PageRank

Wendel Melo
Faculdade de Computação
Universidade Federal de Uberlândia

Recuperação da Informação

PageRank

- Algoritmo criado para ser a espinha dorsal do sistema de buscas do Google (L. Page, S. Brin, R. Motwani, T. Winograd);
- O *PageRank* procura medir a importância de cada página na Web;
- Essa importância é calculada com base na probabilidade de um usuário comum encontrar uma determinada página na Web;
- O algoritmo parte da premissa de que, quanto mais uma página é linkada por outras, maior a probabilidade de que um usuário navegando ao acaso a encontre (e, assim, maior a sua importância);

PageRank

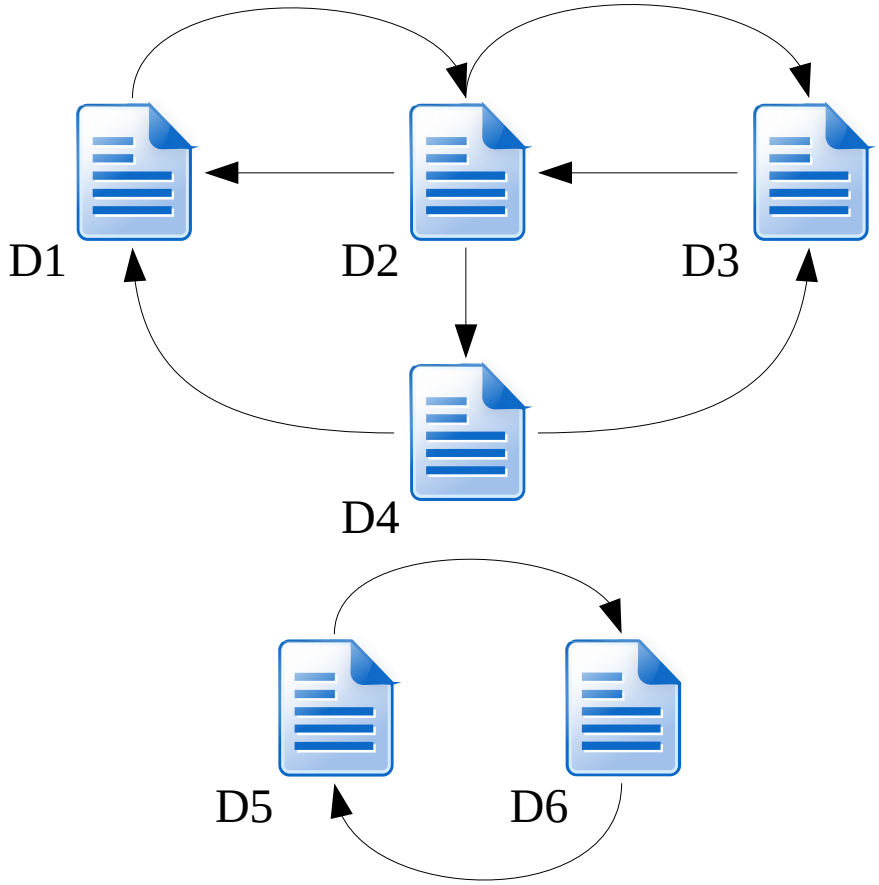
- O algoritmo também se baseia no fato de que a importância da página deve ser maior se ela é apontada por outras páginas que também sejam consideradas importantes.

- Assim, o *PageRank* de uma página *p* é definido de forma **recursiva**, isto é, a partir dos *PageRanks* das páginas que apontam para *p*.

- Importante para evitar trapaças com a métrica de PageRank!

