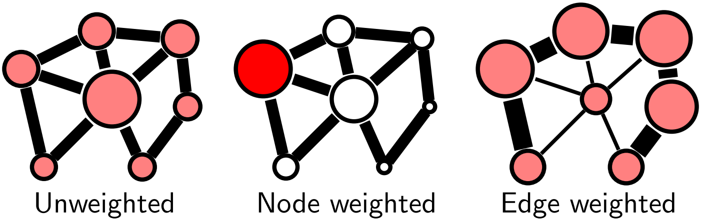
Edge-Weighted Personalized PageRank: Breaking a Decade-Old Performance Barrier

W. Xie **D. Bindel** A. Demers J. Gehrke

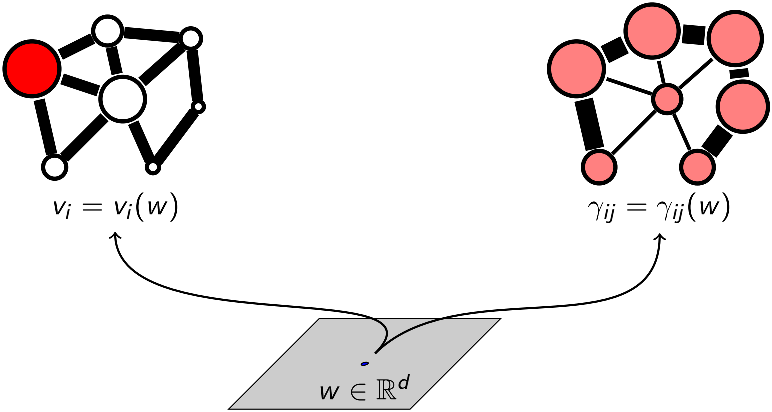
12 Aug 2015

PageRank Model



- Random surfer model: $x^{(t+1)} = \alpha P x^{(t)} + (1 - \alpha)v$ where $P = AD^{-1}$
- Stationary distribution: $Mx = b$ where $M = (I - \alpha P), b = (1 - \alpha)v$

Edge Weight vs Node Weight Personalization



Introduce *personalization parameters* $w \in \mathbb{R}^d$ in two ways:

Node weights: $M\,x(w) = b(w)$

Edge weights: $M(w)\,x(w) = b$

Edge Weight vs Node Weight Personalization

Node weight personalization is well-studied

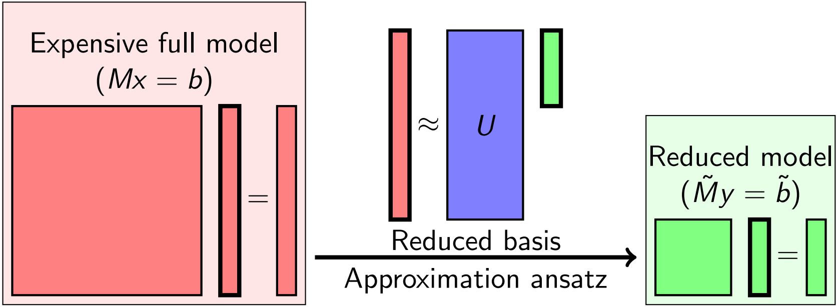
- Topic-sensitive PageRank: fast methods based on linearity
- Localized PageRank: fast methods based on sparsity

Some work on edge weight personalization

- ObjectRank/ScaleRank: personalize weights for different edge types
- But lots of work incorporates edge weights *without* personalization

Our goal: General, fast methods for edge weight personalization

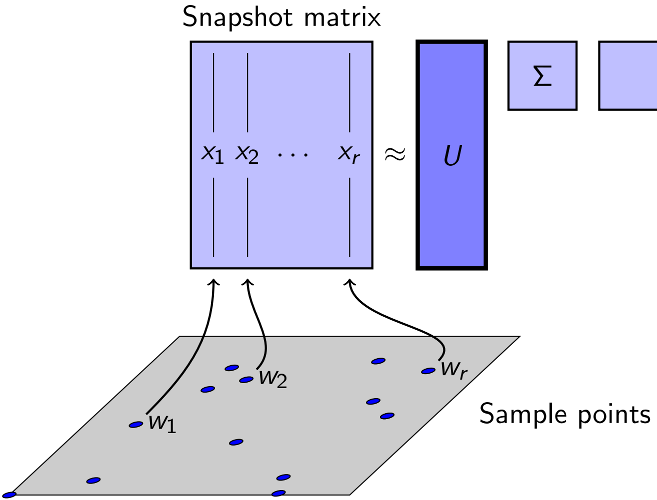
Model Reduction



Model reduction procedure from physical simulation world:

- *Offline*: Construct *reduced basis* $U \in \mathbb{R}^{n \times k}$
- *Offline*: Choose $\geq k$ equations to pick approximation $\hat{x} = Uy$
- *Online*: Solve for $y(w)$ given w and reconstruct $\hat{x}$

Reduced Basis Construction: SVD (aka POD/PCA/KL)



Approximation Ansatz

Want $r = MUy - b \approx 0$. Consider two approximation conditions:

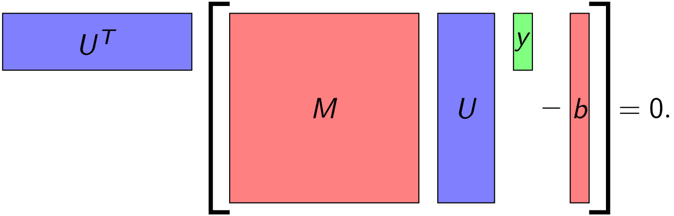
Method	Ansatz	Properties
Bubnov-Galerkin	$U^T r = 0$	Good accuracy empirically Fast for $P(w)$ linear
DEIM	$\min \|r_{\mathcal{I}}\|$	Fast even for nonlinear $P(w)$ Complex cost/accuracy tradeoff

Similar error analysis framework for both (see paper):

$$\text{Consistency} + \text{Stability} = \text{Accuracy}$$

- Consistency: Does the subspace contain good approximants?
- Stability: Is the approximation subproblem far from singular?

Bubnov-Galerkin Method



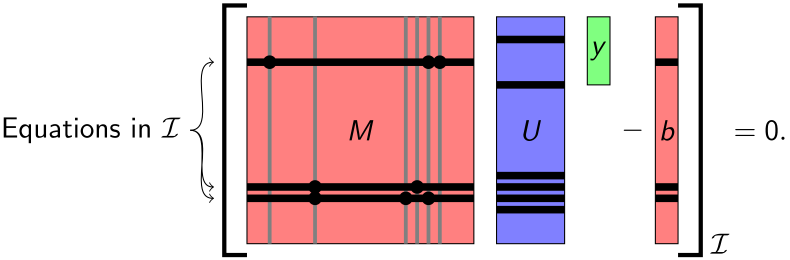
- Linear case: w_i = probability of transition with edge type i

$$M(w) = I - \alpha \left(\sum_i w_i P^{(i)} \right), \qquad \tilde{M}(w) = I - \alpha \left(\sum_i w_i \tilde{P}^{(i)} \right)$$

  where we can precompute $\tilde{P}^{(i)} = U^T P^{(i)} U$

- Nonlinear: Cost to form $\tilde{M}(w)$ comparable to cost of PageRank!

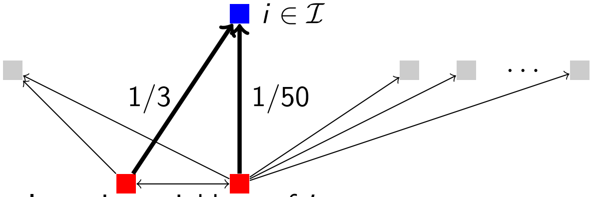
Discrete Empirical Interpolation Method (DEIM)



- Ansatz: Minimize $\|r_{\mathcal{I}}\|$ for chosen indices $\mathcal{I}$
- Only need a few rows of M (and associated rows of U)
- Difference from physics applications: high-degree nodes!

Interpolation Costs

Consider subgraph relevant to one interpolation equation:



- *Really* care about weights of edges incident on $\mathcal{I}$
  - Need more edges to normalize (unless $A(w)$ is linear)
- High in/out degree are expensive but informative
- **Key question**: how to choose $\mathcal{I}$ to balance **cost** vs **accuracy**?

