Multi-Scale Spectral Decomposition of Massive Graphs

Donghyuk Shin
Department of Computer Science
UT-Austin

Group Meeting
Nov 11, 2014

Joint work with Si Si, Inderjit S. Dhillon and Beresford N. Parlett

Introduction

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):

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

where

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

Applications

Spectral decomposition is one of the most informative and fundamental matrix approximations — 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
- Spectral clustering
- Graph matching
- Densest k-subgraph problem
- Principal Component Analysis
- ...

Spectral Decomposition

Standard methods:

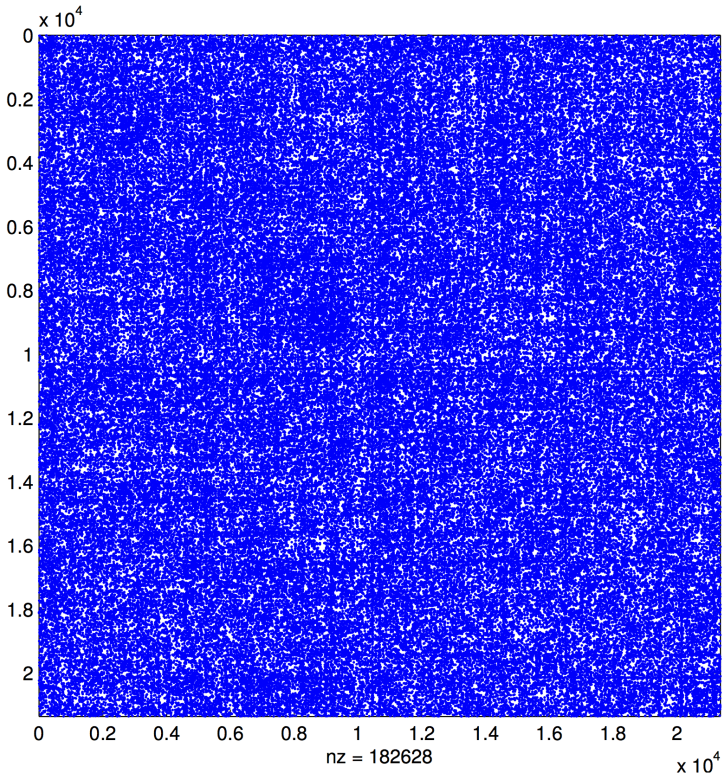
- Power method — restricted to the leading eigenvector
- Single-vector Lanczos — difficulty with multiplicities of eigenvalues
- Block versions — slow convergence due to random initialization
  - Block Lanczos
  - Randomized SVD (Subspace Iteration)

