Learning to Rank Networked Entities

Alekh Agarwal Soumen Chakrabarti Sunny Aggarwal

Presented by
Dong Wang

11/29/2006

We've all heard that a million monkeys banging on a million typewriters will eventually reproduce the entire works of Shakespeare. Now, thanks to the Internet, we know this is not true.

--Robert Wilensky

What's in the Web?

- Highly structured HTML
- Fully general natural language contained within the HTML
- HTML with embedded images
- Most important, link structure which is a directed graph over all the Web pages, with labels on the links (the text of the anchors).

Information Retrieval

Goal: Given a query of several key words, find the most similar document which contains these key words.

Method: Treat the query and all the documents as vectors in a vector space. The dimension of this vector space is the number of all possible key words. If the query vector is Vq, and the ith document's vector is Vdi, we want to take the document which yields the largest Vq'•Vdi.

Problem About Information Retrieval

The document has the largest “similarity” may not be the most relevant document we are looking for.

Example: The query “Bill Clinton” may return a webpage of “Bill Clinton sucks”, with his name appears repeatedly, instead of his biography at the White House webpage.

Google's solution: PageRank

Main idea: Rank all the webpages on the Web by their *importance* or *quality*, and return the webpage in the order of this rank.

$$PR(A) = (1-d) + d(PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))$$

PR(A) can be calculated using a simple iterative algorithm.

Computing the Google PageRank

The World's Largest Eigenvalue Problem

What is PageRank

A search with Google's search engine usually returns a very large number of pages. E.g., a search on 'weather forecast' returns 5.5 million pages.

Although the search returns several million pages, the most relevant pages are usually found within the top ten or twenty pages in the list of results.

How does the search engine know which pages are the most important?

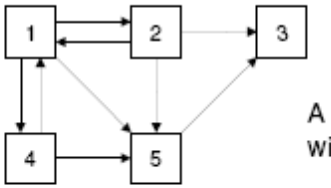
Google assigns a number to each individual page, expressing its importance. This number is known as the PageRank and is computed via the eigenvalue problem

$$Pw = \lambda w$$

where P is based on the link structure of the Internet.

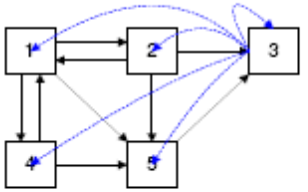
The key problem is to formulate the link structure, i.e., the matrix P, in a proper way.

The Link Structure Matrix *P*



A tiny internet with 5 pages.

The model forming the basis of the PageRank algorithm is a random walk through all the pages of the Internet. Let $p_t(x)$ denote the possibility of being on page x at time t. The PageRank of page x is expressed as $\lim(p_t(x))$ for $t \to \infty$. To make sure the random walk process does not get stuck, pages with no out-links (here: page 3) are assigned artificial links or "teleporters" to all other pages.



$$P = \begin{pmatrix} 0 & 1/3 & 1/5 & 1/2 & 0 \\ 1/3 & 0 & 1/5 & 0 & 0 \\ 0 & 1/3 & 1/5 & 0 & 1 \\ 1/3 & 0 & 1/5 & 0 & 0 \\ 1/3 & 1/3 & 1/5 & 1/2 & 0 \end{pmatrix}$$

The matrix P is irreducible and stochastic and therefore the random walk can be expressed as a Markov chain, and the PageRank of all pages can be computed as the principal eigenvector of P.