Interpolation Accuracy

- Key: keep $M_{\mathcal{I},:}$ far from singular.
- If $|\mathcal{I}| = k$, this is a *subset selection* over rows of MU.
- Have standard techniques (e.g. pivoted QR)
- Want to pick $\mathcal{I}$ *once*, so look at rows of

$$Z = \begin{bmatrix} M(w_1)U & M(w_2)U & \dots \end{bmatrix}$$

  for sample parameters $w^{(i)}$.
- Helps to explicitly enforce $\sum_i \hat{x}_i = 1$
- Several heuristics for cost/accuracy tradeoff (see paper)

Online Costs

If $\ell = \#$ PR components needed, online costs are:

Form $\tilde{M}$	$O(dk^2)$ for B-G
	More complex for DEIM
Factor $\tilde{M}$	$O(k^3)$
Solve for y	$O(k^2)$
Form Uy	$O(k\ell)$

Online costs **do not** depend on graph size!
(unless you want the whole PR vector)

Example Networks

DBLP (citation network)

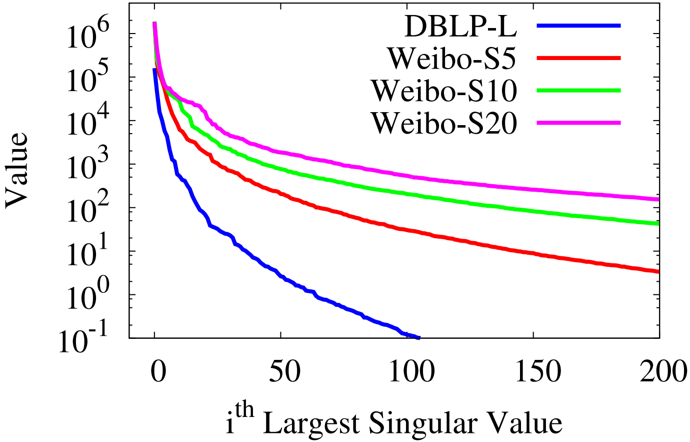
- 3.5M nodes / 18.5M edges
- Seven edge types $\Longrightarrow$ seven parameters
- $P(w)$ linear
- Competition: ScaleRank

Weibo (micro-blogging)

- 1.9M nodes / 50.7M edges
- Weight edges by topical similarity of posts
- Number of parameters = number of topics (5, 10, 20)

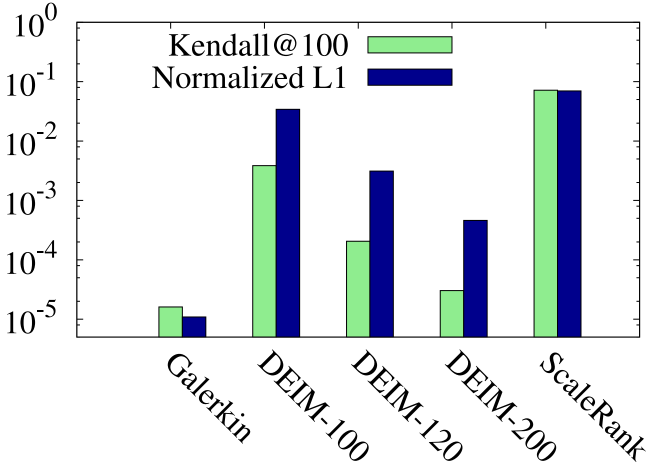
(Studied global and local PageRank – see paper for latter.)

