Multi-Scale Spectral Decomposition of Massive Graphs

Si Si*[1], Donghyuk Shin*[1], Inderjit S. Dhillon[1] and Beresford N. Parlett[2]

[1]Department of Computer Science, University of Texas at Austin [2]Department of Mathematics, University of California, Berkeley

Spectral Decomposition of Graphs

GOAL: Given a graph $\mathcal{G}$ and its $n \times n$ adjacency matrix A, we seek to compute a rank-k approximation with reasonably large k (say $k \sim 10^2$ or 10^3):

$$A \approx U_k \Lambda_k U_k^\mathsf{T} = \sum_{j=1}^{k} \lambda_j \boldsymbol{u}_j \boldsymbol{u}_j^\mathsf{T}$$

- $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_k|$ are the leading eigenvalues of A,
- $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k$ are the corresponding eigenvectors of A.

A key operation needed in numerous machine learning applications:

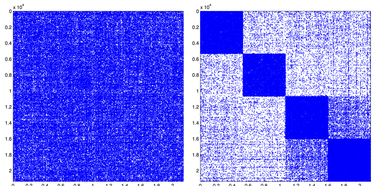
- Label propagation for semi-supervised and multi-label learning
- Recommender systems with side-information
- Link prediction in social network analysis
- Densest k-subgraph problem, ...

Standard methods:

- Power method — restricted to the leading eigenvector
- Single-vector Lanczos — difficulty with multiplicities of eigenvalues
- Block versions — slow convergence due to random initialization
  - Randomized SVD (Subspace Iteration) and Block Lanczos

Cluster Structure of Graphs

- Graphs exhibit cluster structure.



(a) Original graph (b) Clustered graph

$$A = D + \Delta$$

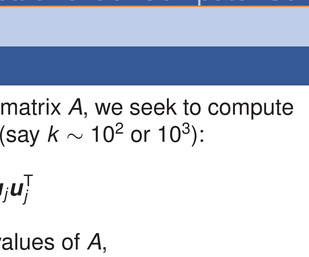
$$D = \begin{bmatrix} A_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A_{cc} \end{bmatrix}, \quad \Delta = \begin{bmatrix} 0 & \cdots & A_{1c} \\ \vdots & \ddots & \vdots \\ A_{c1} & \cdots & 0 \end{bmatrix}$$

CondMat graph (21K nodes and 182K edges) — with 4 clusters, more than 85% of edges appear within clusters.

Overlap between Subspaces of A and D

- **OBSERVATION**: Union of all cluster's subspaces Ω has significant overlap with the dominant subspace U_k of the original graph.

$$\begin{bmatrix} \vdots \\ U_k \\ \vdots \end{bmatrix} \overset{\text{overlap}}{\longleftrightarrow} \begin{bmatrix} U_{k_1}^{(1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & U_{k_c}^{(c)} \end{bmatrix} = \Omega$$



THEOREM 1. Let U_k be the 'true' eigenvectors of A and let $\Omega = U_{k_1}^{(1)} \oplus U_{k_2}^{(2)} \oplus \cdots \oplus U_{k_c}^{(c)}$, where $U_{k_i}^{(i)}$ are the k_i dominant eigenvectors of A_{ii}. For some $\eta \geq 0$, the principal angles $\Theta(\Omega, U_k)$ between Ω and U_k are related to Δ as

$$\|\sin(\Theta(\Omega, U_k))\|_2 \leq \frac{\|\Delta\|_2}{\eta}, \quad \|\sin(\Theta(\Omega, U_k))\|_F \leq \frac{\sqrt{k}\|\Delta\|_F}{\eta}.$$

- We want to find a partition of A such that $\|\Delta\|$ is small in order to make $\|\sin(\Theta(\Omega, U_k))\|$ small $\Rightarrow$ graph clustering.

Multi-Scale Spectral Decomposition (MSEIGS) with Single Level

MAIN IDEA: Use Ω as a good initialization to standard eigensolvers.

ALGORITHM 1: SINGLE-LEVEL MSEIGS

1. Generate c clusters $A_{11}, \cdots, A_{cc}$ via graph clustering.
2. Compute top-r eigenvectors $U_r^{(i)}$ of A_{ii} using standard eigensolvers ($r \leq k$).
3. Select top-k eigenvectors from the c clusters.
4. Form block diagonal matrix $\Omega = U_{k_1}^{(1)} \oplus \cdots \oplus U_{k_c}^{(c)}$ ($\sum_i k_i = k$).
5. Apply block Lanczos with initialization $Q_1 = \Omega$ and compute U_k and Λ_k.