The PageRank Algorithm

The Google matrix P is currently of size 4.2×10^9 and therefore the eigenvalue computation is not trivial. To find an approximation of the principal eigenvector the *power method* is used:

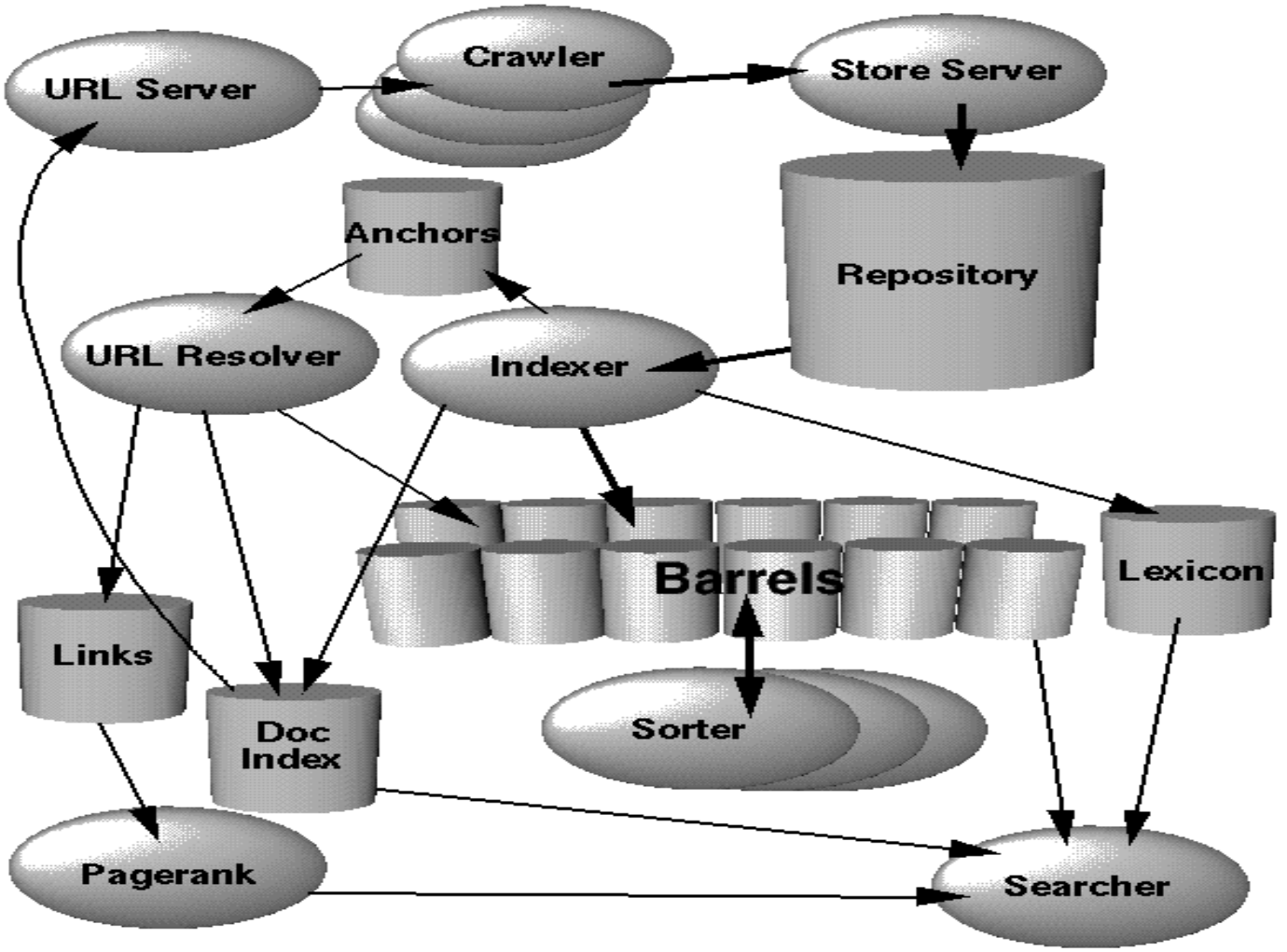
```
w_0 = initial guess
For k = 1 to 50
    w_k = P*w_{k-1}
End
Return w_50
```

The special properties of the matrix P ensures that the largest eigenvalue is $\lambda = 1$, rendering normalisation in the power method unnecessary. Fast convergence of the power method makes 50 iterations adequate.