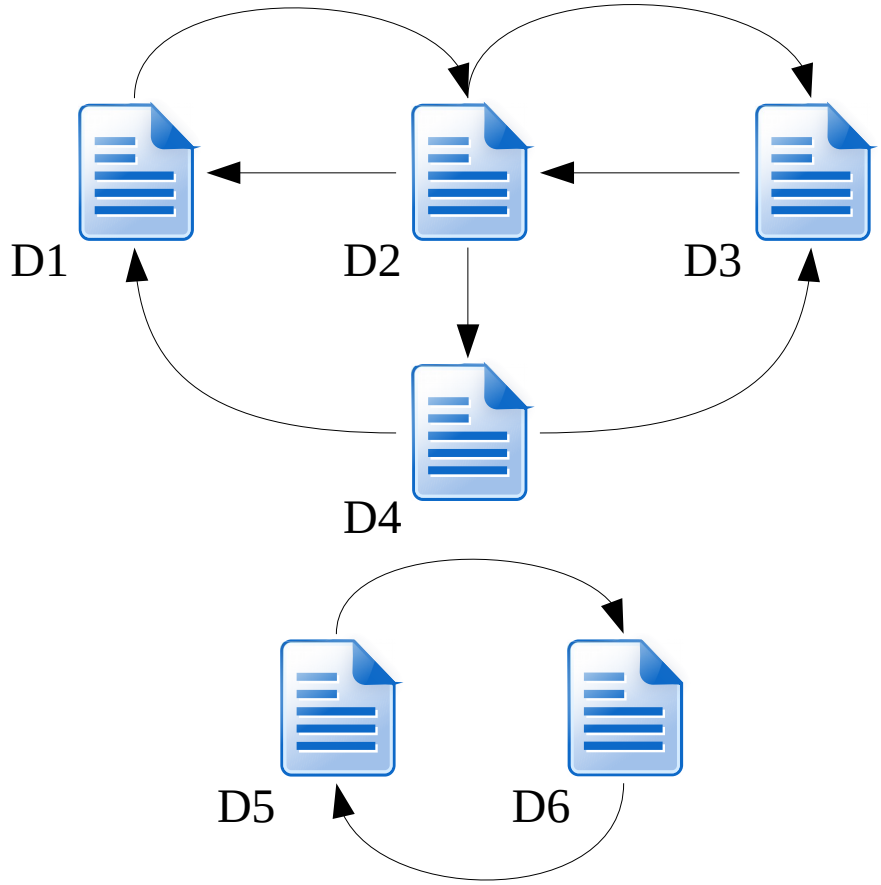
PageRank

- O *PageRank* usa então como base uma cadeia de Markov construída a partir do grafo de documentos da internet;



