Fast Query Execution for Retrieval Models based on Path Constrained Random Walks

Ni Lao, William W. Cohen

Carnegie Mellon University

2010.7.27

Outline

- Background
  - Retrieval models based on Path Constrained Random Walks (ECML PKDD 2010)
- Efficient PCRW
  - Comparing sampling and truncation strategies

Relational Retrieval Problems

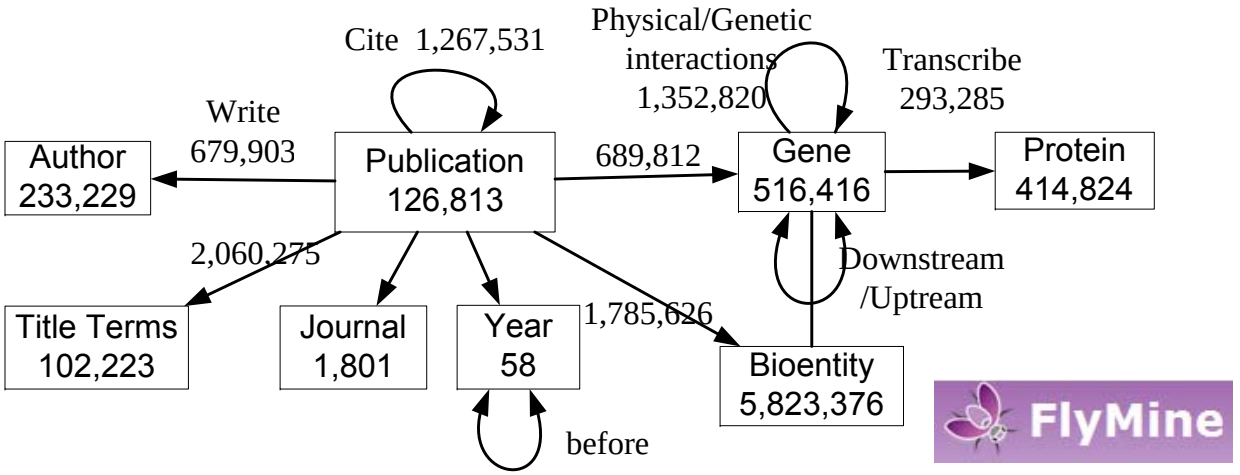
- Data of many retrieval/recommendation tasks can be represented as labeled directed graphs
  - Typed nodes: documents, terms, metadata
  - Labeled edges: authorOf, datePublished

- Can support a family of *typed proximity queries*
  - ad hoc retrieval: term nodes→ documents
  - gene recommendation : user, year→ gene
  - Reference (citation) recommendation: topic → paper
  - Expert finding: topic → user
  - Collaborator recommendation : scientist→ scientist

- How to measure the proximity between query and target nodes?

Biology Literature Data

- Data of this study
  - Yeast: 0.2M nodes, 5.5M links
  - Fly: 0.8M nodes, 3.5M links
  - E.g. the fly graph

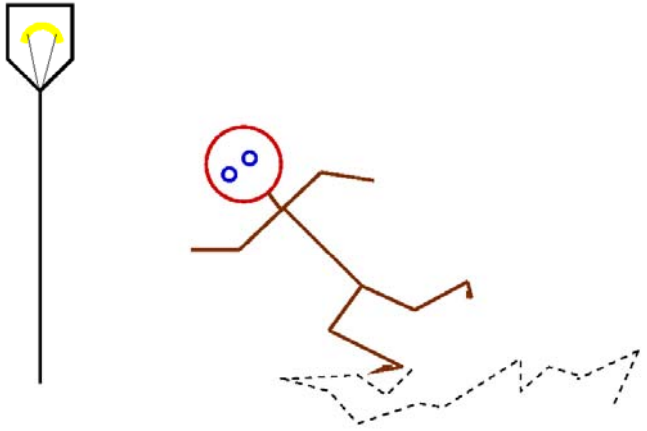


- Tasks
  - Gene recommendation: author, year→gene
  - Reference recommendation: title words,year→paper
  - Expert-finding: title words, genes→author

