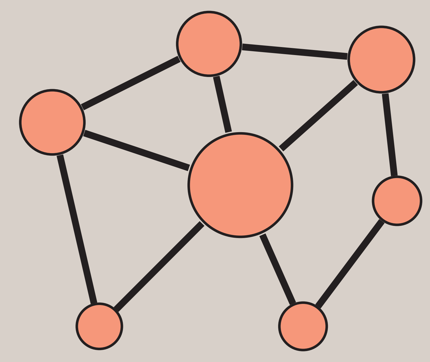
Edge-Weighted Personalized PageRank: Breaking a Decade-Old Performance Barrier

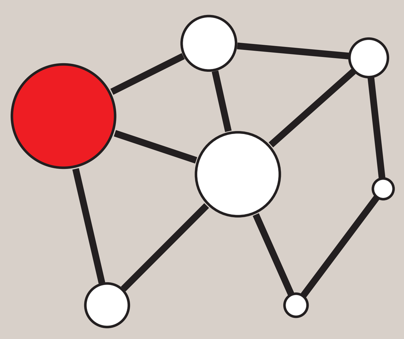
Wenlei Xie David Bindel Alan Demers Johannes Gehrke
Computer Science Department
Cornell University

Weighted PageRank Model

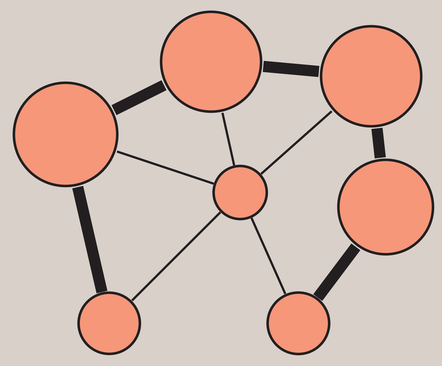
- Model: Random walker with restarts
  $$x^{(t+1)} = \alpha P x^{(t)} + (1-\alpha)v$$
- Stationary equation:
  $$(I - \alpha P)x = (1-\alpha)v$$
- Parameters:
  - Topology and **edge weights** (P)
  - **Restart probability vector** (v)
  - Transition probability α
- Personalize through v or P



Personalizing PageRank: Nodes versus Edges



Node personalization
$(I - \alpha P)x(w) = (1-\alpha)v(w)$



Edge personalization
$(I - \alpha P(w))x(w) = (1-\alpha)v(w)$

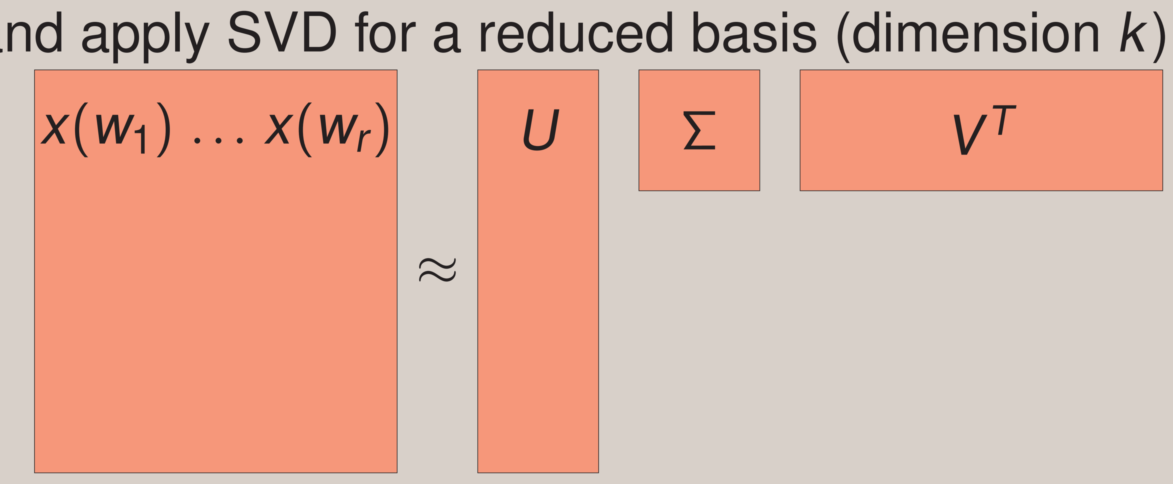
- Personalize PageRank via parameters $w \in \mathbb{R}^d$
- Node personalization: $v = v(w)$
  - Topic-sensitive: $v = Vw$, $V \in \mathbb{R}^{n \times d}$ represents reference topics
  - Ego-centric: v has sparse support (e.g. one node)
  - Linearity or sparsity $\Longrightarrow$ fast methods
- Edge weight personalization: $P = P(w)$
  - Example: typed links, w_i is importance of link type i
  - Even if $P(w)$ is linear, $x(w)$ is nonlinear!