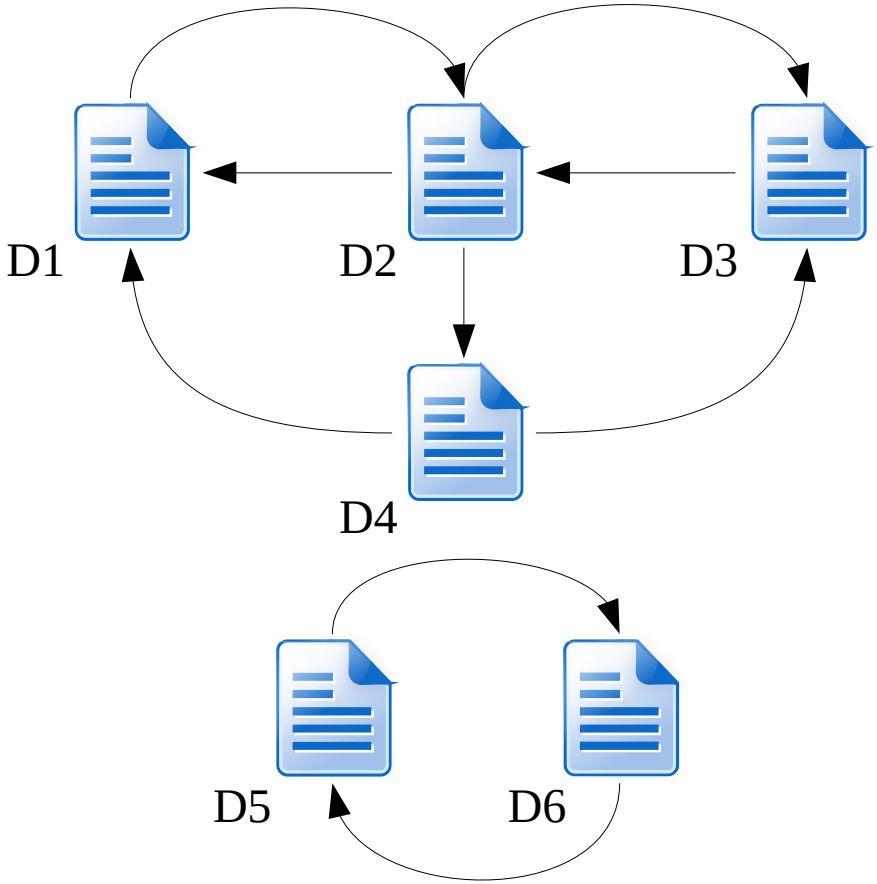
PageRank

- O *PageRank* usa então como base uma cadeia de Markov construída a partir do grafo de documentos da internet;
- Considere o grafo de documentos ao lado, onde as arestas representam links entre os documentos;



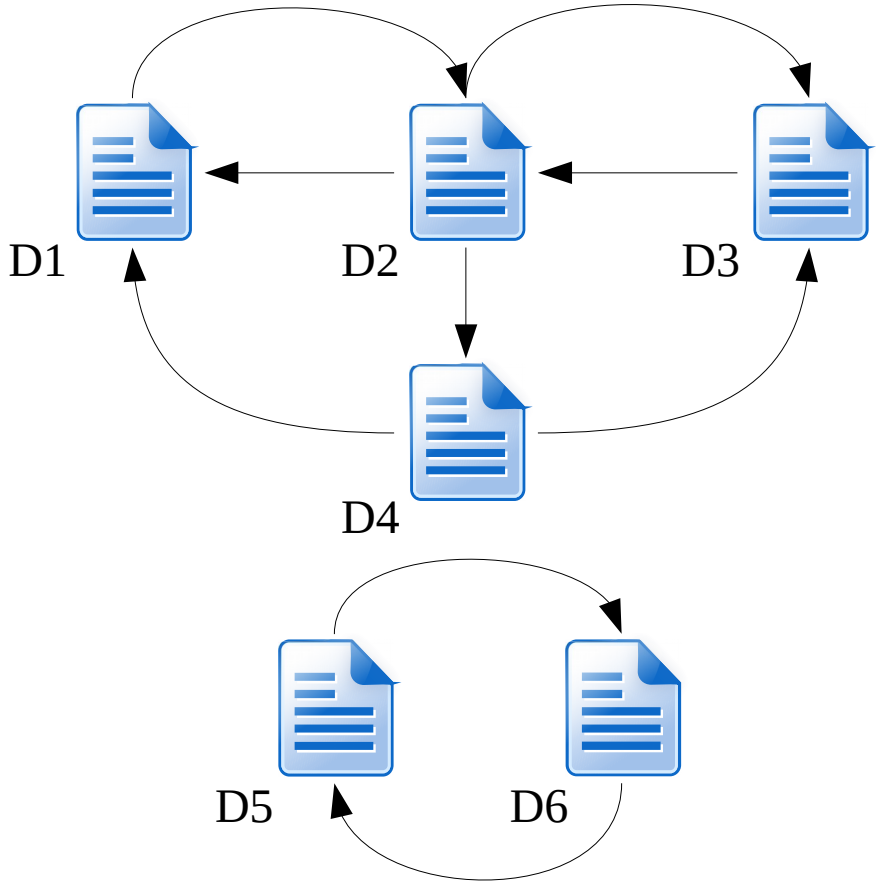
PageRank - Premissas

- O usuário começa sua navegação a partir de uma página aleatória;
- Em princípio, o usuário navega através dos links de uma página para outra;
- Se o usuário estiver visitando uma página *p*, cada página linkada por *p* tem igual chance de ser a próxima acessada;