Singular Value Decay

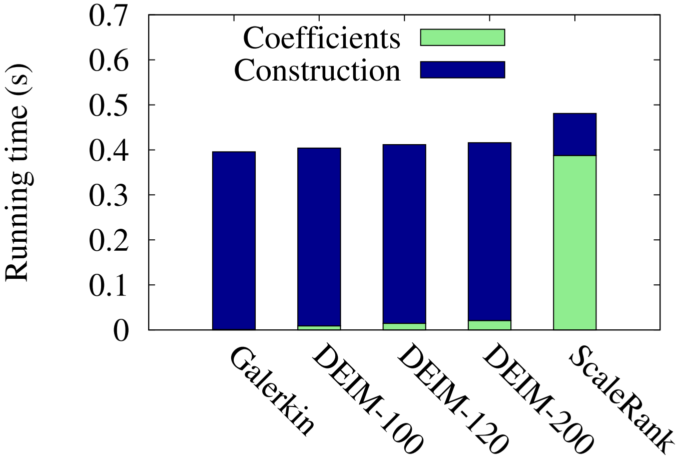


$r = 1000$ samples, $k = 100$

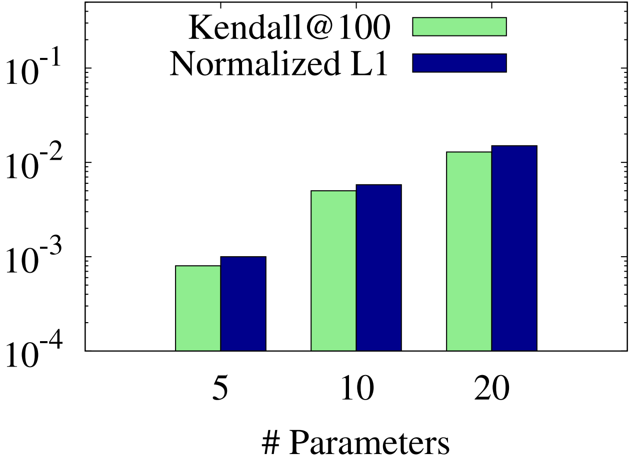
DBLP Accuracy



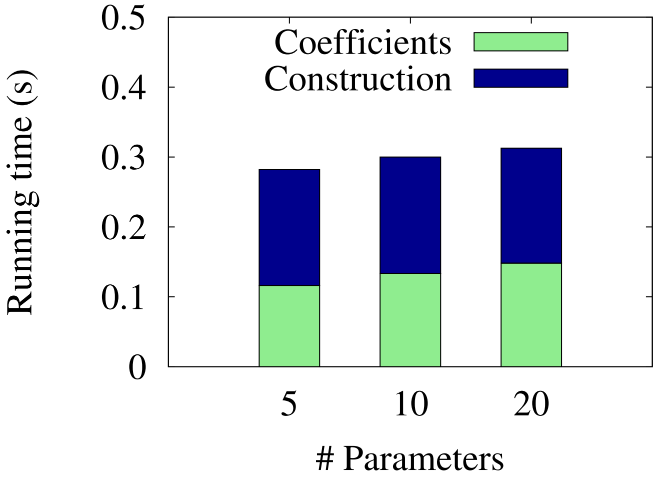
DBLP Running Times (All Nodes)



Weibo Accuracy



Weibo Running Times (All Nodes)

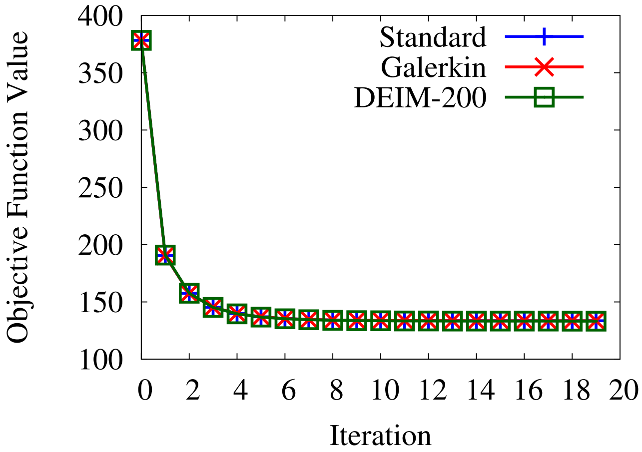


Application: Learning to Rank

Goal: Given $T = \{(i_q, j_q)\}_{q=1}^{|T|}$, find w that mostly ranks i_q over j_1.
(c.f. Backstrom and Leskovec, WSDM 2011)

- Standard: Gradient descent on full problem
  - One PR computation for objective
  - One PR computation for each gradient component
  - Costs $d + 1$ PR computations per step
- With model reduction
  - Rephrase objective in reduced coordinate space
  - Use factorization to solve PR for objective
  - Re-use same factorization for gradient

DBLP Learning Task



(8 papers for training + 7 params)

The Punchline

Test case: DBLP, 3.5M nodes, 18.5M edges, 7 params

Cost per Iteration:

Method	Standard	Bubnov-Galerkin	DEIM-200
Time(sec)	159.3	0.002	0.033

Roads Not Taken

In the paper (but not the talk)

- Selecting interpolation equations for DEIM
- Localized PageRank experiments (Weibo and DBLP)
- Comparison to BCA for localized PageRank
- Quasi-optimality framework for error analysis

Room for future work! Analysis, applications, systems, ...

Questions?

Edge-Weighted Personalized PageRank:
Breaking a Decade-Old Performance Barrier

Wenlei Xie, David Bindel, Johannes Gehrke, and Al Demers

KDD 2015, paper 117

Sponsors:

- NSF (IIS-0911036 and IIS-1012593)
- iAd Project from the National Research Council of Norway