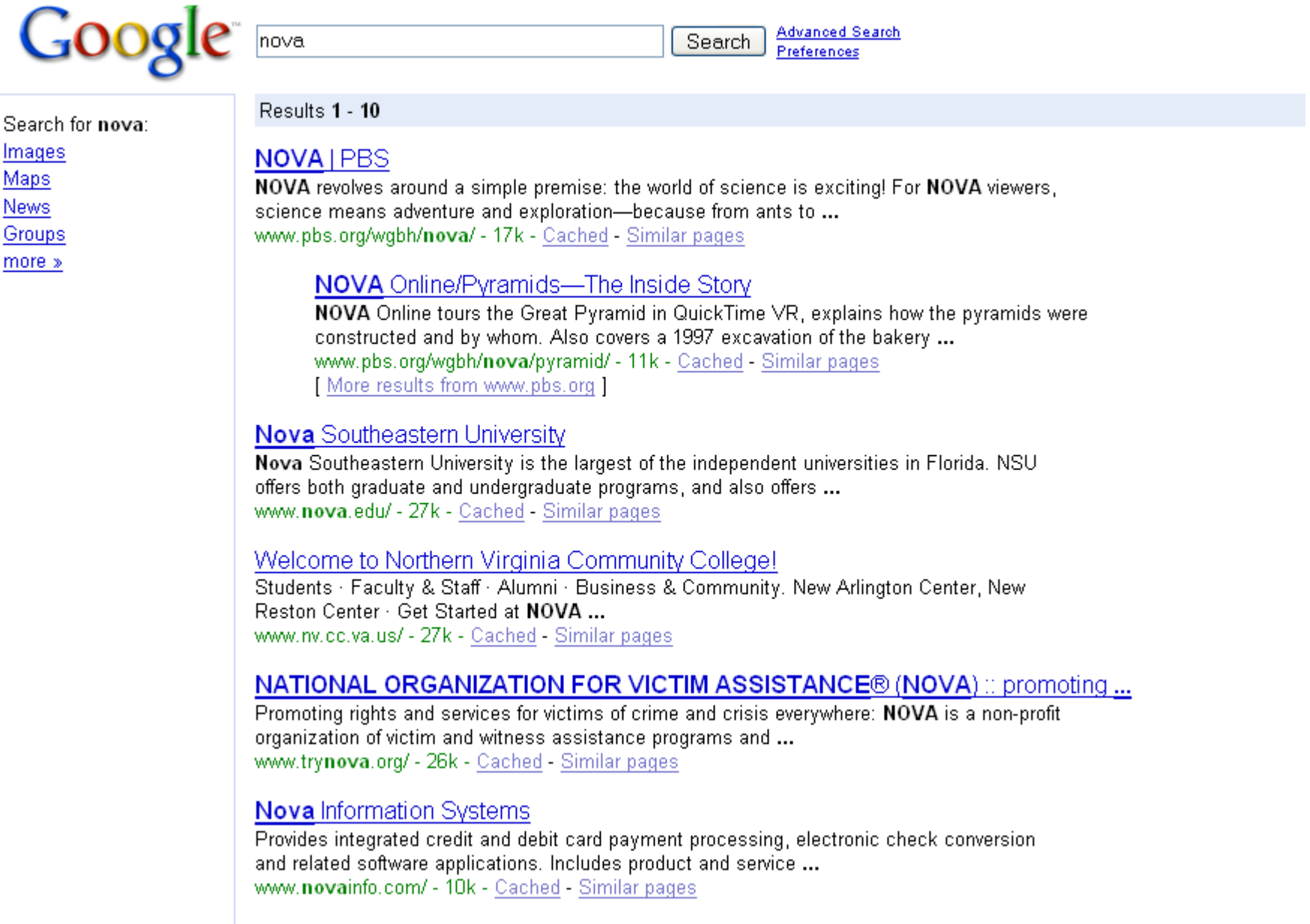
Because the computation involves an extremely large matrix, the matrix-vector multiplications must be implemented in parallel on multi-processor systems.

