Methods for estimating the diagonal of matrix functions

Jesse Laeuchli
Andreas Stathopoulos

CSC 2016

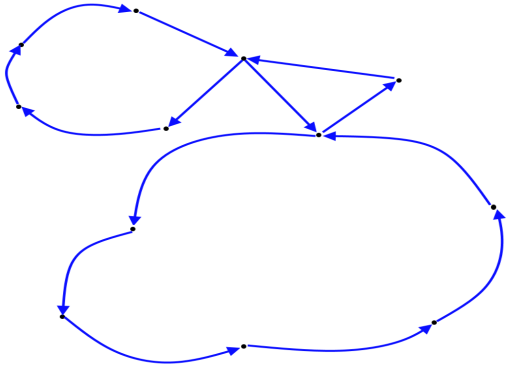
The Problem

Given a large matrix A, and a function f, find $\mathrm{diag(f(A))}$,or $\mathrm{trace(f(A))} = \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{f(A)_{ii}} = \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{f(\lambda_i)}$

- Some important examples of f
  - $f(A) = A^{-1}$
  - $f(A) = A^k$
  - $f(A) = exp(A)$
  - $f(A) = log(A)$

Applications

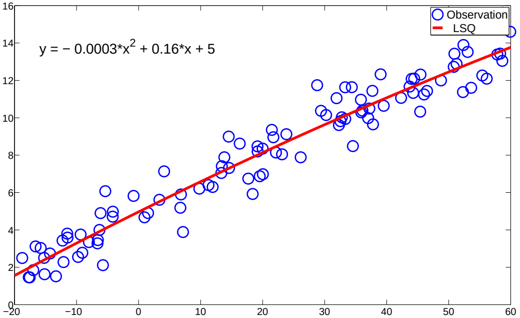
Compute Diagonals of $A^k/\exp(A)$



- Count Triangles/Polygons–Higher distance paths, higher powers of A
- Network Centrality

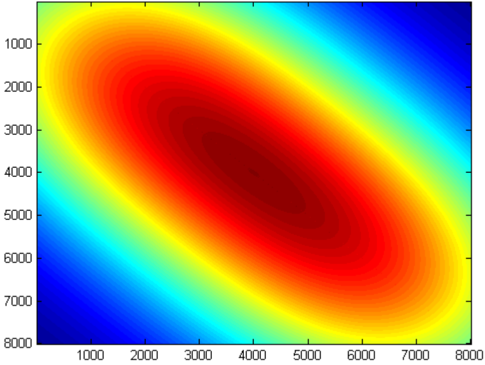
Applications

Compute Diagonals of A^{-1}



- Statistics

Compute Diagonals of A^{-1}



- Uncertainty Quantification

Exact Methods

What if we compute it exactly?

- LU Decomposition
- Eigendecomposition
- Recursive Factorizations
- Takahashi's Equations

All too slow for large matrices

Statistical Trace Estimator

Exact methods infeasible, so we resort to statistical approaches

$$\mathrm{trace}(\mathrm{f}(\mathrm{A})) = \mathbf{E}[\mathrm{z}^{\mathrm{T}}\mathrm{f}(\mathrm{A})\mathrm{z}] \approx \frac{\sum_{\mathrm{i}=0}^{\mathrm{i}=\mathrm{m}} \mathrm{z}_{\mathrm{i}}^{\mathrm{T}}\mathrm{f}(\mathrm{A})\mathrm{z}_{\mathrm{i}}}{\mathrm{m}}$$

$$\mathrm{diag}(\mathrm{f}(\mathrm{A})) = \mathbf{E}[\mathrm{z} \odot \mathrm{f}(\mathrm{A})\mathrm{z}] \approx (\sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{z}_{\mathrm{i}} \odot \mathrm{f}(\mathrm{A})\mathrm{z}_{\mathrm{i}}) \oslash (\sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{z}_{\mathrm{i}} \odot \mathrm{z}_{\mathrm{i}})$$

$$\begin{bmatrix} a \\ b \end{bmatrix} \odot \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} a * c \\ b * d \end{bmatrix}, \begin{bmatrix} a \\ b \end{bmatrix} \oslash \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} a/c \\ b/d \end{bmatrix}$$

