Index Design for Dynamic Personalized PageRank

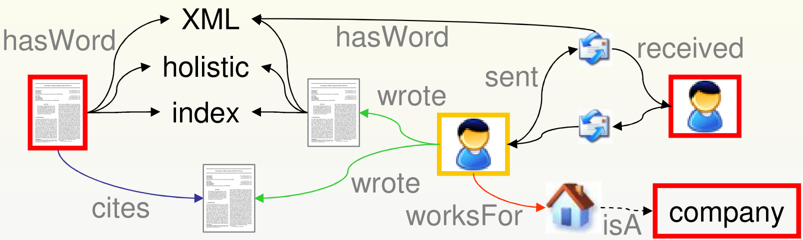
Amit Pathak
Soumen Chakrabarti
Manish Gupta

IIT Bombay
amit@cse.iitb.ac.in soumen@cse.iitb.ac.in manishg@cse.iitb.ac.in

Mar 1, 2008

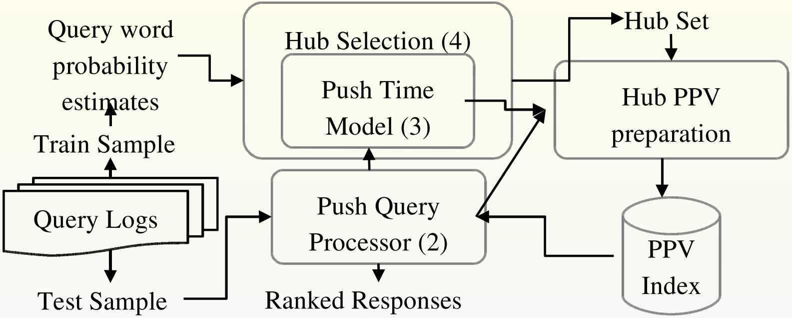
Motivation

- Graph structured databases



- Find expert e from industry to review a submitted paper p

System



- 50–400$\times$ faster than whole-graph Pagerank
- 10–20% smaller index, $>$ 94% accuracy
- Low index size, processing time and query time

Notations

- Graph $G = (V, E)$ with edges $(u, v) \in E$
- Conductance $C(v, u)$ such that $\sum_v C(v, u) = 1$
- Teleport prob $1 - \alpha$ and vector r, $\sum_v r(v) = 1$
- Personalized PageRank (PPR) for vector r is $PPV_r = p_r = \alpha C p_r + (1 - \alpha) r = (1 - \alpha)(I - \alpha C)^{-1} r$
- For node v, $r(v) = 1 \Rightarrow$ its PPV is $PPV_{\delta_v} = PPV_v$

Previous work

- ObjectRank: Connects word node w to all entities where it is mentioned
- Precomputes and stores PPV_w for all words w
- Preprocessing costs increase with increase in graph and vocabulory size
- 22000 CPU hours for 562000 words

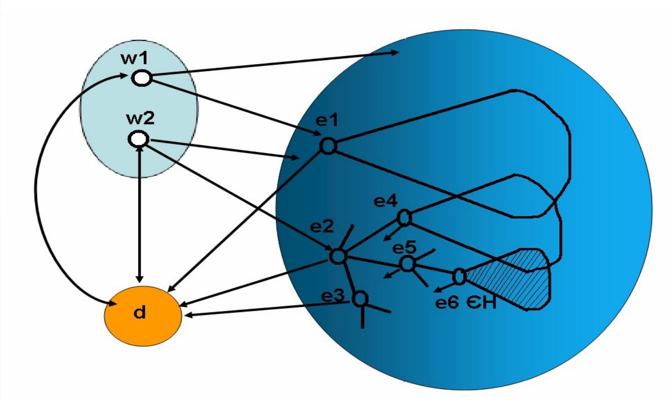
Asynchronous Weight-Pushing Algorithm (BCA)

- $p_r = (1-\alpha)\left(\sum_{k \geq 0} \alpha^k C^k\right) r$

$q \leftarrow r, N_{H,r} \leftarrow \vec{0}, B_{H,r} \leftarrow \vec{0}$
while $\|q\|_1 > \epsilon_{\text{push}}$ **do**
  pick $\arg\max_u q(u)$ {delete-max }
  $\hat{q} \leftarrow q(u)$, $q(u) \leftarrow 0$
  if $u \in H$
    $B_{H,r}(u) \leftarrow B_{H,r}(u) + \hat{q}$
  else
    $N_{H,r}(u) \leftarrow N_{H,r}(u) + (1-\alpha)\hat{q}$
    for each out-neighbor v of u **do**
      $q(v) \leftarrow q(v) + \alpha C(v,u)\hat{q}$ {increase-key }
return $N_{H,r} + \sum_{h \in H} B_{H,r}(h)\, \mathrm{PPV}_h$

