Reputation Systems I

HITS, PageRank, SALSA, eBay, EigenTrust, VKontakte

Yury Lifshits

Caltech

http://yury.name

Caltech CMI Seminar
March 4, 2008

Wiki Definition

Reputation is the opinion (more technically, a social evaluation) of the public toward a person, a group of people, or an organization

Outline

1. Intro
2. Reputations in Hyperlink Graphs
   - HITS
   - PageRank
   - SALSA
3. Trust Reputations
   - eBay
   - EigenTrust
4. Personal Reputations
   - VKontakte

1

Introduction to Reputations

Applications

- Search
- Trust and recommendations
- Motivating openness & contribution
- Keeping users engaged
- Spam protection
- Loyalty programs

Applications

- Search
- Trust and recommendations
- Motivating openness & contribution
- Keeping users engaged
- Spam protection
- Loyalty programs

Online systems: Slashdot, ePinions, Amazon, eBay, Yahoo! Answers, Digg, Wikipedia, World of Warcraft, BizRate.

Applications

- Search
- Trust and recommendations
- Motivating openness & contribution
- Keeping users engaged
- Spam protection
- Loyalty programs

Online systems: Slashdot, ePinions, Amazon, eBay, Yahoo! Answers, Digg, Wikipedia, World of Warcraft, BizRate.

Russian systems: Habr, VKontakte, Photosight

Aspects

- Input information
- Benefits of reputation
- Centralized/decentralized
- Spam protection mechanisms

Main Ideas

- Random walk model
- Rights, limits and thresholds
- Real name, photo, contact and profile information

Challenges

- Spam protection
- Fast computing
- General theory, taxonomy of existing systems
- Reputation exchange market
- What's inside the real systems?

2

Reputations in Hyperlink Graphs

Challenge

How to define the most relevant webpage to "Bill Gates"?

Challenge

How to define the most relevant webpage to “Bill Gates”?

Naive ideas

- By frequency of query words in a webpage
- By number of links from other **relevant** pages

Web Search: Formal Settings

- Every webpage is represented as a weighted set of keywords
- There are hyperlinks (directed edges) between webpages

Web Search: Formal Settings

- Every webpage is represented as a weighted set of keywords
- There are hyperlinks (directed edges) between webpages

Conceptual problem: define a relevance rank based on keyword weights and link structure of the web

HITS Algorithm

1. Given a query construct a **focused subgraph** $F(q)$ of the web
2. Compute **hubs and authorities** ranks for all vertices in $F(q)$

HITS Algorithm

1. Given a query construct a **focused subgraph** $F(q)$ of the web
2. Compute **hubs and authorities** ranks for all vertices in $F(q)$

Focused subgraph: pages with highest weights of query words **and** pages hyperlinked with them

Hubs and Authorities

Mutual reinforcing relationship:

- A good **hub** is a webpage with many links **to** query-authoritative pages
- A good **authority** is a webpage with many links **from** query-related hubs

Hubs and Authorities: Equations

$$a(p) \sim \sum_{q:(q,p)\in E} h(q)$$

$$h(p) \sim \sum_{q:(p,q)\in E} a(q)$$

Hubs and Authorities: Solution

Initial estimate:

$$\forall p : a_0(p) = 1, h_0(p) = 1$$

Iteration:

$$a_{k+1}(p) = \sum_{q:(q,p)\in E} h_k(q)$$

$$h_{k+1}(p) = \sum_{q:(p,q)\in E} a_k(q)$$

We normalize $\bar{a}_k, \bar{h}_k$ after every step

Convergence Theorem

Theorem

Let M be the adjacency matrix of focused subgraph $F(query)$. Then $\bar{a}_k$ converges to principal eigenvector of $M^T M$ and $\bar{h}_k$ converges to principal eigenvector of MM^T

Lessons from HITS

- Link structure is useful for relevance sorting
- Link popularity is defined by linear equations
- Solution can be computed by iterative algorithm

PageRank: Problem Statement

Compute “quality” of every page

PageRank: Problem Statement

Compute “quality” of every page

Idea: base quality on the number of referring pages and their own quality

PageRank: Problem Statement

Compute “quality” of every page

Idea: base quality on the number of referring pages and their own quality

Other factors:

- Frequency of updates
- Number of visitors
- Registration in affiliated directory

Random Walk Model

Network:

Nodes

Directed edges (hyperlinks)

Random Walk Model

Network:

Nodes

Directed edges (hyperlinks)

Model of random surfer

Start in a random node

Use a random outgoing edge

with probability $1 - \varepsilon$

Move to a random node with probability ε

Random Walk Model

Network:

Nodes

Directed edges (hyperlinks)

Model of random surfer

Start in a random node

Use a random outgoing edge

with probability $1-\varepsilon$

Move to a random node with probability ε

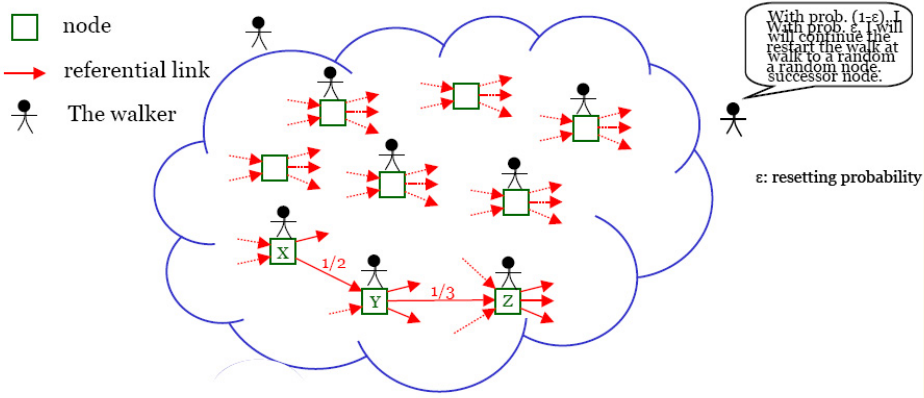
Limit probabilities

For every k the value $PR_k(i)$ is defined as

probability to be in the node i after k steps

Fact: $\lim_{k\to\infty} PR_k(i) = PR(i)$, i.e.

all probabilities converge to some limit ones


node
referential link
The walker
With prob. (1-ε), I will continue the walk to a random successor node.
With prob. ε, I will restart the walk at a random node.
ε: resetting probability
X
Y
Z
1/2
1/3

PageRank Equation

Let $T_1, \ldots, T_n$ be the nodes referring to i
Let $C(X)$ denote the out-degree of X

Claim: $PR(i) = \varepsilon/N + (1-\varepsilon)\sum_{i=1}^{n} \frac{PR(T_i)}{C(T_i)}$

PageRank Equation

Let $T_1, \ldots, T_n$ be the nodes referring to i
Let $C(X)$ denote the out-degree of X

Claim: $PR(i) = \varepsilon/N + (1-\varepsilon)\sum_{i=1}^{n} \frac{PR(T_i)}{C(T_i)}$

Proof?

PageRank Equation

Let $T_1, \ldots, T_n$ be the nodes referring to i
Let $C(X)$ denote the out-degree of X

Claim: $PR(i) = \varepsilon/N + (1-\varepsilon)\sum_{i=1}^{n} \frac{PR(T_i)}{C(T_i)}$

Proof?

By definition of $PR_k(i)$:
$PR_0(i) = 1/N$
$PR_k(i) = \varepsilon/N + (1-\varepsilon)\sum_{i=1}^{n} \frac{PR_{k-1}(T_i)}{C(T_i)}$
Then just take the limits of both sides

PageRank Equation

Let $T_1, \ldots, T_n$ be the nodes referring to i
Let $C(X)$ denote the out-degree of X

Claim: $PR(i) = \varepsilon/N + (1-\varepsilon)\sum_{i=1}^{n} \frac{PR(T_i)}{C(T_i)}$

Proof?

By definition of $PR_k(i)$:
$PR_0(i) = 1/N$
$PR_k(i) = \varepsilon/N + (1-\varepsilon)\sum_{i=1}^{n} \frac{PR_{k-1}(T_i)}{C(T_i)}$
Then just take the limits of both sides

Practical solution: to use $PR_{50}(i)$ computed via iterative formula instead of $PR(i)$

PageRank as an Eigenvector

Let us define a matrix L:

$l_{ij} := \varepsilon/N$, if there is no edge from i to j

$l_{ij} := \varepsilon/N + (1-\varepsilon) \cdot \frac{1}{C(j)}$, if there is an edge

PageRank as an Eigenvector

Let us define a matrix L:

$l_{ij} := \varepsilon/N$, if there is no edge from i to j

$l_{ij} := \varepsilon/N + (1-\varepsilon)\cdot\frac{1}{C(j)}$, if there is an edge