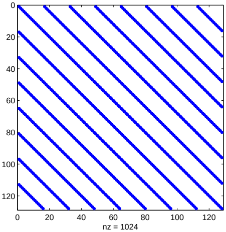
How to pick z? Random

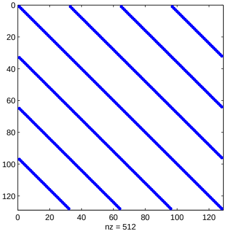
- Random Methods
  - Gaussian$\{z =$ from Gaussian Distribution $\}$
  - Hutchinson$\{z = 1, -1$ probability $1/2$ $\}$
  - Canoncial basis e_i with random i
  - Mixing of diagonals DFT$_n e_i$,Hadamard$_n e_i$ with random i

Estimator	**Variance of the Sample**
Gaussian	$2\|A\|_F^2$
Hutchinson's	$2(\|A\|_F^2 - \sum_{i=1}^n A_{ii}^2)$
Unit Vector	$n\sum_{i=1}^n A_{ii}^2 - \mathrm{trace}^2(\mathrm{A})$

How to pick z? Deterministic

- Deterministic
  - Hadamard$_n e_i$ with $i = 1 : n$

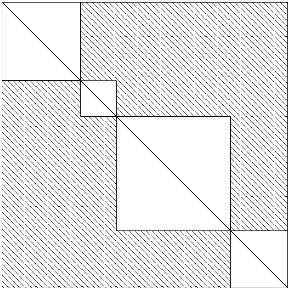




$$H_1 = \begin{bmatrix} 1 \end{bmatrix} \; H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \; H_{2^k} = \begin{bmatrix} H_{2^{k-1}} & H_{2^{k-1}} \\ H_{2^{k-1}} & -H_{2^{k-1}} \end{bmatrix} = H_2 \otimes H_{2^{k-1}}$$

Probing

- Color the associated graph of A, reorder nodes with the same color to be adjacent
- Can then recover the diagonal of an m-colorable matrix with exactly m vectors $x_i^m = \begin{cases} 1, & \text{if } i \in C_m \\ 0, & \text{otherwise.} \end{cases}$
- If we color graph of $f(A)$, then we can recover $\mathrm{diag(f(A))}$



1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
0	1	0	0
0	1	0	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	0	1
0	0	0	1
0	0	0	1

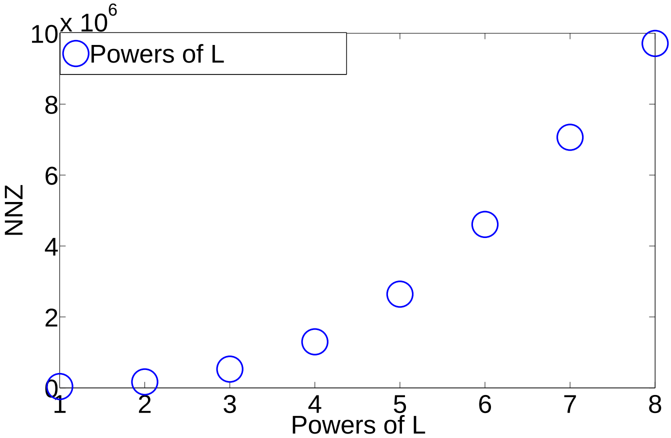
Probing $f(A)$

- Computing $f(A)$ is hard. If we had it, we would be done
- $f(A)$ is dense. It takes too many colors to probe
- Look at structure of polynomial $p_k(A)$ approximating $f(A)$

Problems With Probing-What We Address

Methods for estimating the diagonal of matrix functions

- Even lower order A^k expensive to compute and store



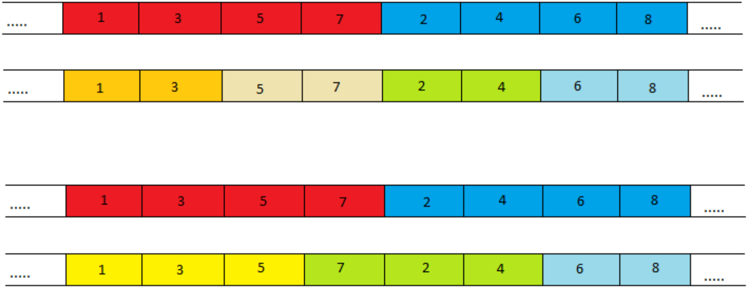
Problems With Probing-What We Address

- Probing vectors for A^k not guaranteed to be a subset of vectors for A^{k+1}

$$\begin{matrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{matrix} \not\subset \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \quad \textbf{but} \quad \begin{matrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{matrix} \subset \begin{matrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