Random Walks with Restart (RWR) as A Proximity Measured

- RWR is a commonly used similarity measure on Labeled Graphs
  - Topic-sensitive Pagerank (Haveliwala, 2002)
  - Personalized Pagerank (Jeh &. Widom, 2003)
  - ObjectRank (Balmin et al., 2004),
  - Personal information management (Minkov & Cohen, 2007)

- RWR can be improved by supervised learning of edge weights
  - quadratic programming (Tsoi et al., 2003),
  - simulated annealing (Nie et al., 2005),
  - back-propagation (Diligenti et al., 2005; Minkov & Cohen, 2007),
  - limit memory Newton method (Agarwal et al., 2006)



The Limitation of RWR

- One-parameter-per-edge label is limited because the context in which an edge label appears is ignored
  - E.g. (observed from real data)

Path	Comments
$author \xrightarrow{Read} paper \xrightarrow{Contain} gene \xrightarrow{Contain^{-1}} paper$	Don't read about **genes** which I have already read
$author \xrightarrow{Read} paper \xrightarrow{Write^{-1}} author \xrightarrow{Write} paper$	Read about my favorite **authors**

Path	Comments
$author \xrightarrow{Write} paper \xrightarrow{Contain} gene \xrightarrow{Contain^{-1}} paper$	Read about the **genes** that I am working on
$author \xrightarrow{Write} paper \xrightarrow{publish^{-1}} institute \xrightarrow{publish} paper$	Don't need to read paper from **my own lab**

Path Constrained Random Walk –A New Proximity Measure

- Our previous work (Lao& Cohen, ECML 2010)
  - learn a weighted combination of simple “path experts”, each of which corresponds to a particular labeled path through the graph

- Reference recommendation--an example
  - In the TREC-CHEM Prior Art Search Task, researchers found that it is more effective to first find patents about the topic, then aggregate their citations
  - Our proposed model can discover this kind of retrieval schemes and assign proper weights to combine them. E.g.

Weight Path

$$272.4\ word \xrightarrow{HasTitle^{-1}} paper \xrightarrow{Cite^{-1}} paper \xrightarrow{Cite} paper$$

$$156.7\ word \xrightarrow{HasTitle^{-1}} paper \xrightarrow{Cite} paper$$

$$41.4\ word \xrightarrow{HasTitle^{-1}} paper$$

Definitions

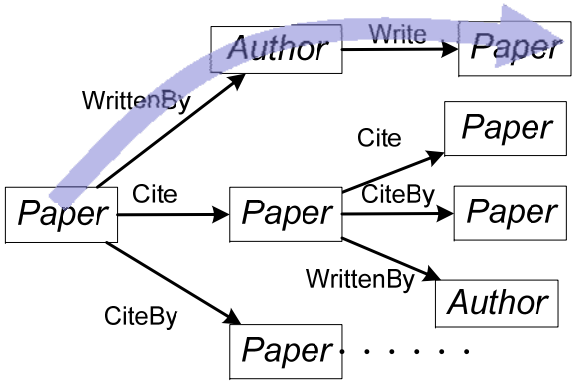
- An *Entity-Relation graph* G=(**T**,**E**,**R**), is
  - a set of *entities types* **T**={T}
  - a set of *entities* **E**={e}, Each entity is typed with $e.T \in$ **T**
  - a set of relations **R**={R}

- A *Relation path* $P=(R_1, \ldots, R_n)$ *is a sequence of relations*
  - E.g. $year \xrightarrow{PublishedIn^{-1}} paper$