Goal: Fast edge-weighted personalization

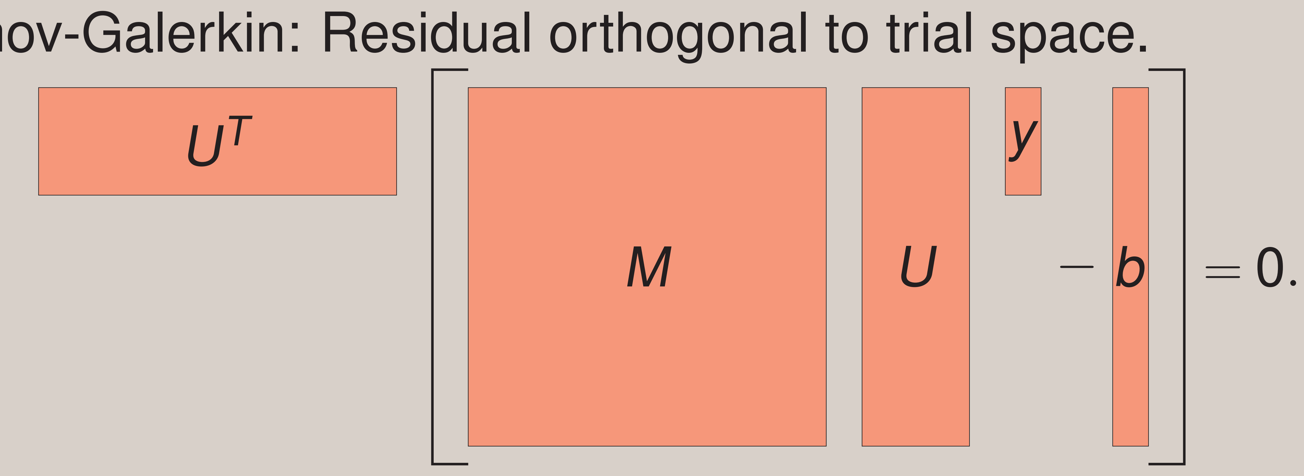
Model Reduction Framework

Idea: Use *model reduction* – common in physical simulations.

- Sample and apply SVD for a reduced basis (dimension k).

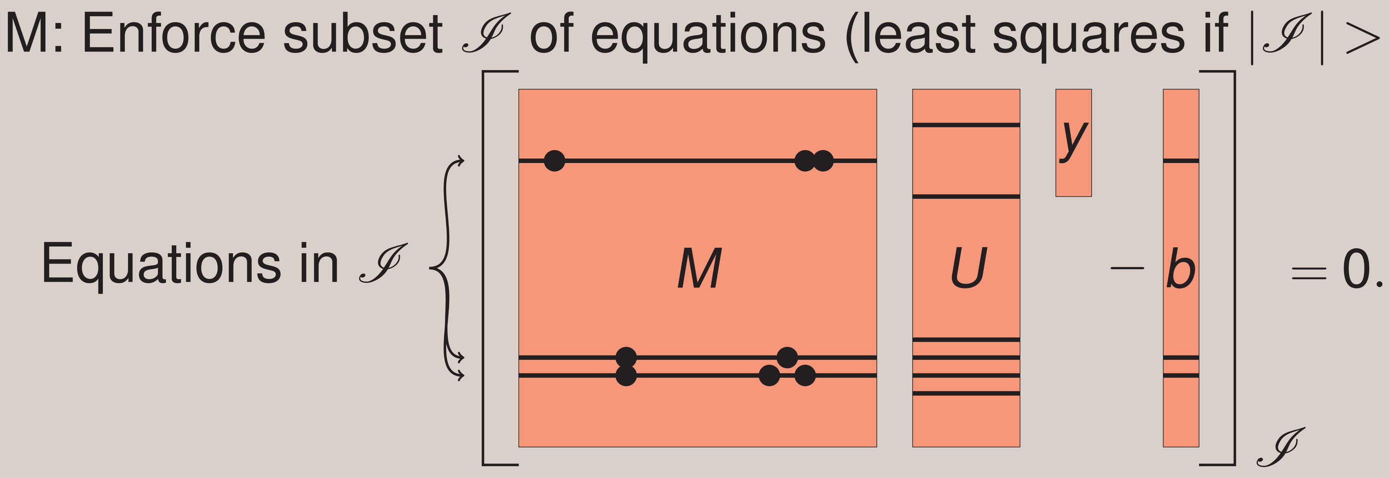


- Approximate $\hat{x}(w) = Uy(w)$ via $\geq k$ equations chosen offline.
  - Bubnov-Galerkin: Residual orthogonal to trial space.