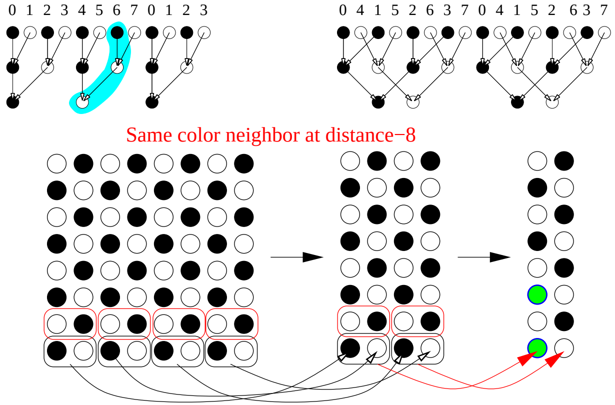
Hierarchical Coloring

- What is Hierarchical Coloring?
  - Colors must be nested
  - Colors split into the same number of new colors
  - Ensures reusability of probing vectors



Hierarchical Coloring via Multilevel

How to color a arbitrary graph? Use multi-level strategy.
Problems if merging is done naively.

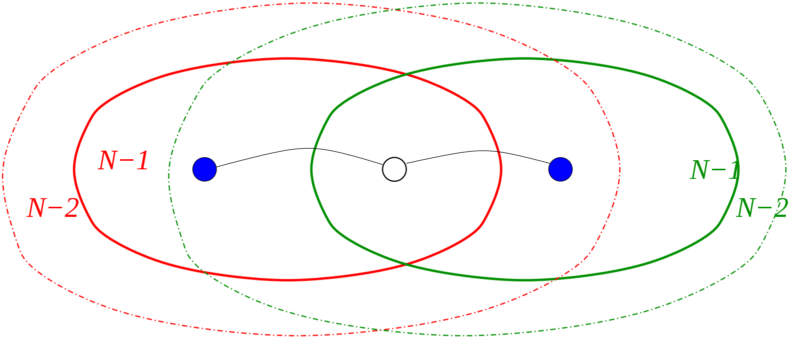


Two green nodes still at distance−2 after 3 levels

Hierarchical Coloring via Multilevel

Methods for estimating the diagonal of matrix functions

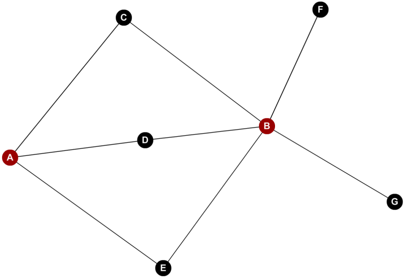
Our strategy, merge distance 1 nodes, and distance 2 neighbourhoods

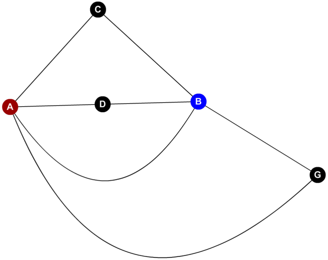


Hierarchical Coloring via Multilevel

Methods for estimating the diagonal of matrix functions

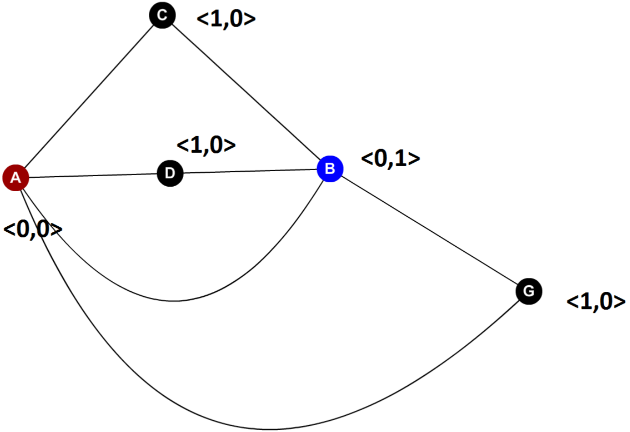
Example Graph. One level of merging





Ensure Hierarchical Coloring

- Create Mixed-Radix Coordinate
- Interpret as color at each level
- Ensures colorings are hierarchical



Statistical Considerations

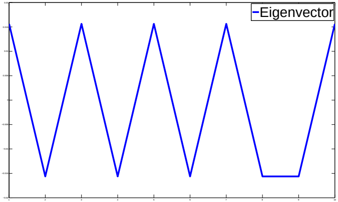
How well should we expect to do with HP?
Pure statistical is $O(\frac{1}{\sqrt{s}})$
Hierarchical Probing depends on structure, g(x) describes fall-off of elements