    $year \xrightarrow{PublishedIn^{-1}} paper \xrightarrow{Cite} paper$

- *Path Constrained Random Walk*
  - Given a query q=($\mathbf{E}_q$,T_q)
  - Recursivelv define a distribution for each path

$$h_{E_q,P}(e) = \sum_{e' \in range(P')} h_{E_q,P'}(e') \cdot \frac{I(R_\ell(e',e))}{|R_\ell(e')|}$$



Supervised Retrieval Model Based on PCRW

- A retrieval model can rank target entities by linearly combine the distributions of different paths

$$score(e;\theta,L)=\sum_{P\in\mathbf{P}(q,L)}h_P(e)\theta_P$$

  - or in matrix form $s=A\theta$

- This mode can be optimized by maximizing the probability of the observed relevance

$$p_e^{(m)}=p(y_e^{(m)}=1\,|\,q^{(m)};\theta)=\frac{\exp(\theta^T A_e^{(m)})}{1+\exp(\theta^T A_e^{(m)})}$$

  - Given a set of training data D={($q^{(m)}$, $A^{(m)}$, $y^{(m)}$)}, $y_e^{(m)}$=1/0

- This PCRW based model can significantly improve retrieval quality over the RWR based models (Lao & Cohen, ECML 2010)

Outline

- Background
  - Retrieval Models with PCRW (ECML PKDD 2010)
- Efficient PCRW
  - Comparing sampling and truncation strategies

The Need for Efficient PCRW

- Random walk based model can be expensive to execute
  - Especially for dense graphs, or long random walk paths
- Popular speedup strategies are
  - Sampling (finger printing) strategies
    - Fogaras 2004
  - Truncation (pruning) strategies
    - Chakrabarti 2007
  - Build two-level representations of graphs offline
    - Raghavan et al., 2003, He et al., 2007, Dalvi et al., 2008
    - Tong et al. 06---low-rank matrix approximation of the graph
    - Chakrabarti 2007 ---precompute Personalized Pagerank Vectors (PPVs) for a small fraction of nodes
- In this study, we will compare different sampling and truncation strategies applied to PCRW

Four Strategies for Efficient Random Walks

- Fingerprinting (sampling)
  - Simulate a large number of random walkers

$$h_{i+1}(e) = \frac{\#\text{times the walkers visit } e}{\#\text{walkers}}$$

- Fixed Truncation
  - Truncate by fixed value

$$h_{i+1}(e) = max(0, h_{i+1}(e) - \varepsilon)$$

- Beam Truncation
  - Keep top W probable entities

$$h_{i+1}(e) = max(0, h_{i+1}(e) - \varepsilon_W)$$

- Weighted Particle Filtering
  - A combination of exact inference and sampling

Weighted Particle Filtering

Algorithm 1 Weighted Particle Filtering

Input: distribution $h_i(e)$, relation R, threshold ε_{min}
Output: $h_{i+1}(e)$
Set $h_{i+1}(e) = 0$ (should not take any time)
for each e with $h_i(e) \neq 0$ **do**
  $size_{new} = h_i(e)/|R(e)|$
  if $size_{new} > \varepsilon_{min}$ **then**
    for each $e' \in R(e)$ **do**
      $h_{i+1}(e') += size_{new}$
    end for
  else
    for k=1..$floor(h_i(e)/\varepsilon_{min})$ **do**
      randomly pick $e' \in R(e)$
      $h_{i+1}(e') += \varepsilon_{min}$
    end for
  end if
end for