DTU

High Level Google Architecture

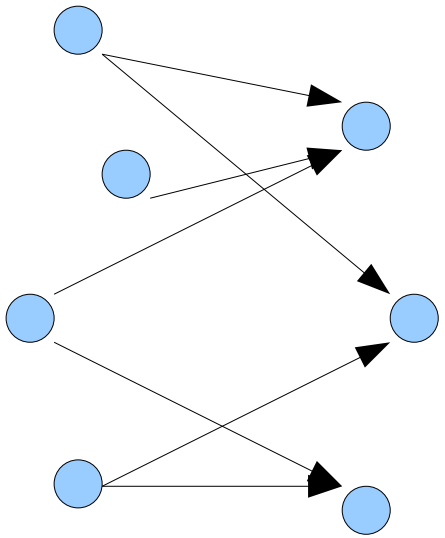


An example


Google
nova
Search
Advanced Search
Preferences
Search for nova:
Images
Maps
News
Groups
more »
Results 1 - 10
NOVA | PBS
NOVA revolves around a simple premise: the world of science is exciting! For NOVA viewers, science means adventure and exploration—because from ants to ...
www.pbs.org/wgbh/nova/ - 17k - Cached - Similar pages
NOVA Online/Pyramids—The Inside Story
NOVA Online tours the Great Pyramid in QuickTime VR, explains how the pyramids were constructed and by whom. Also covers a 1997 excavation of the bakery ...
www.pbs.org/wgbh/nova/pyramid/ - 11k - Cached - Similar pages
[More results from www.pbs.org]
Nova Southeastern University
Nova Southeastern University is the largest of the independent universities in Florida. NSU offers both graduate and undergraduate programs, and also offers ...
www.nova.edu/ - 27k - Cached - Similar pages
Welcome to Northern Virginia Community College!
Students · Faculty & Staff · Alumni · Business & Community. New Arlington Center, New Reston Center · Get Started at NOVA ...
www.nv.cc.va.us/ - 27k - Cached - Similar pages
NATIONAL ORGANIZATION FOR VICTIM ASSISTANCE® (NOVA) :: promoting ...
Promoting rights and services for victims of crime and crisis everywhere: NOVA is a non-profit organization of victim and witness assistance programs and ...
www.trynova.org/ - 26k - Cached - Similar pages
Nova Information Systems
Provides integrated credit and debit card payment processing, electronic check conversion and related software applications. Includes product and service ...
www.novainfo.com/ - 10k - Cached - Similar pages

Another Ranking Family: HITS

Hypertext Induced Topic Selection is a link analysis algorithm which rates webpages for their authority values and hub values.



a(p) := sum of h(q), for all q pointing to p

h(q) := sum of a(p), for all p pointed by q

Some Other Ranking Algorithms

In standard PageRank, all edges are considered the same.

1. In ObjectRank, Intelligent Surfer and XRank, the random walk favors nodes containing query keywords in a fixed, arbitrary manner.
2. In topic-sensitive Pagerank, the random walk preferentially moves to webpages on a specified topic.
3. In personalized Pagerank, the random walk preferentially moves to pages visited by the user in the past.

Problems about current algorithms and this paper's contribution

1. Current algorithms do not model network connections or relations between entities. So this paper associated the edges with a few types.

2. Current algorithms only find the stationary distribution of a reasonable but arbitrary Markov walk over a network, do not learn from relevance feedback. So this paper estimate the transition probability on each edge separately and learn the probability from relevance feedback.

Partial order "$\prec$"

We need to learn the vector $w \in R^d$, Given a partial order $\prec$ involving some of the entities. If $u \prec v$, we want $w'x_u \leq w'x_v$. The partial order is set arbitrary by the user, because the user always has one or few favorite communities.

