Multi-Scale Eigendecomposition

Donghyuk Shin
Dept of Computer Science
UT Austin

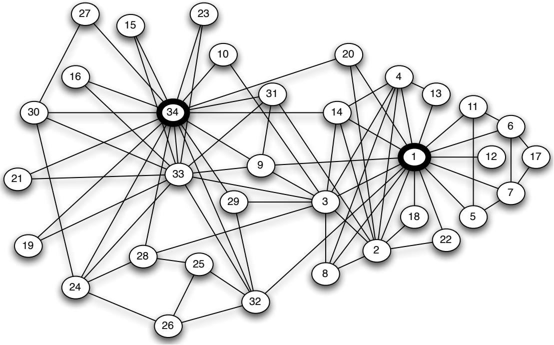
Lab Meeting
Nov 7, 2012

Joint work with S. Si and I. S. Dhillon

Social Networks

- Nodes: individual actors (people or groups of people)
- Edges: relationships (social interactions) between nodes

Example: Karate Club Network:



Low-Rank Approximations

Truncated SVD:

$$A^i \approx V\Lambda^i V^\mathrm{T}$$

allows approximations of matrix functions:

$$\begin{aligned} f(A) &\approx Vf(\Lambda)V^T \\ (I-\beta A)^{-1} &\approx V(I-\beta\Lambda)^{-1}V^T \end{aligned}$$

BUT SVD has Major Drawback. For example, in LiveJournal graph:

- 3.8 million nodes and 65 million edges (avg. degree = 4.5)
- Requires about 1GB of memory to store adjacency matrix
- Rank-100 approximation requires about 5.7GB of memory

PROBLEM: SVD Does Not Preserve SPARSITY

Clustered Low Rank Approximation (CLRA)

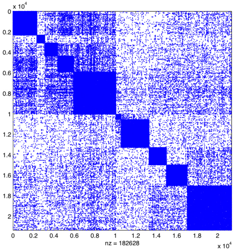
Input: An $m \times m$ adjacency matrix A of a graph, number of clusters c
Output: Clustered low rank approximation of A

1: Cluster the graph into c clusters
2: Compute a low rank approximation of each cluster

$$U_i S_i U_i^T \approx A_{ii}$$

3: Extend the cluster-wise approximations, into an approximation for the entire matrix A

$$S_{ij} = U_i^T A_{ij} U_j$$

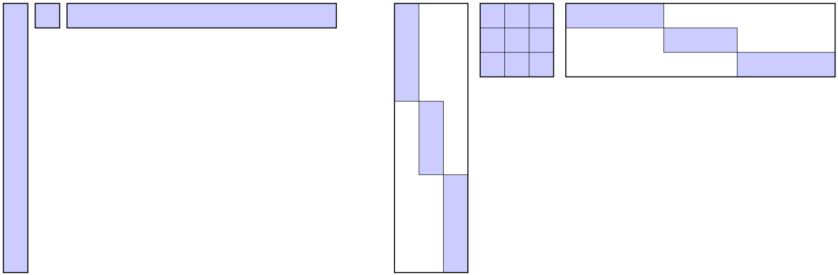


(Savas and Dhillon 2011)

Clustered Low Rank Approximation (CLRA)

Low rank: $A \approx U \Sigma U^{\mathsf{T}}$

Clustered low rank: $A \approx \begin{bmatrix} U_1 & 0 & 0 \\ 0 & U_2 & 0 \\ 0 & 0 & U_3 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \begin{bmatrix} U_1 & 0 & 0 \\ 0 & U_2 & 0 \\ 0 & 0 & U_3 \end{bmatrix}^{\mathsf{T}}$