Analysis of Single-level MSEIGS

- Good initialization is important for block Lanczos.

THEOREM 2. Let $A \approx \bar{U}_k \bar{\Lambda}_k \bar{U}_k^\mathsf{T}$ be the rank-k approximation computed by Algorithm 1. The approximation error can be bounded as

$$\|A - \bar{U}_k \bar{\Lambda}_k \bar{U}_k^\mathsf{T}\|_2 \leq 2\|A - A_k\|_2 \left(\frac{\|\Delta\|_2^2}{\eta^2 - \|\Delta\|_2^2} \right)^{\frac{1}{2(q+1)}},$$

where A_k is the best rank-k approximation of A and q is the number of iterations of block Lanczos.

THEOREM 3. Let $\bar{\lambda}_1 \geq \cdots \geq \bar{\lambda}_k$ be the approximate eigenvalues obtained after q steps of block Lanczos in Algorithm 1.

$$\lambda_i \leq \bar{\lambda}_i \leq \lambda_i + \frac{(\lambda_1 - \lambda_i)\|\Delta\|_2^2}{T_{q-1}^2(\frac{1+\nu_i}{1-\nu_i})(\eta^2 - \|\Delta\|_2^2)},$$

where $\nu_i = (\lambda_i - \lambda_{k+1})/(\lambda_i - \lambda_1)$ and $T_m(x)$ is the m-th Chebyshev polynomial.

- Clusters can be processed independently – naturally parallelizable
- Larger clusters have more influence – assign rank k_i to cluster i based on $\|A_{ii}\|_F / \sum_i \|A_{ii}\|_F$.
- Oversample eigenpairs from clusters (e.g., $1.2k$).
- Better cluster quality (small $\|\Delta\|$) implies higher accuracy.

Vertices shuffled (%)	0	20	40	60	80	100
Within-cluster edges (%)	86.31	64.57	47.08	35.43	27.42	24.92
Avg. $\cos(\Theta(\bar{U}_k, U_k))$	0.9980	0.9757	0.9668	0.9475	0.9375	0.9268

Trade-off in the Number of Clusters c

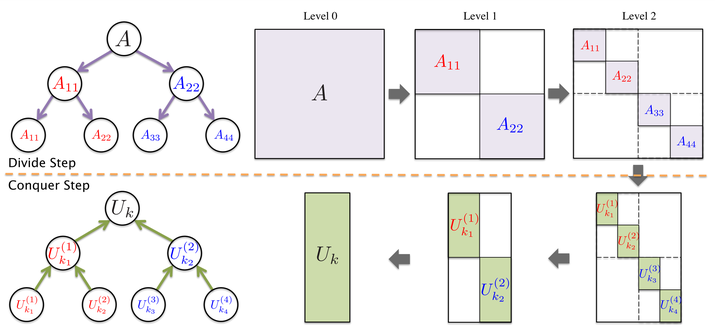
- Small c will give:
  - Larger clusters — increases time to compute eigenpairs of A_{ii}
  - Smaller $\|\Delta\|$ — faster convergence of single-level MSEIGS
- Large c will give:
  - Smaller clusters — faster to compute eigenpairs of A_{ii}
  - Larger $\|\Delta\|$ — slower convergence of single-level MSEIGS

$\Rightarrow$ Use hierarchical clustering

Multi-Scale Spectral Decomposition (MSEIGS)

ALGORITHM 2: MSEIGS

1. Construct hierarchical clustering of graph.
2. Compute approximation at the bottom level (leaf clusters).
3. Compute approximation of parent cluster using subspaces of child clusters.
4. Top level (entire graph) outputs U_k and Λ_k of A.



Early-Termination Strategy (MSEIGS-Early)

- Computing the exact spectral decomposition can be quite time consuming.
- Highly accurate eigenpairs are not essential for many applications.