Scoring feature vectors

A max-margin search for w introduces a set of slack variables $s^*_{ij} \geq 0$ and we need to solve the quadratic optimization

$$\min_{s \geq \mathbf{0}; w \in \mathbb{R}^d} w'w + B \sum_{(i,j):i \prec j} s_{ij} \quad \text{subject to}$$

$$w'x_j - w'x_i + s_{ij} \geq 1 \quad \forall i \prec j \qquad \text{(RankSVM)}$$

Note that if $w'x_j \geq 1 + w'x_i$ then $i \prec j$ is satisfied. And no graphical connection is modeled between any x_i and x_j, they remain independent feature vectors.

PageRank and Teleport optimization

C is a |v| X |v| transition matrix which is designed as:

$$C(j,i) = \begin{cases} \alpha \frac{[\![(i,j) \in E]\!]}{\text{OutDegree}(i)} + (1-\alpha) r_j, & i \in V_o \\ r_j, & \text{otherwise} \end{cases}$$

(UnweightedPagerank)

There are two design variables. α is the probability of walking to a neighbor instead of jumping to a random node; r is the teleport or personalization vector.

Given teleport vector $r \in R^{|v|}$, the Pagerank vector satisfies

$$p = \alpha A' p + (1-\alpha) r, \quad \text{and therefore}$$
$$p = (1-\alpha)(\mathbb{I} - \alpha A')^{-1} r = Mr$$

Hard Constraint and Soft Constrint

Hard constraint:

$$\min_{r \in \mathbb{R}^{|V|}} (Mr - Mr^U)'(Mr - Mr^U)$$
$$\text{s.t.} \quad \Pi M r \geq \mathbf{0}, \quad r \geq \mathbf{0}, \quad \mathbb{1}'r = 1.$$

Soft constraint:

$$\min_{r \in \mathbb{R}^{|V|}} (Mr - Mr^U)'(Mr - Mr^U) + \underline{B\, r'(M'\Pi'\Pi M)r}$$
$$\text{s.t.} \quad r \geq \mathbf{0}, \quad \mathbb{1}'r = 1.$$

Tuning edge weights

Any edge e with type t(e) has a strictly positive weight β(t(e)) > 0, a nonexistent edge has weight zero. The modified Pagerank transition matrix is

$$C(j,i) = \begin{cases} \alpha \frac{\beta(t(i,j))}{\mathrm{OutWeight}(i)} + (1-\alpha) r_j, & i \in V_o \\ r_j, & \text{otherwise} \end{cases} \qquad \text{(WeightedPagerank)}$$

Note that we are looking for β such that the p that solves p = Cp also satisfies $\prec$, now we are facing quadratic equality constraints, which is difficult to solve.

Primal formulation

We add a special dummy node d, and directed edges (v,d) and (d,v) for all v $\in$ V. The augmented graph is called G' = (V', E')

$$\min_{\{0 \le p_{uv} \le 1\}} \sum_{(u,v) \in E'} p_{uv} \log p_{uv} \quad \text{(HardObjective)}$$

$$\text{such that} \sum_{(u,v) \in E'} p_{uv} - 1 = 0 \quad \text{(Total)}$$

$$\forall v \in V' : -\sum_{(u,v) \in E'} p_{uv} + \sum_{(v,w) \in E'} p_{vw} = 0 \quad \text{(Balance)}$$

$$\forall v \in V_o : -\alpha p_{vd} + (1-\alpha) \sum_{(v,w) \in E} p_{vw} = 0 \quad \text{(Teleport)}$$

$$\forall u \prec v : \sum_{(w,u) \in E'} p_{wu} - \sum_{(w,v) \in E'} p_{wv} \le 0 \quad \text{(Preference)}$$