PageRank - Premissas

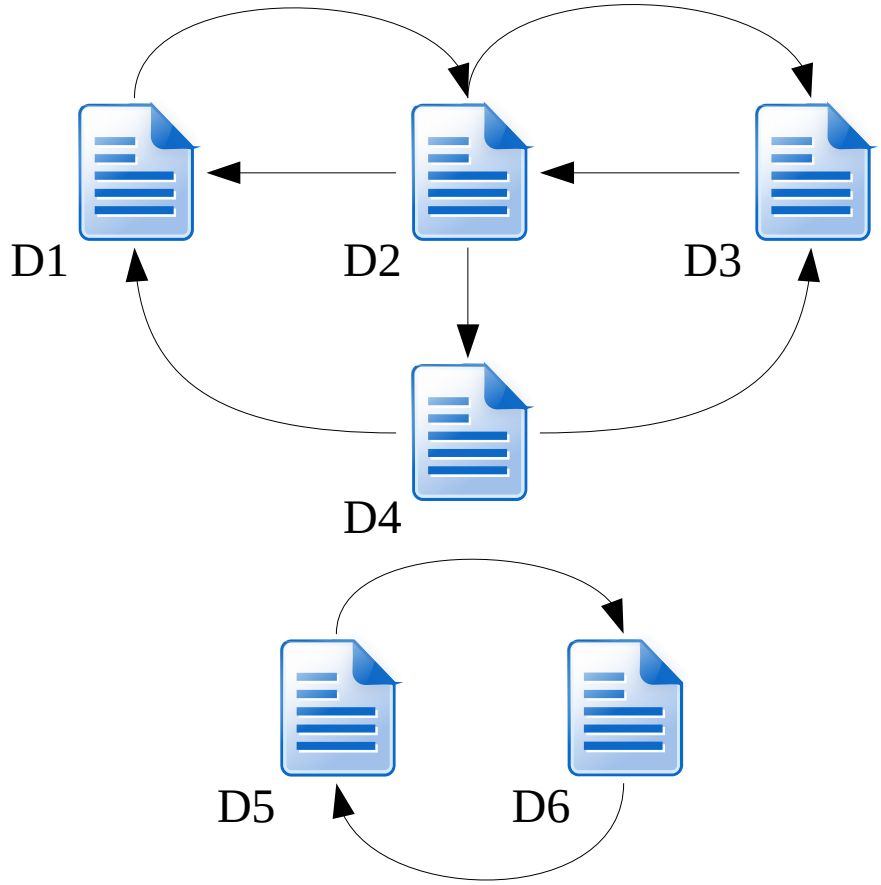
- Por exemplo, suponha que o usuário esteja navegando no documento D4;
- A partir daí, haveria, em princípio, 50% de chance do usuário em seguida visitar D1 e 50% de chance do usuário em seguida visitar D3;



PageRank - Premissas

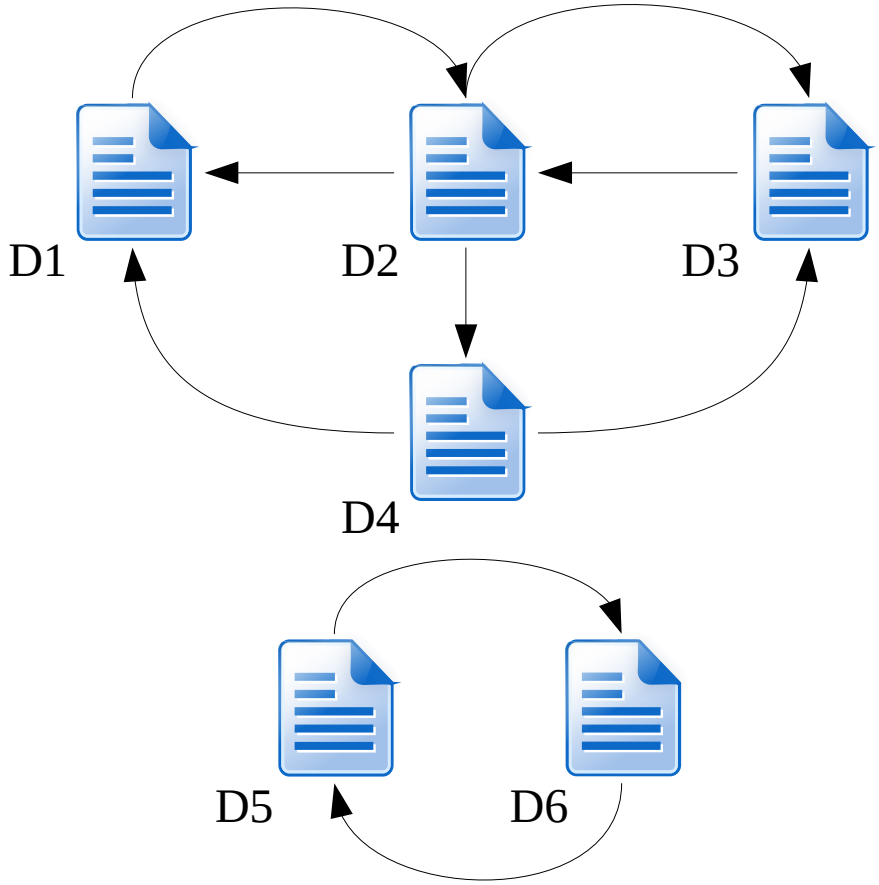
- Pode-se então partir de um doc inicial aleatório e realizar um nº X de transições aleatórias de docs, respeitando o grafo de ligações;