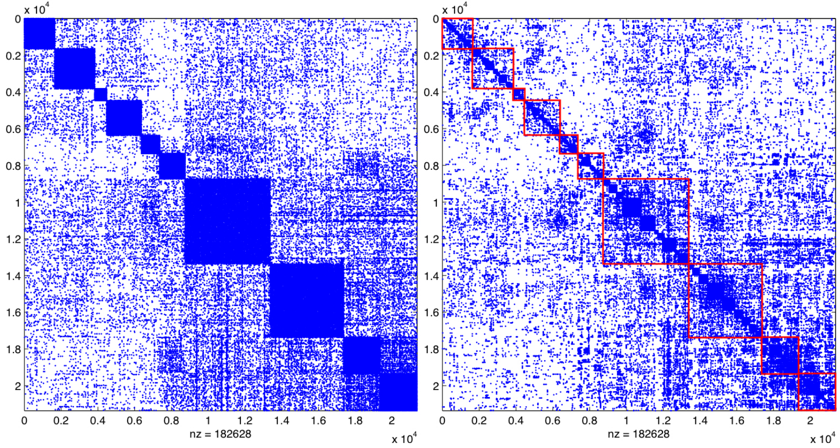
- Observe $\mathrm{diag}(U_1, U_2, \cdots, U_c)$ and has the same memory usage as U
- Matrix S captures the associations between clusters
- Experiments show that most of U is contained in $\mathrm{diag}(U_1, U_2, \cdots, U_c)$

Proposed Method:
Multi-Scale Eigendecomposition

Hierarchical Clustering: arXiv data graph **condMat**

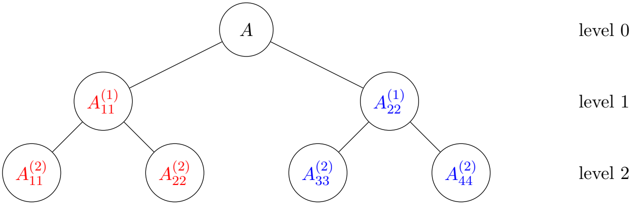
Networks exhibit a hierarchical structure

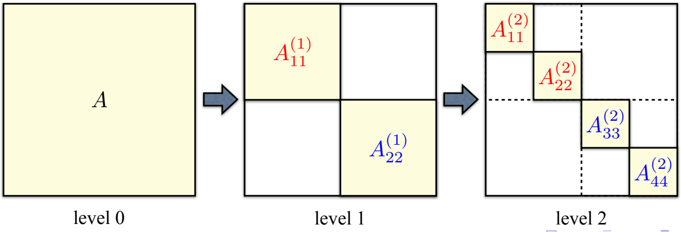
- Exploit hierarchical structure



Hierarchical Clustering

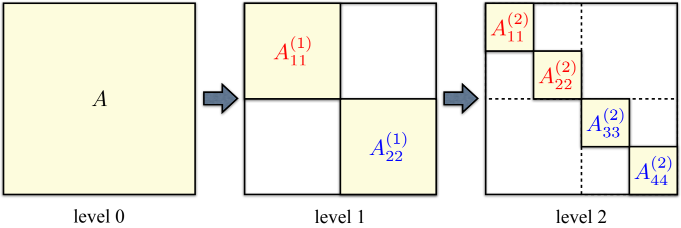
- Top-down approach
- Fast multilevel graph clustering algorithms:
  GRACLUS (Dhillon et al. 2007) and METIS (Karypis and Kumar 1999)





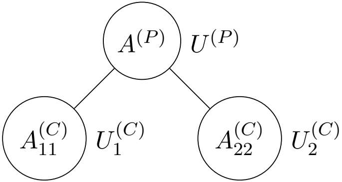
CLRA at Bottom Level

- Compute clustered low-rank approximation at bottom level



$$A \approx \begin{bmatrix} U_1^{(2)} & 0 & 0 & 0 \\ 0 & U_2^{(2)} & 0 & 0 \\ 0 & 0 & U_3^{(2)} & 0 \\ 0 & 0 & 0 & U_4^{(2)} \end{bmatrix} \begin{bmatrix} S_{11}^{(2)} & S_{12}^{(2)} & S_{13}^{(2)} & S_{14}^{(2)} \\ S_{21}^{(2)} & S_{22}^{(2)} & S_{23}^{(2)} & S_{24}^{(2)} \\ S_{31}^{(2)} & S_{32}^{(2)} & S_{33}^{(2)} & S_{34}^{(2)} \\ S_{41}^{(2)} & S_{42}^{(2)} & S_{43}^{(2)} & S_{44}^{(2)} \end{bmatrix} \begin{bmatrix} U_1^{(2)} & 0 & 0 & 0 \\ 0 & U_2^{(2)} & 0 & 0 \\ 0 & 0 & U_3^{(2)} & 0 \\ 0 & 0 & 0 & U_4^{(2)} \end{bmatrix}^T$$

Subspace Approximation



Q: How to get $U^{(P)}$?