Constraint inclusion heuristic

Input: $V^{'}$, $E^{'}$, $\prec$ and tolerance ϵ
Let $\mathcal{B}, \mathcal{T}, \mathcal{P}$ be current sets of dual variables
$\mathcal{B} \leftarrow \varnothing$, $\mathcal{T} \leftarrow \varnothing$, $\mathcal{P} \leftarrow \varnothing$ (implicitly all $\beta, \pi, \tau = 0$)
repeat
  {estimate violations}
  $\mathcal{V}(\beta_v) = |\,\mathrm{OutFlow}(v) - \mathrm{InFlow}(v)|$
  $\mathcal{V}(\tau_v) = |\alpha p_{vs} - (1-\alpha)\sum_{(v,w)\in E} p_{vw}|$
  $\mathcal{V}(\pi_{u,v}) = \mathrm{InFlow}(v) - \mathrm{InFlow}(u)$
  discard candidates with violation $\mathcal{V}$ less than ϵ
  $\mathcal{B} \leftarrow \mathcal{B} \cup \arg\max^{(k)}_{v\in V'} \mathcal{V}(\beta_v)$
  $\mathcal{T} \leftarrow \mathcal{T} \cup \arg\max^{(k)}_{v\in V'} \mathcal{V}(\tau_v)$
  $\mathcal{P} \leftarrow \mathcal{P} \cup \arg\max^{(k)}_{(u,v)\in \prec} \mathcal{V}(\pi_{u,v})$
  Run dual optimizer over variables in $\mathcal{B}, \mathcal{T}, \mathcal{P}$
until $\mathcal{B}, \mathcal{T}, \mathcal{P}$ stabilize or test accuracy saturates

Experiment results

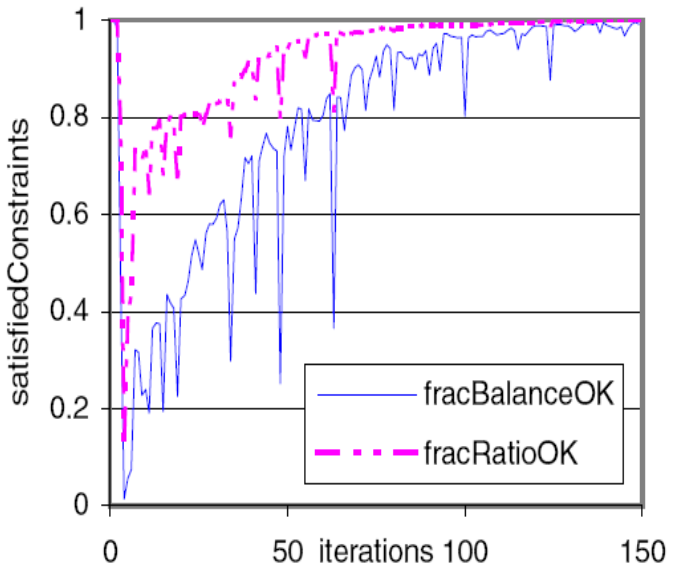


Figure 3: Satisfaction of primal feasibility constraints (Balance) and (Teleport) as dual optimization progresses.

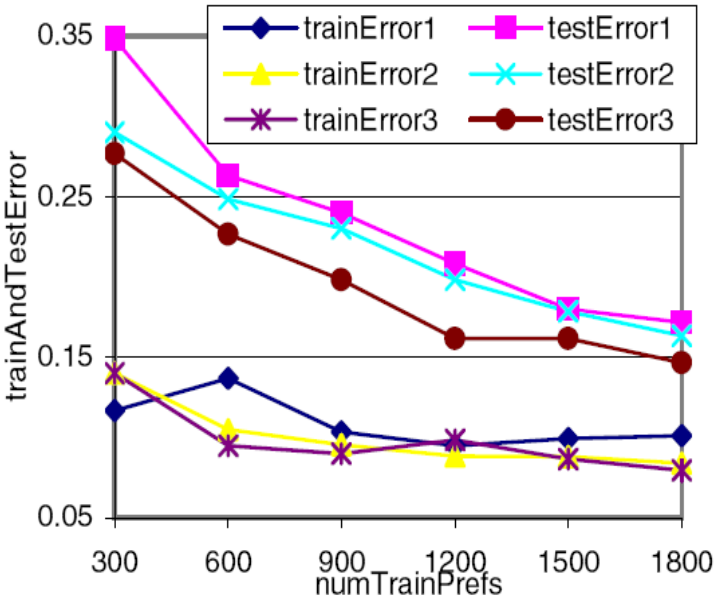


Figure 4: Reduction in test error as training $|\prec|$ is increased, for three random choices of the hidden teleport seed.