- Feito isso, basta contabilizar o número de vezes que cada doc foi visitado e dividir por X para ter a probabilidade de acesso de cada documento;



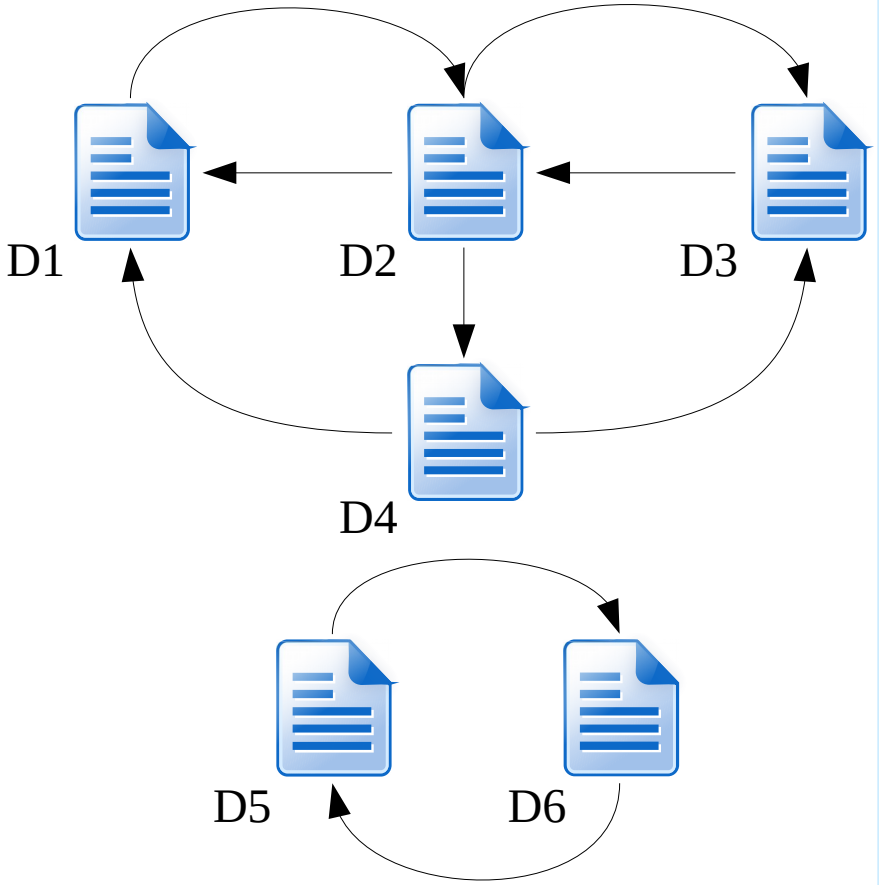
PageRank - Premissas

- O problema dessa abordagem é que, se o processo começar em D5, por exemplo, apenas D5 e D6 serão acessados, fazendo com os *PageRanks* dos demais docs acabem valendo zero;