Reduced problem: $y = (U^T M U)^{-1}(U^T b)$.
Pro: Good accuracy.
Con: Expensive to form $U^T M U$ online unless $P(w)$ linear

  - DEIM: Enforce subset $\mathcal{I}$ of equations (least squares if $|\mathcal{I}| > k$).



Reduced problem: $y = (M_{\mathcal{I},:}U)^\dagger b_{\mathcal{I}}$.
Pro: Cheap (?) to form $M_{\mathcal{I},:}U$ for general M.
Con: Choose $\mathcal{I}$ to balance cost vs accuracy (see paper!)

- Reconstruct PageRank vector $\hat{x} = Uy$ (in whole or in part).

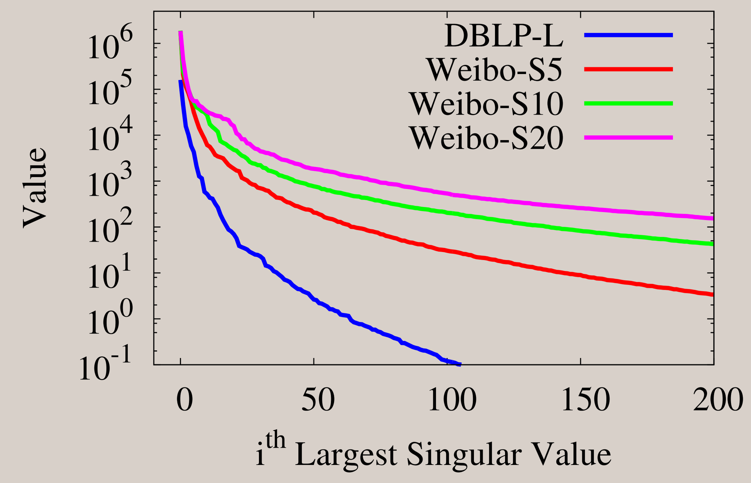
Example Networks

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.)