Learning Edge Conductances

The conductance matrix C used in UnweightedPagerank is modified to reflect edge weights, as follows:

$$C(j,i) = \begin{cases} \frac{\alpha\beta(t(i,j))}{\sum_{j'} \beta(t(i,j'))}, & (i,j) \in E \\ 1 - \alpha[\![i \in V_o]\!], & i \in V, j = d, \\ r_j & i = d, j \in V \\ 0, & \text{otherwise} \end{cases} \quad \text{(Conductance)}$$

Note that C is a function of β, and we seek a set of βs such the p solves p = Cp and p satisfies $\prec$.

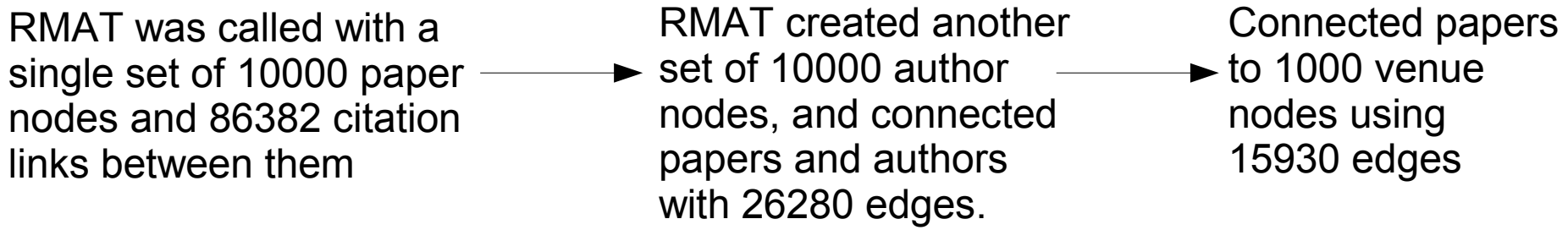
Iterative approximation to $\partial p_u / \partial \beta(t)$

Because we are searching for β(t)s, we need to find the gradient of loss(pu-pv) w.r.t. β(t) for each type t. Let g(u,t) be an approximation to $\partial p_u / \partial \beta(t)$

Initialize all $p_v^{(0)} \leftarrow 1/|V'|$ and $g^{(0)}(v,t) = 0$ for all v, t
$\ell \leftarrow 0$
while any element of p or g changes significantly **do**
  $\ell \leftarrow \ell + 1$
  for each u set $p_u^{(\ell)} \leftarrow \sum_v C(u,v) p_v^{(\ell-1)}$
  for each node u and type t **do**
    $g^{(\ell)}(u,t) \leftarrow \sum_v \frac{\partial C(u,v)}{\partial \beta(t)} p_v^{(\ell-1)} + C(u,v) g^{(\ell-1)}(v,t)$
  end for
end while

Experiment : Generating realistic typed graphs

Generating a synthetic graph through a single call to RMAT and then randomly assigning types and weights would lead to very unrealistic graphs. We want to generate natural graphs with typed nodes and edges, such as DBLP or CiteSeer citation graphs



Generating $\prec$ using hidden edge weights

The default weights are assigned as
paper-author : 6, 10
paper-paper: 20
paper-venue: 1, 4

These weights are “hidden” to the search engine. We will see in the results that the approximate gradient-descent is very effective at recovering the hidden parameters that led to $\prec$, in terms of both accuracy and speed.