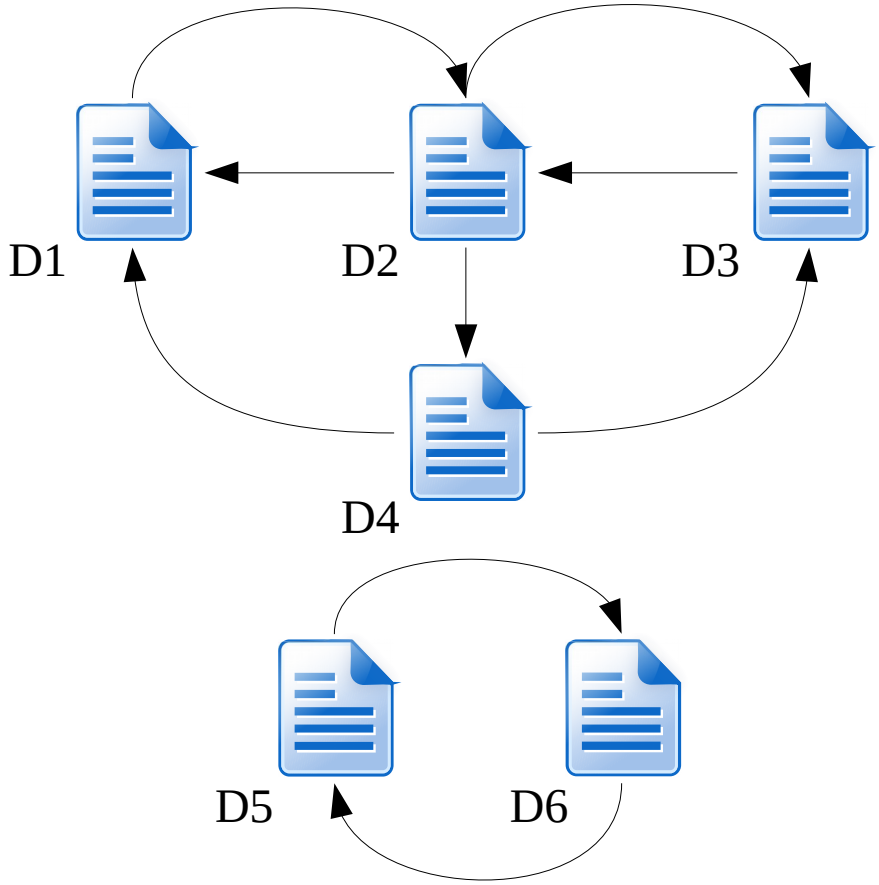
PageRank - Premissas

- Sendo assim, inserimos um fator *d (damping factor)* que controla a probabilidade do usuário seguir um dos links presentes no documento corrente;
- Desse modo, definimos como (*1–d*) a probabilidade do usuário saltar para qualquer documento do grafo após visualizar o documento corrente;



PageRank - Premissas

- Sendo assim, seja *PR(j)* o *PageRank* do documento *j*;
- *PR(j)* é calculado como a probabilidade de, em um dado momento, o usuário vir a acessar o documento *j*;
- Temos então dois modos de calcular *PR(j)*:



PageRank – Método da amostragem

- O primeiro modo (**método da amostragem**) é partir de um doc inicial e realizar X transições aleatórias, considerando que:

  1) Com probabilidade *d* o usuário segue algum link do doc corrente

  2) Com probabilidade (*1-d*) o usuário pode saltar a navegação para qualquer documento;

- Assim, *PR(j)* será a_j/X, onde a_j é o número de vezes em que o documento *j* foi acessado e X o número de transições.

PageRank - Cálculo

- O segundo modo é definir um valor inicial *PR(j)* para cada documento *j*, *PR(j) = 1/N* :

- A partir daí, atualizamos iterativamente cada *PR(j)*. A cada iteração, recalculamos o *PageRank* de todos os documentos j:

$$PR(j) := \frac{1-d}{N} + d \sum_{i \in L(j)} \frac{PR(i)}{NumLinks(i)}$$

- Onde *N* é o nº total de docs, *NumLinks(i)* é o nº de links na página *i*, e *L(j)* é o conjunto de docs que contém links para o doc *j*.