Notation:

$\overline{PR_k} = (PR_k(1), \ldots, PR_k(N))$

$\overline{PR} = (PR(1), \ldots, PR(N))$

PageRank as an Eigenvector

Let us define a matrix L:

$l_{ij} := \varepsilon/N$, if there is no edge from i to j

$l_{ij} := \varepsilon/N + (1-\varepsilon) \cdot \frac{1}{C(j)}$, if there is an edge

Notation:

$\overline{PR_k} = (PR_k(1), \ldots, PR_k(N))$

$\overline{PR} = (PR(1), \ldots, PR(N))$

We have:

$PR_k = L^k PR_0$

$PR = L\,PR$

PageRank as an Eigenvector

Let us define a matrix L:

$l_{ij} := \varepsilon/N$, if there is no edge from i to j

$l_{ij} := \varepsilon/N + (1-\varepsilon) \cdot \frac{1}{C(j)}$, if there is an edge

Notation:

$$\overline{PR_k} = (PR_k(1), \ldots, PR_k(N))$$

$$\overline{PR} = (PR(1), \ldots, PR(N))$$

We have:

$$PR_k = L^k PR_0$$

$$PR = L\,PR$$

SALSA

- Construct query-specific directed graph $F(q)$
- Transform $F(q)$ into undirected bipartite undirected graph W
- Define its column weighted and row weighted versions W_c, W_r
- Consider "hub-authority" random walk: $a^{(k+1)} = W_c^T W_r a^{(k)}$
- Define authorities as the limit value of $a^{(k)}$ vector

3

Trust Reputations

eBay

- Buyers and sellers
- Bidirectional feedback evaluation after every transaction
- eBay Feedback: +/-, four criteria-specific ratings, text comment
- Total score: sum of +/- Feedback points
- 1, 6, 12, months and lifetime versions

EigenTrust

- Local trust $c_ij \geq 0$ is based on personal experience
- Normalization $\sum_{j=1}^{n} c_{ij} = 1$
- Experience matrix C
- Trust equation $t_i^{(k)} = \sum_{j=1}^{n} c_{ij} \cdot t_j^{(k-1)}$

  $t_i^{(k)} = (C^T)^n c_i$
- Trust vector t is the principle eigenvector of C: $t = \lim t_i^{(k)}$

EigenTrust: Pre-Trusted Nodes

- Starting vector. Let $\mathcal{P}$ is the set of pre-trusted nodes. Use $t^{(0)} = 1/|\mathcal{P}|$
- Local trust. Assume ε local trust from any node to any pre-trusted node

4

Personal Reputations

VKontakte

What is VKontakte.ru?

- Russian “Facebook-style” website
- Name means “in touch” in Russian
- 8.5M users (February 2008)
- Working on English language version

VKontakte Rating

1. First 100 points: real name and photo, profile completeness
2. Then: paid points (via SMS) gifted by your supporters
3. Any person has 1 free reference link, initially pointing to a person who invited him to VKontakte. Bonus points (acquired by rules 2 and 3) are propagating with 1/4 factor by reference links.

VKontakte Rating

1. First 100 points: real name and photo, profile completeness
2. Then: paid points (via SMS) gifted by your supporters
3. Any person has 1 free reference link, initially pointing to a person who invited him to VKontakte. Bonus points (acquired by rules 2 and 3) are propagating with 1/4 factor by reference links.

Rating benefits:

- Basis for sorting: friends lists, group members, event attendees
- Bias for “random six friends” selection

References

J. Kleinberg
Authoritative sources in a hyperlinked environment

L. Page, S. Brin, R. Motwani, T. Winograd
The Pagerank citation ranking: Bringing order to the web

R. Lempel, S. Moran
The stochastic approach for link-structure analysis (SALSA) and the TKC effect

D. Houser, J. Wooders
Reputation in Auctions: Theory, and Evidence from eBay

S.D. Kamvar, M.T. Schlosser, H. Garcia-Molina
The Eigentrust algorithm for reputation management in P2P networks

VKontakte Team
`http://vkontakte.ru/rate.php?act=help` (in Russian)

http://yury.name

Ongoing project: http://businessconsumer.net http://yury.name

Ongoing project: http://businessconsumer.net

Thanks for your attention!

Questions?

Second part (March 11, 4pm):

- Spam protection for reputations
- Open problems