinks* de navegação são desconsiderados, formando um grafo orientado apenas com arestas interdomínios. Definindo o sub-grafo com G=(V,E), esse é constituído de uma coleção V de páginas conectadas por seus *hyperlinks*, em que os vértices são as páginas e as arestas orientadas (p,q) Є E são *hyperlinks* de p para q.

A seguir, apresentamos o pseudocódigo da rotina de expansão do sub-grafo:

```
Subgraph(σ,ε,t,d)
σ: palavra da consulta
ε: máquina de busca baseada em texto
t: tamanho do núcleo inicial
```

d: número de páginas que apontam para as páginas do núcleo
Seja R_σ o conjunto das t mais significantes páginas retornadas por ε com σ
$S_\sigma := R_\varsigma$
Para cada página Є R_σ
  Seja A(p) o conjunto de todas as páginas que p aponta
  Seja A^- (p) o conjunto de todas as páginas que apontam para p
  Adicione todas as páginas de A(p) em S_σ
  Se $|A^-$ (p) < d| então
        Adicione todas as páginas de A^- (p) em Sσ
  senão
        Adicione um conjunto arbitrário d de páginas de A^- (p) em S_σ
Retorne S_σ

Após a construção do sub-grafo, o problema converge para a extração e classificação das autoridades existentes, considerando apenas a estrutura de *hyperlinks* existente. Analisando a estrutura, as autoridades relevantes presentes possuem não só um alto grau de *hyperlinks* que as apontam, como também possuem grupos de páginas que as apontam em comum. Essas páginas são denominadas *hubs* e são responsáveis por vincular as autoridades comuns excluindo as páginas que possuem um alto grau de *hyperlinks* de chegada e não são relevantes para o assunto (figura 2).

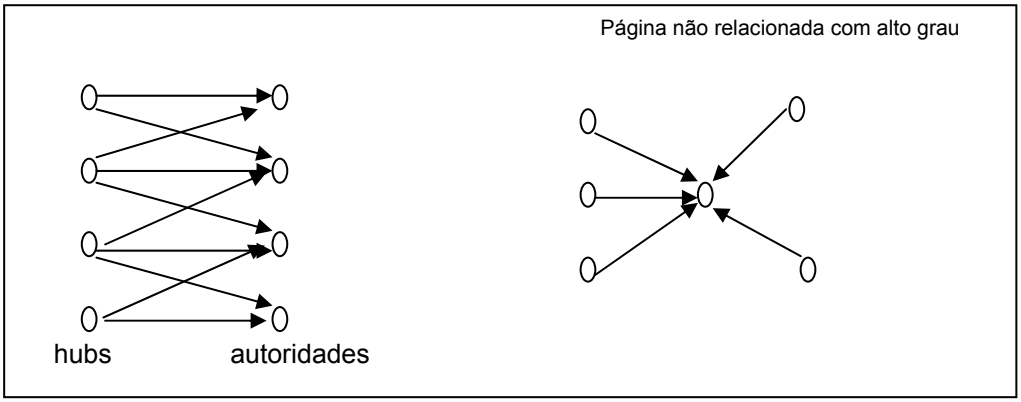

Figura 2: *Hubs* e autoridades.

Sendo assim, autoridades e *hubs* exibem uma relação de interdependência: uma boa autoridade será uma página apontada por bons *hubs* e um bom *hub* será uma página que aponta para boas autoridades.

Com base nessa relação, um algoritmo iterativo foi desenvolvido permitindo encontrar os *hubs* e as autoridades. Cada página i possui dois pesos não negativos associados, um para autoridade ($a^{<i>}$) e um para *hub* ($h^{<i>}$). Desta

forma, é natural reescrever a relação de interdependência como: se uma página p aponta para páginas com altos valores de a, então ela deve receber um valor alto para h, e se p é apontada por páginas com valores altos de h, essa deve receber um valor alto para a.

Em face do exposto, dois operadores foram definidos para atualizar os pesos de autoridade (I) e os de *hubs* (O):

- I : $a^{<p>} \leftarrow \sum_{q:(q,p)\in E} h^{<q>}$;
- O : $h^{<p>} \leftarrow \sum_{q:(p,q)\in E} a^{<q>}$.

Tais operadores representam claramente a relação de interdependência entre *hubs* e autoridades.

Para encontrar o valor de equilíbrio entre os pesos, são aplicados, alternadamente, os operadores I e O até que a estabilidade seja alcançada[1,2], ou seja, que os valores de a e h das páginas se tornem inalterados com a iteração do algoritmo.

A seguir, apresentamos o pseudocódigo do algoritmo de iteração:

```
Iterate(G,k)
( G : coleção de n páginas com hyperlinks )
( k: número de iterações do algoritmo )
Seja z o vetor (1,1,1...1) Є R^n
a_0:=z
h_0:=z
Para i := 1,2..k
  Aplicar I em (a_{i-1},h_{i-1}), obtendo a`_i
  Aplicar O em (a`_i,h_{i-1}), obtendo h`_i
  Normalizar a`_i, obtendo a_k
  Normalizar h`_i, obtendo h_k
Retornar (a_k,h_k)
```

Por fim, basta ordenar as coordenadas do vetor a, retornadas pelo algoritmo *iterate*, para expor as páginas de maior autoridade. O mesmo se aplica ao vetor h para as páginas de maior *hub*.