A: SVD on $A^{(P)}$

$\longrightarrow$ Computationally expensive as cluster size increases

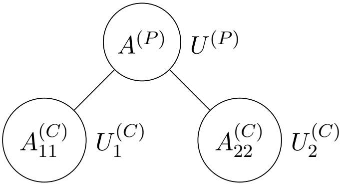
Subspaces between two adjacent levels should be close to each other:

$$U^{(P)} \text{ should be close to } \begin{bmatrix} U_1^{(C)} & 0 \\ 0 & U_2^{(C)} \end{bmatrix}$$

A: Use child cluster's subspace to compute parent cluster's subspace

$\longrightarrow$ Much faster than computing from scratch!

Approximation of Dominant Subspace



1: Capture range space of $A^{(P)}$ by Y:

$$Y = A^{(P)} \begin{bmatrix} U_1^{(C)} & 0 \\ 0 & U_2^{(C)} \end{bmatrix}$$

2: Compute Q as an orthonormal basis for Y

3: $A \approx Q(Q^T A Q)Q^T = QBQ^T \approx Q(V_r \Lambda_r V_r^T)Q^T$

4: $U^{(P)} = QV_r$

Multi-Scale Eigendecomposition

Input: adjacency matrix A, number of levels ℓ, number of clusters c at each node, target rank r.

Output: rank-r eigen-decomposition of $A \approx U \Sigma U^T$.

for $i = 0$ **to** ℓ **do**
    for $j = 1$ **to** c^i **do**
        Cluster $A_{jj}^{(i)}$ into c clusters (e.g. Graclus)
    end for
end for

Compute cluster-wise eigenvectors $U^{(\ell)}$ of $A^{(\ell)}$ (e.g. using `eigs`).
for $i = \ell - 1$ **to** 0 **do**
    Compute $U^{(i)}$ using $U^{(i+1)}$
end for
$\Sigma = {U^{(0)}}^T A U^{(0)}$.
$U = U^{(0)}$.

return U, Σ

Experimental Results

Clustering of Networks

- Networks

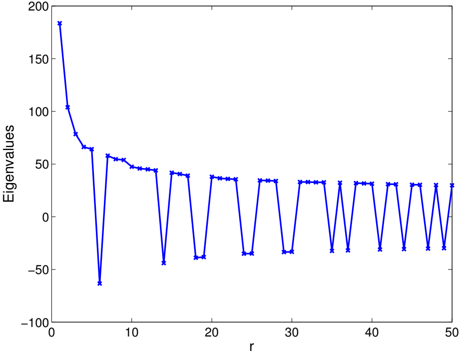
Network	# of nodes	# of edges	# of levels	# of clusters at deepest level
Epinions	32,223	684,026	13	58
MySpace	2,086,141	90,918,158	13	114

- % of within-cluster edges using Graclus vs. random clustering

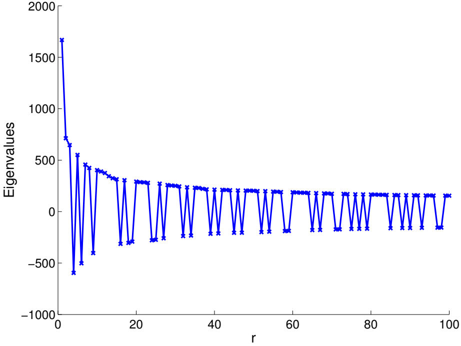
Hierarchy	Epinions	MySpace
Level 1	83.50 (50.18)	98.6 (61.6)
Level 2	66.35 (27.87)	88.0 (35.2)
Level 3	52.02 (14.32)	69.5 (18.0)
Level 4	43.01 (8.91)	64.3 (13.0)
Level 5	35.87 (5.12)	56.3 (8.4)