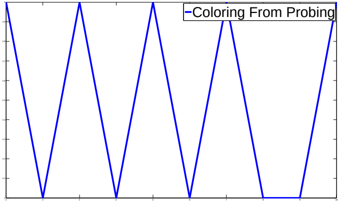
- $g(x) = c$
  - $O(\frac{1}{\sqrt{s}})$
- $g(x) = 1 - x$
  - $O(\frac{1}{s})$

If there is no structure Hierarchical Probing does as well as the statistical methods

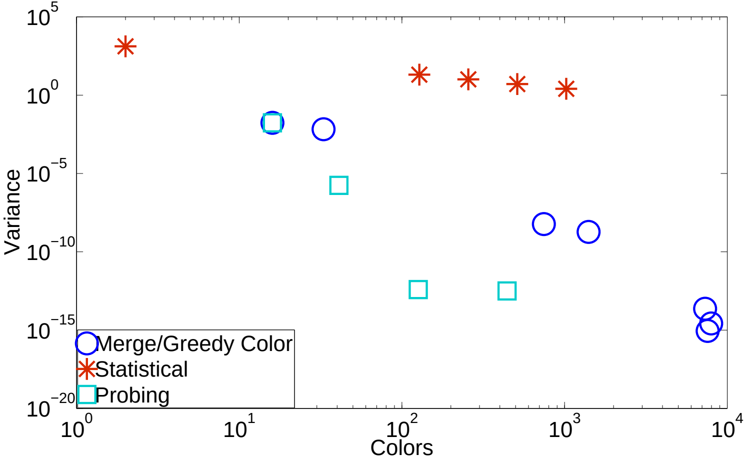
Different Coloring Methods

Top eigenvectors can be used to divide graph into bipartite groups in multiple ways

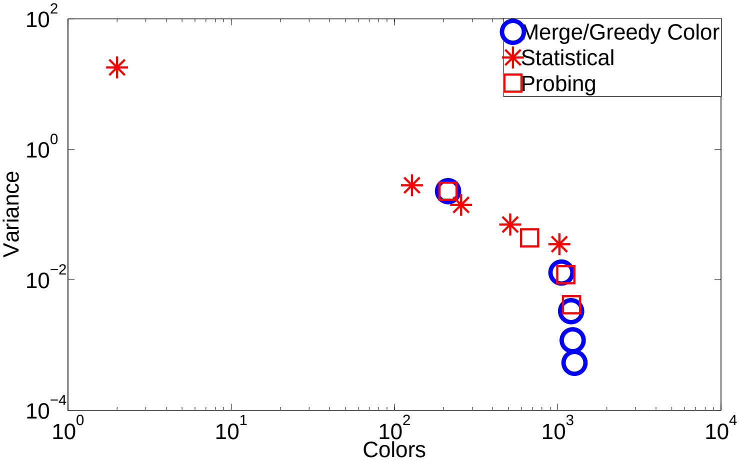




Covariance Matrix

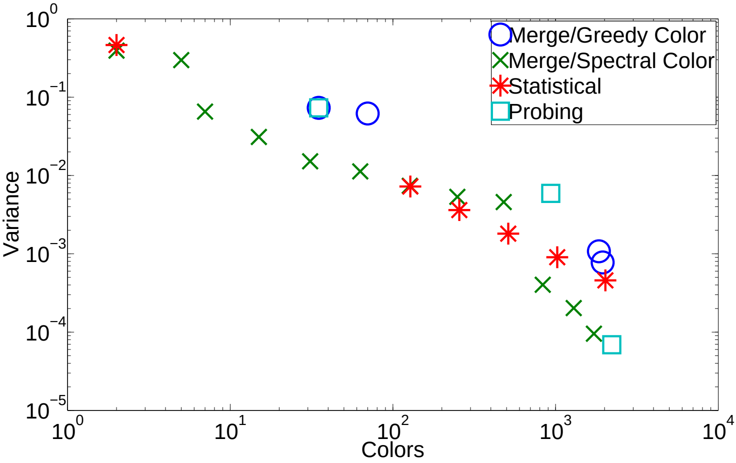


Uncertainty Quantification



Wiki-Vote

Methods for estimating the diagonal of matrix functions



Conclusion

- Exploits structure of matrix
- Statistical bounds
- More efficient than classical probing