Start from exact inference

switch to sampling when the branching is heavy

Experiment Setup

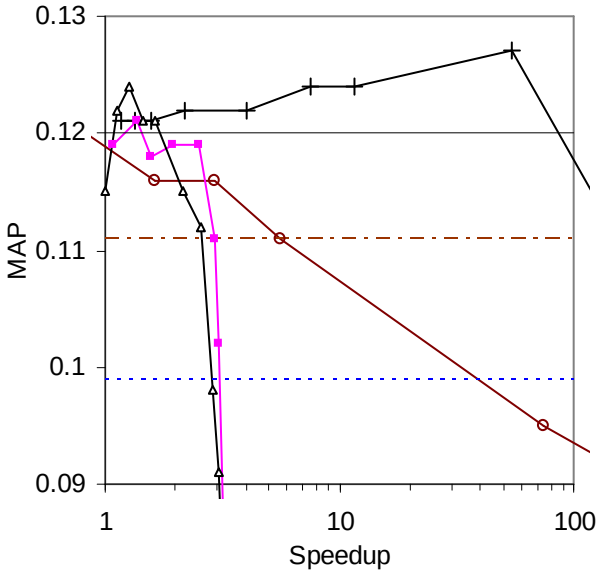
- Data sources
  - Yeast and fly data bases
- Automatically labeled tasks generated from publications
  - Gene recommendation: author, year→gene
  - Reference recommendation: title words,year→paper
  - Expert-finding: title words, genes→author
- Data split
  - 2000 training, 2000 tuning, 2000 test
- Time variant graph (for training)
  - each edge is tagged with a time stamp (year)
  - When doing random walk, only consider edges that are earlier than the query

Example Features

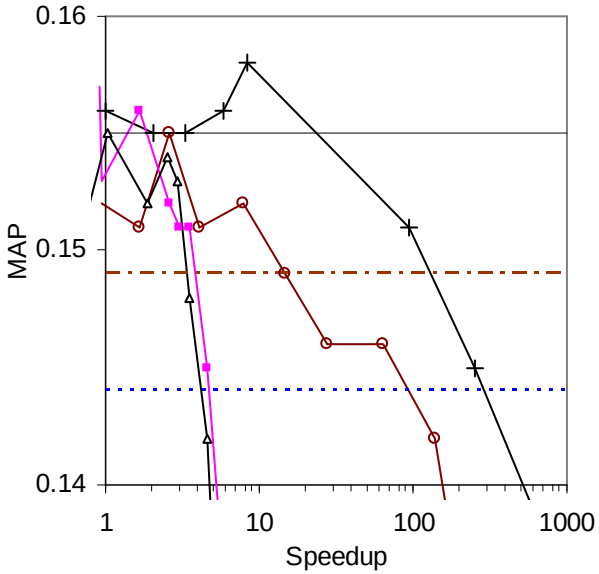
- A model trained for reference recommendation task on the yeast data

ID	Weight	Feature	
1	272.4	$word \rightarrow paper \xrightarrow{Cite^{-1}} paper \xrightarrow{Cite} paper$	1) Papers co-cited with the on-topic papers
2	156.7	$word \rightarrow paper \xrightarrow{Cite} paper$	2) Aggregated citations of the on-topic papers
3	100.5	$gene \rightarrow paper \xrightarrow{Cite^{-1}} paper \xrightarrow{Cite} paper$	
4	83.7	$word \rightarrow paper \xrightarrow{Cite^{-1}} paper$	
5	50.2	$gene \rightarrow paper \xrightarrow{Cite} paper$	
6	41.4	$word \rightarrow paper$	6) Resembles an ad-hoc retrieval system
7	29.3	$year \rightarrow paper \xrightarrow{Cite} paper$	7,8) Papers cited during the past two years
8	13.0	$year \xrightarrow{Before^{-1}} year \rightarrow paper \xrightarrow{Cite} paper$	
	...		
9	3.7	$T^* \rightarrow paper \xrightarrow{Cite} paper$	9) Well cited papers
10	2.9	GAL4>Nature. 1988. GAL4-VP16 is an unusually potent transcriptional activator.	
11	2.1	CYC1>Cell. 1979. Sequence of the gene for iso-1-cytochrome c in Saccharomyces cerevisiae.	10,11) (Important) early papers about specific query terms (genes)
	...		
12	-5.4	$year \xrightarrow{Before^{-1}} year \rightarrow paper$	12,13) General papers published during the past two years
13	-39.1	$year \rightarrow paper$	
14	-49.0	$T^* \rightarrow year \rightarrow paper$	14) old papers