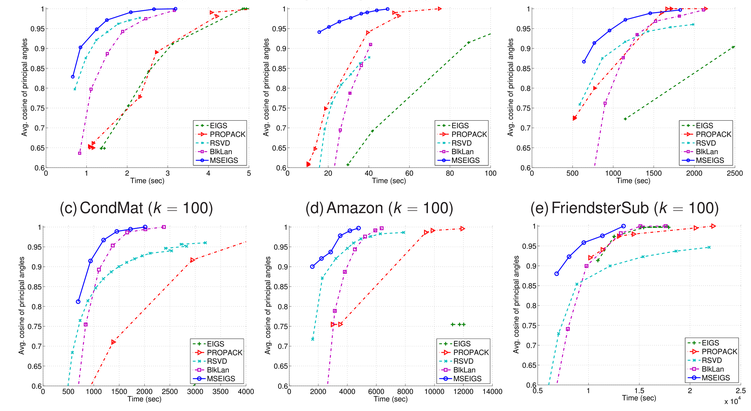
Fast early termination strategy (**MSEIGS-EARLY**):

- Prematurely terminate MSEIGS at a certain level ℓ.
- Select the top-k eigenpairs from all c_ℓ clusters as an approximation of A.

Approximation Results

Dataset	CondMat	Amazon	RoadCA	LiveJournal	Friendster	SDWeb
#nodes	21K	335K	2.00M	4.00M	10.00M	82.29M
#nonzeros	183K	1.85M	5.53M	69.36M	83.67M	3.68B

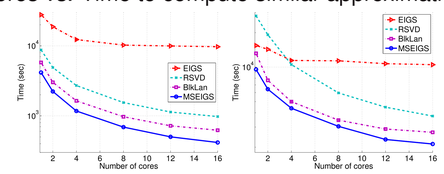
- Time vs. Average cosine of principal angles:



(c) CondMat ($k = 100$) (d) Amazon ($k = 100$) (e) FriendsterSub ($k = 100$)

(f) RoadCA ($k = 200$) (g) LiveJournal ($k = 500$) (h) SDWeb ($k = 50$)

- Number of cores vs. Time to compute similar approximations:



(i) LiveJournal (j) SDWeb

Application Results

- Label propagation in semi-supervised learning: $F^{(t+1)} = \alpha S F^{(t)} + (1-\alpha) Y$

$$F^* = (1-\alpha)(I - \alpha S)^{-1} Y \approx (1-\alpha) U_k (I - \alpha \Lambda_k)^{-1} U_k^\mathsf{T} Y$$

S is the normalized affinity matrix; Y is the $n \times l$ initial label matrix.

Method	Aloi (k = 1500, #classes= 1000)		Delicious (k = 1000, #labels= 983)		
	time(sec)	acc(%)	time(sec)	top3-acc(%)	top1-acc(%)
Truncated	1824.8	59.87	3385.1	45.12	48.89
CG	2921.6	60.01	1094.9	44.93	48.73
EIGS	3890.9	60.08	458.2	45.11	48.51
RSVD	964.1	59.62	359.8	44.11	46.91
BlkLan	1272.2	59.96	395.6	43.52	45.53
MSEIGS	**767.1**	60.03	**235.6**	44.84	49.23
MSEIGS-Early	**176.2**	58.98	**61.36**	44.71	48.22

- Inductive Matrix Completion (IMC) for recommender systems:

$$\min_{W \in \mathbb{R}^{f_c \times r}, H \in \mathbb{R}^{f_d \times r}} \sum_{(i,j) \in \Omega} (R_{ij} - \boldsymbol{x}_i^\mathsf{T} W H^\mathsf{T} \boldsymbol{y}_j)^2 + \frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2),$$

R is the user-item matrix; $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ are user and item features, respectively. Principal components of a social network are used as user features.

Method	Flixster (k = 100)		Amazon (k = 500)		LiveJournal (k = 500)	
	eig-time	recall@20	eig-time	recall@20	eig-time	recall@20
Katz	-	0.1119	-	0.3224	-	0.2838
MC	-	0.0820	-	0.4497	-	0.2699
EIGS	120.51	0.1472	871.30	0.4999	12099.57	0.4259
RSVD	85.31	0.1491	369.82	0.4875	7617.98	0.4294
BlkLan	104.95	0.1465	882.58	0.4687	5099.79	0.4248
MSEIGS	**36.27**	0.1489	**264.47**	0.4911	**2863.55**	0.4253
MSEIGS-Early	**21.88**	0.1481	**179.04**	0.4644	**1545.52**	0.4246

Conclusions

- MSEIGS: a novel divide-and-conquer based framework for approximating the top-k spectral decomposition of large-scale graphs.
- Exploits cluster structure of the graph — easily parallelizable.
- Future work: dealing with graphs that cannot fit into memory.

*Equal contribution to the work.

Email: {ssi,dshin,inderjit}@cs.utexas.edu, parlett@math.berkeley.edu

NIPS 2014