PageRank - Cálculo

- A partir daí, atualizamos iterativamente cada *PR(j)*. A cada iteração, recalculamos o *PageRank* de todos os documentos j:

$$PR(j) := \frac{1-d}{N} + d \sum_{i \in L(j)} \frac{PR(i)}{NumLinks(i)}$$

- Onde *N* é o nº total de docs, *NumLinks(i)* é o nº de links na página *i*, e *L(j)* é o conjunto de docs que contém links para o doc *j;*

PageRank - Cálculo

- A partir daí, atualizamos iterativamente cada *PR(j)*. A cada iteração, recalculamos o *PageRank* de todos os documentos j:

$$PR(j) := \underbrace{\frac{1-d}{N}} + d \sum_{i \in L(j)} \frac{PR(i)}{NumLinks(i)}$$

Probabilidade do doc j ser acessado por meio do salto aleatório para um doc qualquer da base

- Onde *N* é o nº total de docs, *NumLinks(i)* é o nº de links na página *i*, e *L(j)* é o conjunto de docs que contém links para o doc *j;*

PageRank - Cálculo

- A partir daí, atualizamos iterativamente cada *PR(j)*. A cada iteração, recalculamos o *PageRank* de todos os documentos j:

$$PR(j) := \underbrace{\frac{1-d}{N}}_{\text{Probabilidade do doc j ser acessado por meio do salto aleatório para um doc qualquer da base}} + \underbrace{d \sum_{i \in L(j)} \frac{PR(i)}{NumLinks(i)}}_{\text{Probabilidade do doc j ser acessado por meio do no link presente no doc i que aponta para j.}}$$

- Onde *N* é o nº total de docs, *NumLinks(i)* é o nº de links na página *i*, e *L(j)* é o conjunto de docs que contém links para o doc *j;*

PageRank - Cálculo

- A partir daí, atualizamos iterativamente cada *PR(j)*. A cada iteração, recalculamos o *PageRank* de todos os documentos j:

$$PR(j) := \frac{1-d}{N} + d \sum_{i \in L(j)} \frac{PR(i)}{NumLinks(i)}$$

- Onde *N* é o nº total de docs, *NumLinks(i)* é o nº de links na página *i,* e *L(j)* é o conjunto de docs que contém links para o doc *j;*

- Ao final de cada iteração, é preciso dividir cada PR(j) pelo somatório de todos os PR(j) para garantir que a soma dos PageRanks de todas as páginas seja igual 1.

PageRank - Cálculo

- O processo é repetido até que os valores de *PR(j)* convirjam, isto é, entre duas iterações consecutivas não haja mudança significativa em algum *PR(j)*.

PageRank – Algoritmo Iterativo

1) Para *j := 1, 2, ..., N*:
2) $PR(j) := \frac{1}{N}$;
3) Repita
   {
4) Para *j := 1, 2, ..., N*:
5) $\bar{PR}(j) := PR(j)$;
6) Para *j := 1, 2, ..., N*:
7) $PR(j) := \frac{1-d}{N} + d \sum_{i \in L(j)} \frac{\bar{PR}(i)}{NumLinks(i)}$;
8) Para *j := 1, 2, ..., N*:
9) $PR(j) := \frac{PR(j)}{\sum_{i=1}^{N} PR(i)}$;
10) } enquanto existir *j* tal que $|\bar{PR}(j) - PR(j)| > \epsilon$

PageRank – Algoritmo Iterativo

1) Para *j := 1, 2, ..., N*:
2) $PR(j) := \frac{1}{N}$;

Inicializa o *PageRank* para todos os docs

3) Repita
   {
4) Para *j := 1, 2, ..., N*:
5) $\bar{PR}(j) := PR(j)$;
6) Para *j := 1, 2, ..., N*:
7) $PR(j) := \frac{1-d}{N} + d \sum_{i \in L(j)} \frac{\bar{PR}(i)}{NumLinks(i)}$;
8) Para *j := 1, 2, ..., N*:
9) $PR(j) := \frac{PR(j)}{\sum_{i=1}^{N} PR(i)}$;
10) } enquanto existir *j* tal que $|\bar{PR}(j) - PR(j)| > \epsilon$

PageRank – Algoritmo Iterativo

1) Para *j := 1, 2, ..., N*:
2) $PR(j) := \frac{1}{N}$;