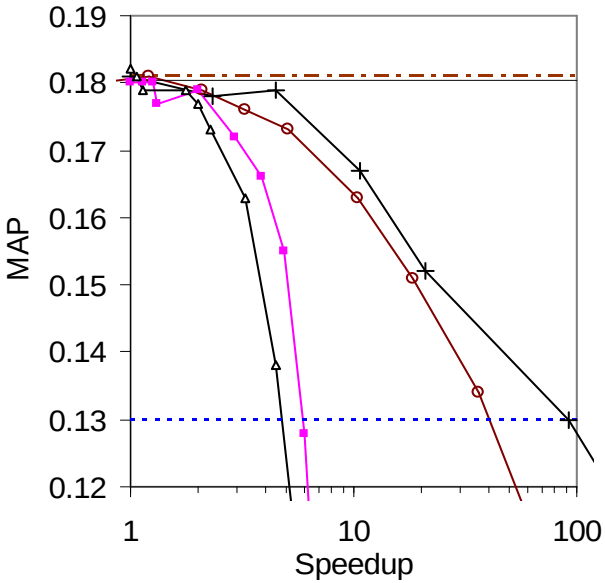
Results on The Yeast Data



T_0 = 0.17s, L= 3



T_0 = 1.6s, L = 4

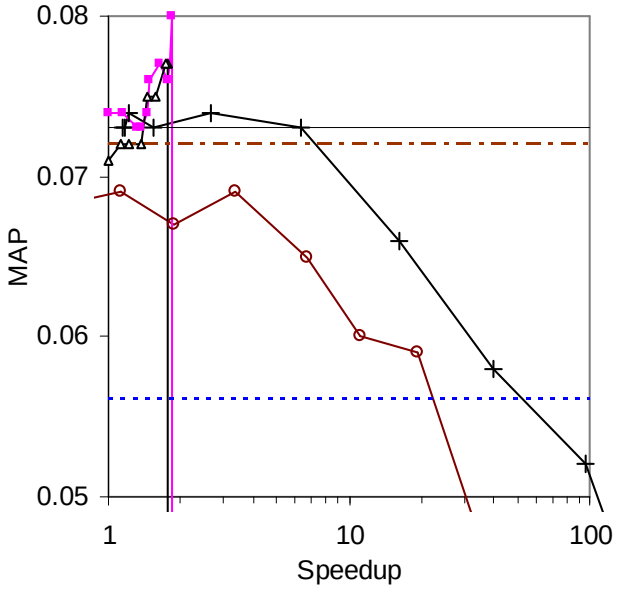


T_0 = 2.7s, L= 3

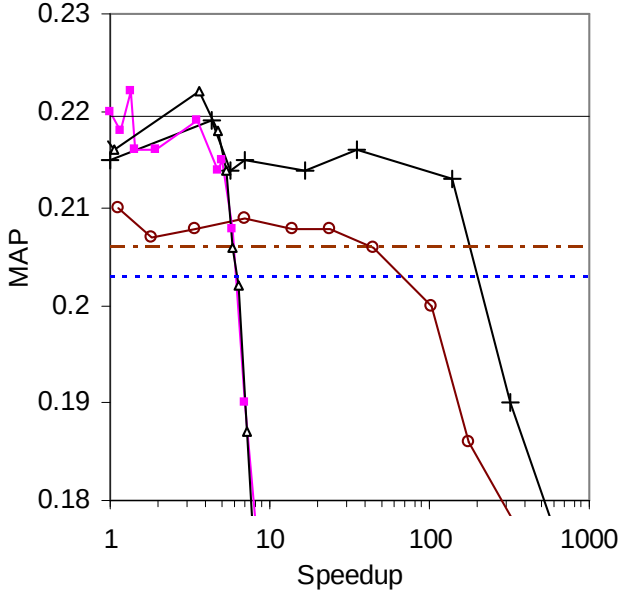
RWR
Particle Filtering
Fixed Truncation
Beam Truncation
Exact
Exact(Edge-Parameter)
Exact(No Learning)