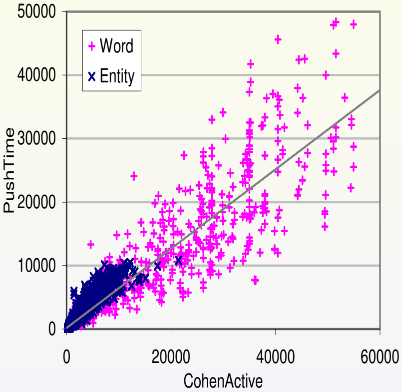
Cost-benefit model

- d is the dummy node; $w1$, $w2$ are word nodes
- $e1, e2, \cdots, e6$ are entity nodes
- Shaded area represents the work saved if $e6 \in$ hubset H



Modelling push algorithm performance

- Cost-benefit optimizer
- Estimate BCA running time
- Exact number of push steps
- PushActive$(H, \delta_o, \epsilon_{\text{push}})$
- PathActive$(H, \delta_o, \epsilon_{\text{push}})$
- Cohen's Algorithm
- CohenActive$(H, \delta_o, \epsilon_{\text{push}})$ by $D = -\log \epsilon_{\text{push}}$



Hubset Selection

- Benefit: Work saved by inclusion of node u in H
- Cost: Space to save PPV_u
- Work saved for one query if $u \in H$ is estimated by a regression from CohenActive
- Work saved by u over query workload is $\sum_w \tilde{f}(w)\,\text{WorkSaved}(H, \delta_w, u)$

Clipped PPV$_u$ storage space

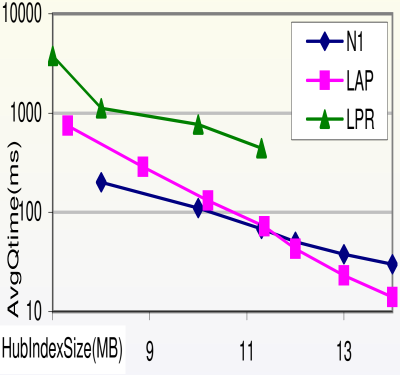
- Full $PPV_h \ \forall h \in H$ takes huge space
- Clipping decreases size without much decrease in accuracy
- Cost-benefit optimizer needs clipped PPV size
- Estimate using power-law distribution of PPV size and modified Cohen

Hub inclusion policies

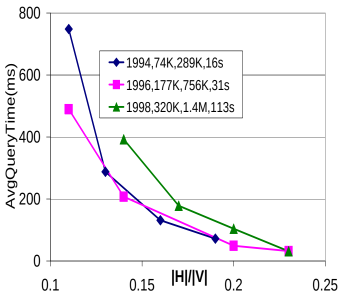
- Large PageRank (LPR): order by decreasing global PageRank with $r = \vec{1}/|V|$
- Naive one-shot or N1: Chakrabarti ordered hubs u one-shot by only $N_{H,\tilde{f}}(u)$
- LookAhead Progressive (LAP): Include a fixed number of nodes with high benefit/cost into the hubset at each iteration.

Experiments

- CiteSeer corpus - 709K words, 1.1M entities, 3.7M edges
- Temporal snapshots - 1994, 1996, 1998, 2000
- Lucene text indices - **55, 139, 259, 378 MB** resp
- 1.9M CiteSeer queries – 2.68 words/query
- Disjoint 100K train queries and 10K test queries
- Beats Chakrabarti wrt index size (10$\times$) and query speed (10$\times$). RAG, precision and τ accuracy (at rank 20) of **0.998, 0.95 and 0.94**



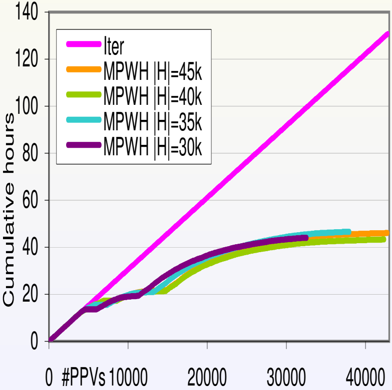
Experiments



- By scaling $|H|$ at a small fraction of $|V|$, HubRank query times can be held independent of $|G|$.
- Our index size is 9–13MB for 1994 graph

Building PPV index

- Baseline: compute PPV_h for each $h \in H$ independently using Power Iterations. Time $\propto |H|$.
- MPWH (Max PPR wrt H): First schedule nodes h which block many "heavy" paths from other (pending) hubs; estimated by CohenActive.



Bibliography

- Pavel Berkhin, *Bookmark-coloring approach to personalized pagerank computing*, Internet Mathematics **3** (2007), no. 1, 41–62.
- Andrey Balmin, Vagelis Hristidis, and Yannis Papakonstantinou, *Objectrank: Authority-based keyword search in databases.*, VLDB, 2004, pp. 564–575.
- Soumen Chakrabarti, *Dynamic personalized PageRank in entity-relation graphs*, www (Banff), May 2007.
- Edith Cohen, *Estimating the size of the transitive closure in linear time*, focs, 1994, pp. 190–200.
- Glen Jeh and Jennifer Widom, *Scaling personalized web search*, WWW Conference, 2003, pp. 271–279.