Eigenvalues of A

- Epinions



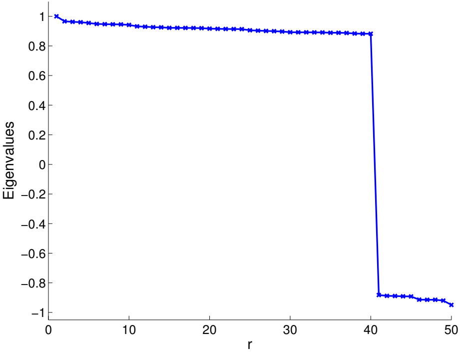
- MySpace



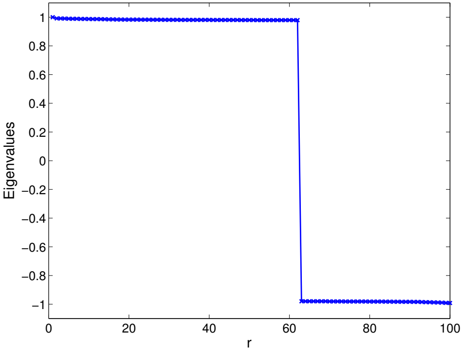
(ordered by magnitude)

Eigenvalues of $D^{-1/2}AD^{-1/2}$

- Epinions



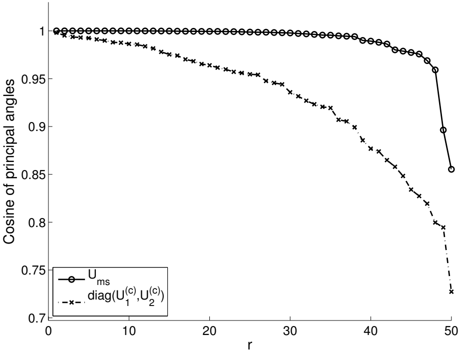
- MySpace



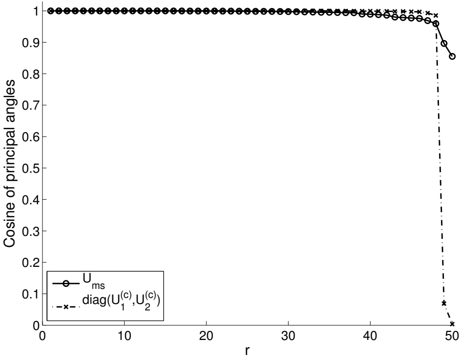
Principal Angles: `Epinions`

Principal angles between U computed by `eigs`

- A



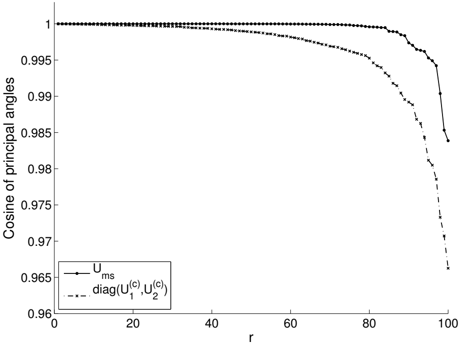
- $D^{-1/2}AD^{-1/2}$



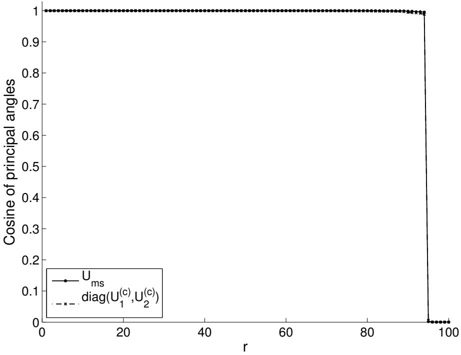
Principal Angles: MySpace

Principal angles between U computed by eigs

- A



- $D^{-1/2}AD^{-1/2}$



Computation Time

- A

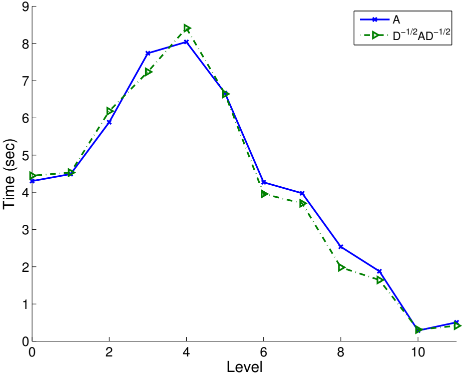
Network		Epinions	MySpace	MySpace
Rank		50	100	200
eigs		81.21	8208.59	32450.40
MSLP	Intermediate levels	50.57	5881.68	13170.26
	Bottom level	93.73	8804.68	16045.80
MSLP Total		144.30	14686.36	29216.06