Carnegie Mellon

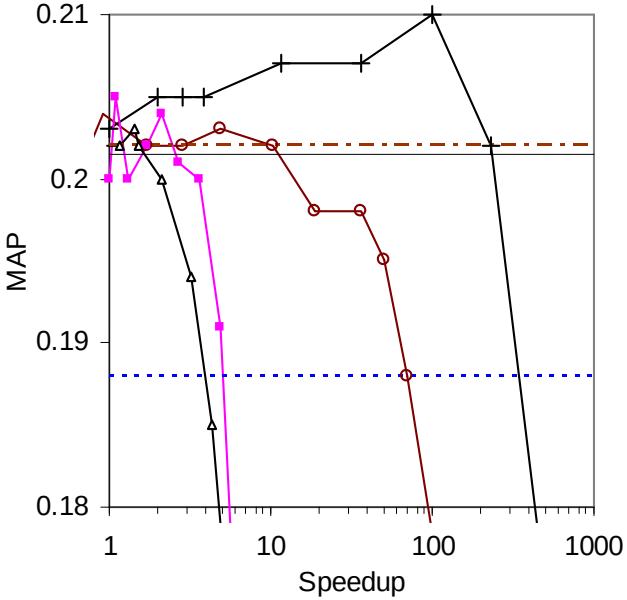
Results on the Fly Data



T_0 = 0.15s, L= 3



T_0 = 1.8s, L= 4



T_0 = 0.9s, L= 3

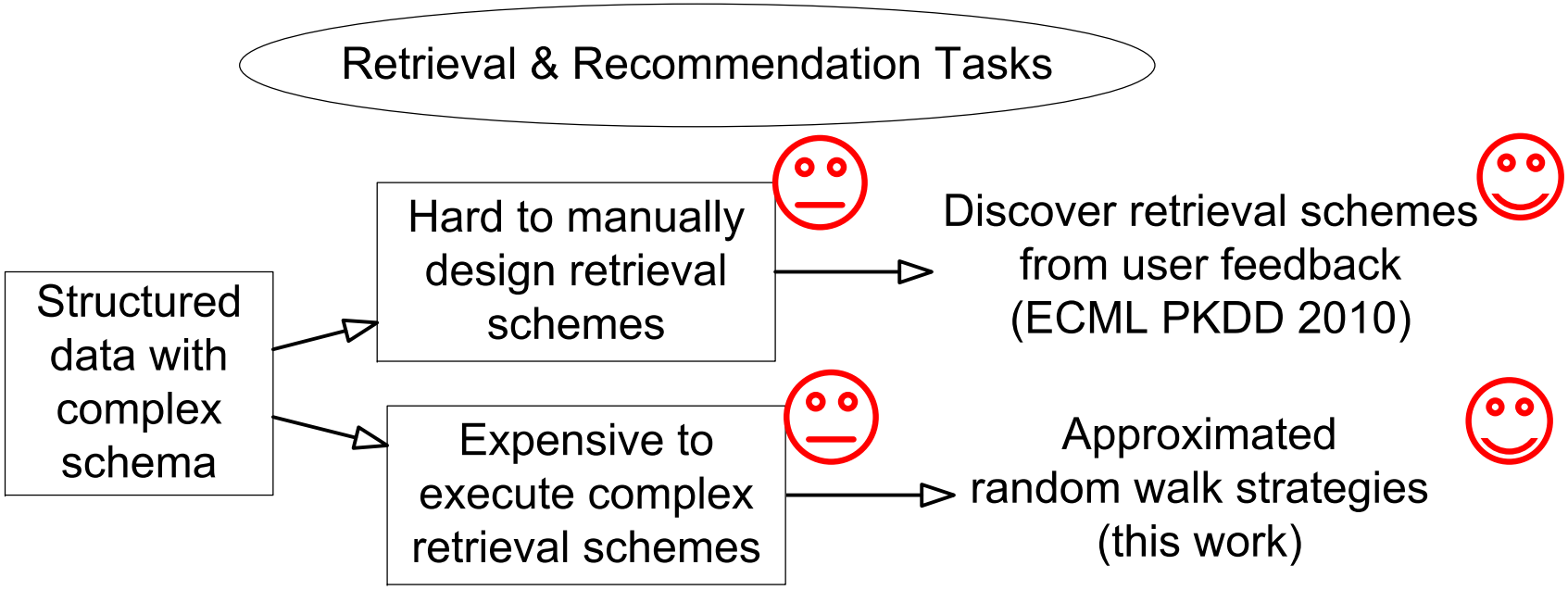
RWR
Particle Filtering
Fixed Truncation
Beam Truncation
Exact
Exact(Edge-Parameter)
Exact(No Learning)