Results

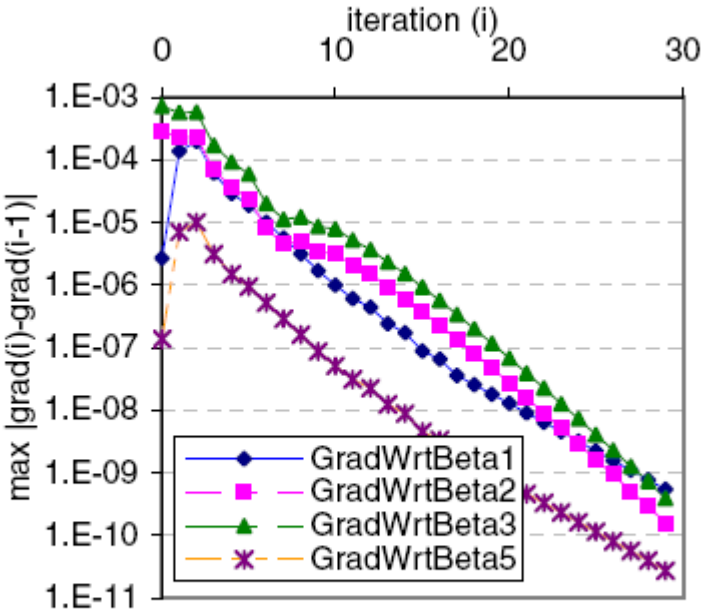


Figure 11: Like Pagerank itself, the gradients converge within very few iterations.

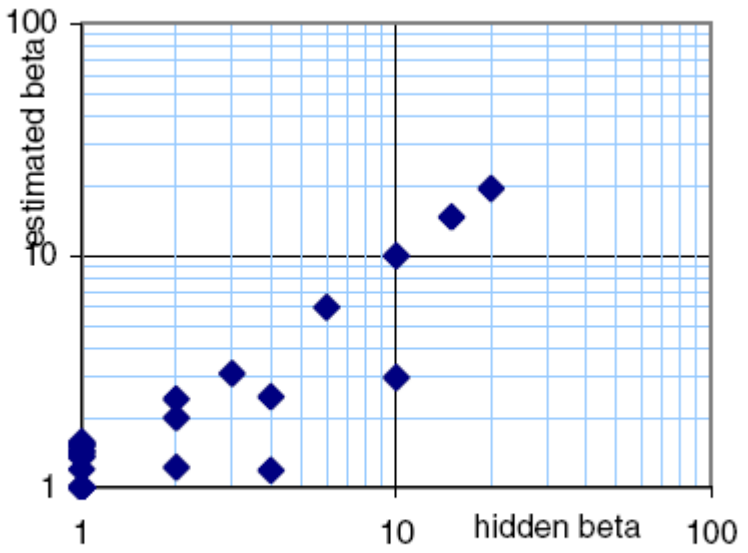


Figure 13: Accuracy of estimation of hidden βs.

Conclusions

The study targeted at learning the parameters of Markovian walks in graphs to satisfy pairwise preference constraints between nodes.

Two learning problems in this framework are presented:

1. The preferences hint at one or more favored communities that the learning algorithm must discover. A maximum entropy flow estimation algorithm for this setting was proposed.
2. Edges have types that determine their conductance, and the learner must estimate these weights. An approximate gradient-descent algorithm for this setting was proposed.

Related papers

K. Anywanwu, A. Maduko, and A. Sheth. SemRank: Ranking complex semantic relationship search results on the semantic Web. In *WWW Conference*, pages 117-127, Chiba, Japan, 2005.

S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW Conference*, 1998.

T. H. Haveliwala. Topic-sensitive PageRank. In *WWW*, pages 517-526, 2002.

R. Herbrith, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *International Conference on Artificial Neural Networks*, pages 97-102, 1999.

J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *JACM*, 46(5):604-632, 1999.

Questions?