- $D^{-1/2}AD^{-1/2}$

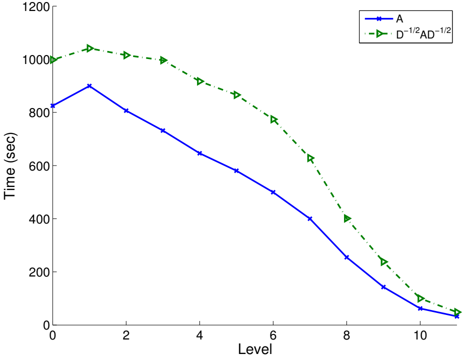
Network		Epinions	MySpace
Rank		50	100
eigs		182.77	56685.60
MSLP	Intermediate levels	49.47	8023.53
	Bottom level	96.12	16907.05
MSLP Total		145.59	24930.58

Computation time of intermediate levels

- Epinions



- MySpace



References

[1] I. S. Dhillon, Y. Guan and B. Kulis. *Weighted Graph Cuts without Eigenvectors: A Multilevel Approach*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007.

[2] B. Savas and I. S. Dhillon. *Clustered Low Rank Approximation of Graphs in Information Science Applications*. SDM, 2011.

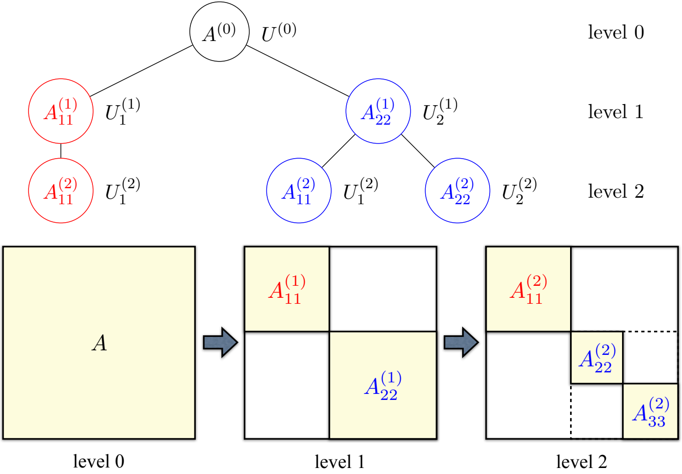
[3] H. Song, B. Savas, T. Cho, V. Dave, Z. Lu, I. S. Dhillon, Y. Zhang and L. Qiu. *Clustered Embedding of Massive Social Networks*. SIGMETRICS, 2012.

[4] D. Shin, S. Si and I. S. Dhillon. *Multi-Scale Link Prediction*. CIKM, 2012.

[5] X. Sui, T. Lee, J. Whang, B. Savas, S. Jain, K. Pingali and I. S. Dhillon. *Parallel Clustered Low-rank Approximation of Graphs and Its Application to Link Prediction*. LCPC, 2012.

Unbalanced Tree

- Divide the graph into smaller subgraphs via hierarchical clustering
- $A_{11}^{(1)} = A_{11}^{(2)}$



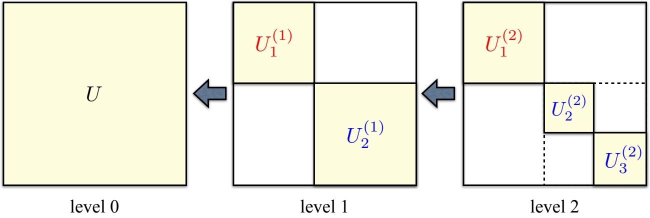
Clustered Low Rank Approximation

- Compute clustered low-rank approximation at bottom level

$$A \approx \begin{bmatrix} U_1 & 0 & 0 \\ 0 & U_2 & 0 \\ 0 & 0 & U_3 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \begin{bmatrix} U_1 & 0 & 0 \\ 0 & U_2 & 0 \\ 0 & 0 & U_3 \end{bmatrix}^\mathsf{T}$$

  where $S_{ij} = U_i^\mathsf{T} A_{ij} U_j$ for $i \neq j$

- Obtain low-rank approximation for higher levels using lower level's approximation



Much faster than computing from scratch!