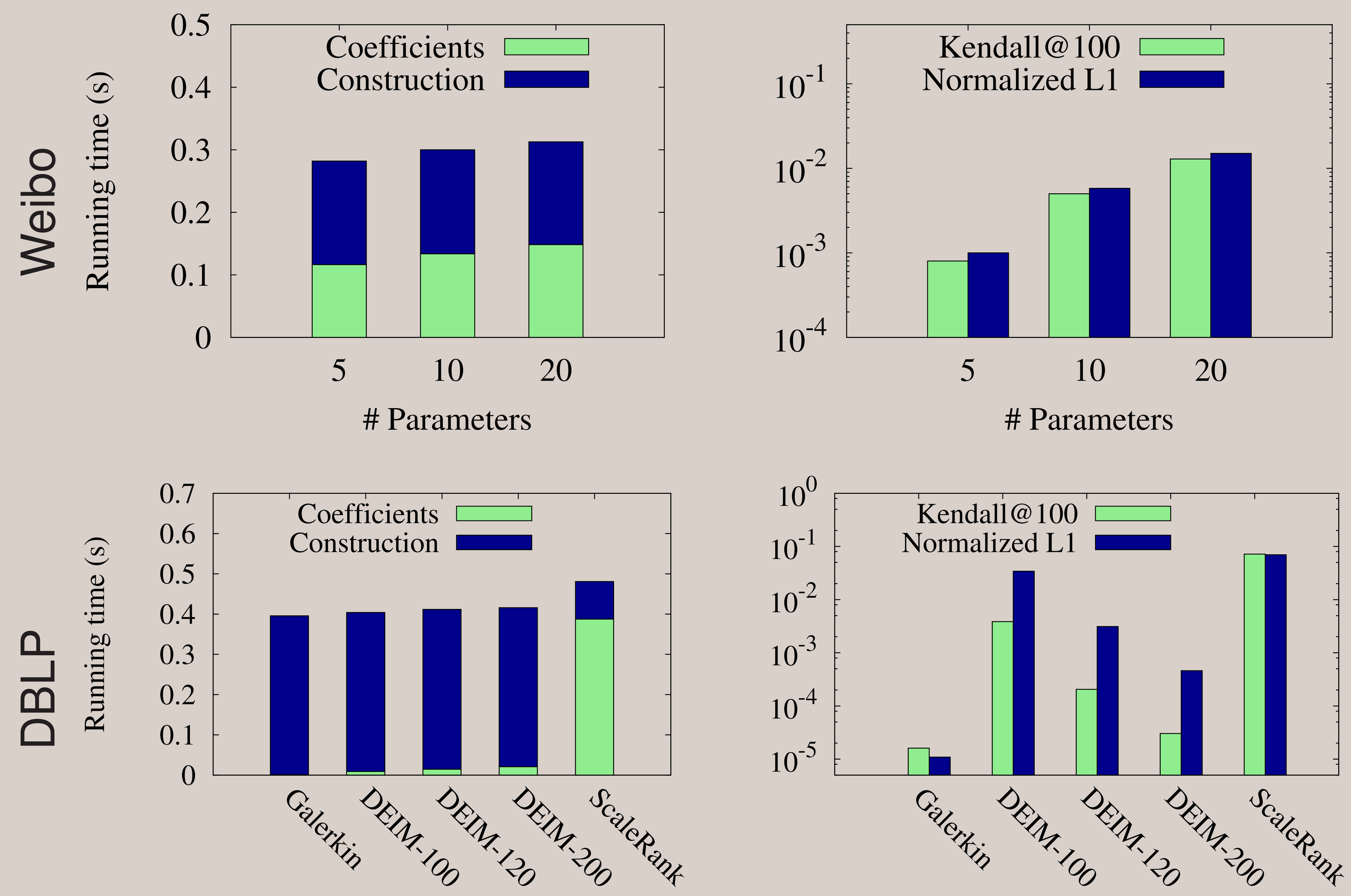
Reduced Model Setup



	DBLP	Weibo
Sample	6 hr	17 hr
SVD	0.8 hr	0.4 hr
$U^T MU$	2 min	N/A
Choose $\mathcal{I}$	11 min	12-18 min

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

Online Runtime and Accuracy

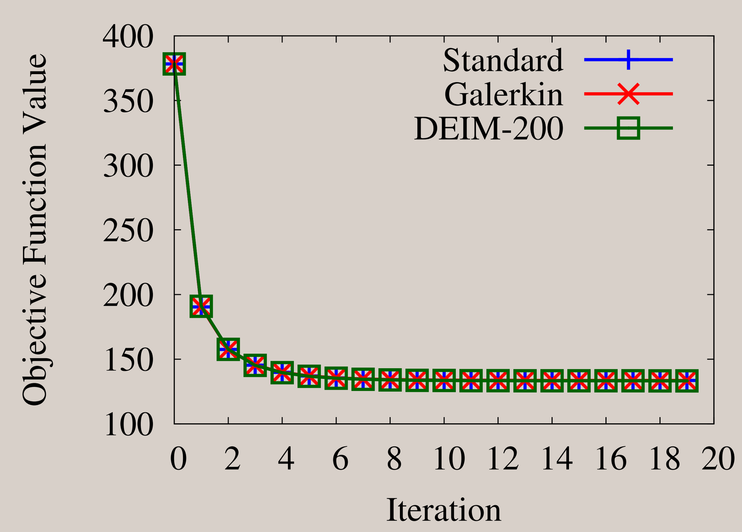


- Weibo: Nonlinear $\Longrightarrow$ only DEIM practical (use $|\mathcal{I}| = 200$)
  - Dominant solve cost: forming $M_{\mathcal{I},:}U$
  - Forming Uy costs more than finding y
- DBLP: Compare Galerkin, DEIM (vary $|\mathcal{I}|$), and ScaleRank
  - Galerkin is fast and accurate (when P linear)
  - Model reduction beats ScaleRank in time and accuracy
- All cases: Construction is time to form *whole* PageRank vector

Application: Learning to Rank

- Training data: $T = (i_q, j_q)$, i_q better than j_1.
- Optimization problem:
  $$\min_w L(w) = \sum_{(i,j) \in T} l(x_i(w) - x_j(w)) + \lambda \|w\|^2.$$
- Cost/step = $d+1$ PageRank systems
  - Objective + d gradient components
  - Model reduction $\Longrightarrow$ most of the work is in the reduced space!
  - Don't even need whole PageRank (just elements used in training)

Results: Learning to Rank (DBLP)



Time per iteration (s)	
Standard	159.3
Bubnov-Galerkin	0.002
DEIM-200	0.033

Model reduction $\Longrightarrow$
Interactive rates