Inicializa o *PageRank* para todos os docs

3) Repita
   {
4) Para *j := 1, 2, ..., N*:
5) $\bar{PR}(j) := PR(j)$;

Copia o valor atual de PR para $\overline{\text{PR}}$ para todos os docs

6) Para *j := 1, 2, ..., N*:
7) $PR(j) := \frac{1-d}{N} + d \sum_{i \in L(j)} \frac{\bar{PR}(i)}{NumLinks(i)}$;
8) Para *j := 1, 2, ..., N*:
9) $PR(j) := \frac{PR(j)}{\sum_{i=1}^{N} PR(i)}$;
10) } enquanto existir *j* tal que $|\bar{PR}(j) - PR(j)| > \epsilon$

PageRank – Algoritmo Iterativo

1) Para *j := 1, 2, ..., N*:
2) $PR(j) := \frac{1}{N}$;

Inicializa o *PageRank* para todos os docs

3) Repita
   {
4) Para *j := 1, 2, ..., N*:
5) $\bar{PR}(j) := PR(j)$;

Copia o valor atual de PR para $\overline{\text{PR}}$ para todos os docs

6) Para *j := 1, 2, ..., N*:
7) $PR(j) := \frac{1-d}{N} + d \sum_{i \in L(j)} \frac{\bar{PR}(i)}{NumLinks(i)}$;

Atualiza o valor do PR de cada documento

8) Para *j := 1, 2, ..., N*:
9) $PR(j) := \frac{PR(j)}{\sum_{i=1}^{N} PR(i)}$;

10) } enquanto existir *j* tal que $|\bar{PR}(j) - PR(j)| > \epsilon$

PageRank – Algoritmo Iterativo

1) Para *j := 1, 2, ..., N*:
2) $PR(j) := \frac{1}{N}$;

Inicializa o *PageRank* para todos os docs

3) Repita
   {
4) Para *j := 1, 2, ..., N*:
5) $\bar{PR}(j) := PR(j)$;

Copia o valor atual de PR para $\overline{\text{PR}}$ para todos os docs

6) Para *j := 1, 2, ..., N*:
7) $PR(j) := \frac{1-d}{N} + d \sum_{i \in L(j)} \frac{\bar{PR}(i)}{NumLinks(i)}$;

Atualiza o valor do PR de cada documento

8) Para *j := 1, 2, ..., N*:
9) $PR(j) := \frac{PR(j)}{\sum_{i=1}^{N} PR(i)}$;

Corrige os PR's para que a soma de todos os PR's resulte em 1. (Lembre-se que PR(j) é a prob do doc j ser acessado)

10) } enquanto existir *j* tal que $|\bar{PR}(j) - PR(j)| > \epsilon$

PageRank – Algoritmo Iterativo

1) Para *j := 1, 2, ..., N*:
2) $PR(j) := \frac{1}{N}$;

Inicializa o *PageRank* para todos os docs

3) Repita
   {
4) Para *j := 1, 2, ..., N*:
5) $\bar{PR}(j) := PR(j)$;

Copia o valor atual de PR para $\overline{\text{PR}}$ para todos os docs

6) Para *j := 1, 2, ..., N*:
7) $PR(j) := \frac{1-d}{N} + d \sum_{i \in L(j)} \frac{\bar{PR}(i)}{NumLinks(i)}$;

Atualiza o valor do PR de cada documento

8) Para *j := 1, 2, ..., N*:
9) $PR(j) := \frac{PR(j)}{\sum_{i=1}^{N} PR(i)}$;

Corrige os PR's para que a soma de todos os PR's resulte em 1. (Lembre-se que PR(j) é a prob do doc j ser acessado)

10) } enquanto existir *j* tal que $|\bar{PR}(j) - PR(j)| > \epsilon$

Testa se há mudança significativa nos PR's

PageRank

- Atualmente, *PageRank* não é o único algoritmo para classificar páginas utilizado pelo Google;
- Todavia ainda é um algoritmo muito importante na área de RI, e pode ser utilizado para construir um algoritmo de ranqueamento mais elaborado.