Carnegie Mellon

Observations

- Sampling strategies are more efficient than truncation strategies
  - At each step, the truncation strategies need to generate exact distribution before truncation
- Particle filtering produces better MAP than fingerprinting
  - By reducing the variances of estimations
- Retrieval quality is improved in some cases
  - By producing better weights for the model

Summary

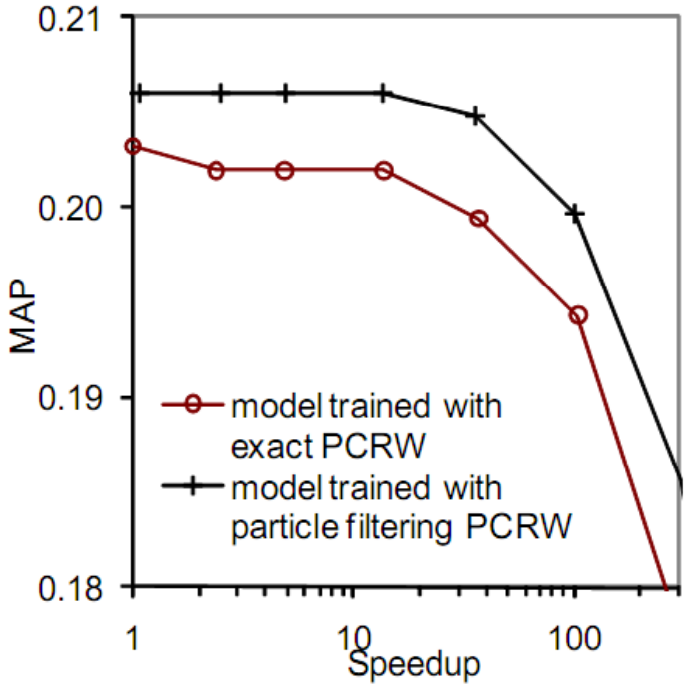

Retrieval & Recommendation Tasks
Structured data with complex schema
Hard to manually design retrieval schemes
Discover retrieval schemes from user feedback (ECML PKDD 2010)
Expensive to execute complex retrieval schemes
Approximated random walk strategies (this work)

The End

- Thanks
- This work is supported by NSF grant IIS-0811562 and NIH grant R01GM081293

Some Evidence

- Compares retrieval qualities with two fixed models but various particle filtering settings at query time



- The models are trained with exact RW and particle filtering ($\varepsilon_{min} = 0.001$) for the reference recommendation task on Yeast data

Some More Evidence

- PF keeps the distributions sparse, and reduces the weights of dense features
  - (Left) shows for each relation path the average number of nodes with non-zero probability
  - (Right) shows for each relation path the ratio of its weight in the particle filtering model vs. its weight in the exact PCRW model. '+' and '−' indicate positive and negative weights