Software: ARPACK (Matlab's `eigs` function), PROPACK, SVDPACK, SLEPc, etc

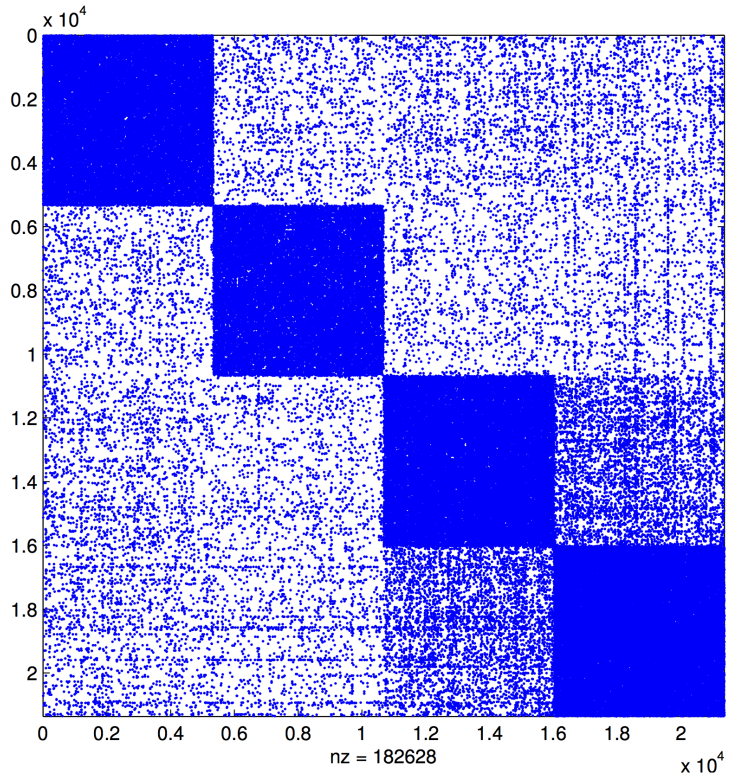
Motivation

Cluster Structure

Graphs exhibit cluster structure.
CondMat dataset: 21,362 nodes and 182,628 edges.



(a) Original graph



(b) Clustered graph

More than 85% of edges appear within clusters.

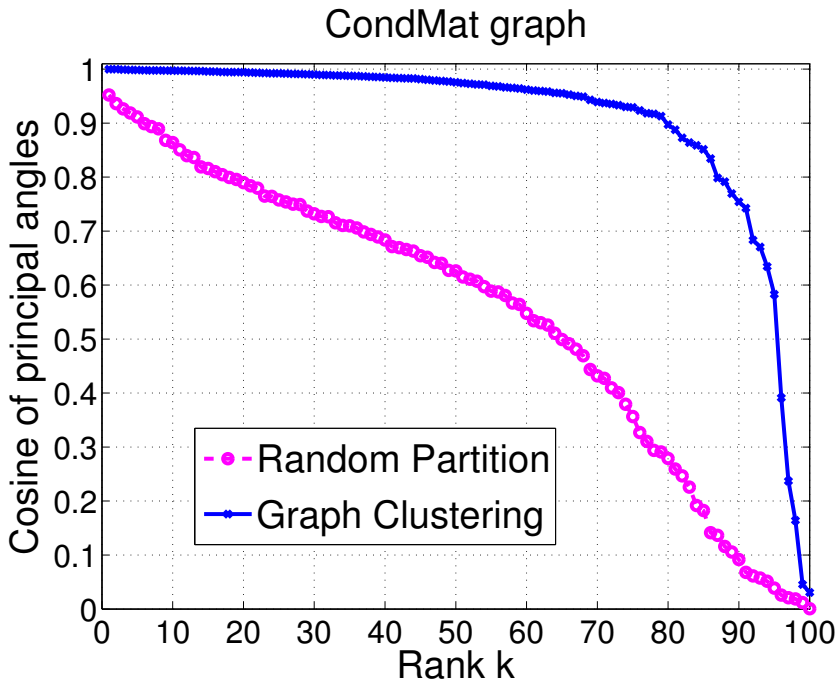
Subspace of Clusters

Graph clustering vs. Random partition

$$A = D + \Delta$$

$$D = \begin{bmatrix} A_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A_{cc} \end{bmatrix}$$

$$\Delta = \begin{bmatrix} 0 & \cdots & A_{1c} \\ \vdots & \ddots & \vdots \\ A_{c1} & \cdots & 0 \end{bmatrix}$$



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

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

Principal Angles

Theorem

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

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

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

Fast graph clustering algorithms: Metis, Graclus, GEM, Nerstrand.

Proposed Method: Multi-Scale Spectral Decomposition

Single-level MSEIGS

Main Idea

Use Ω obtained from the clusters of A as a good initialization to standard eigensolvers.

Multi-Scale Spectral Decomposition (MSEIGS) with single level:

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

Approximation Error

Theorem

Let $A \approx \bar{U}_k \bar{\Lambda}_k \bar{U}_k^{\mathsf{T}}$ be the rank-k approximation computed by our method. 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 for block Lanczos.

Good initialization is important for block Lanczos.

Remarks on Single-level MSEIGS

- Approximation of clusters can be computed independently — naturally parallelizable
- Larger clusters have more influence — assign rank k_i to cluster i based on $\|A_{ii}\|_F / \sum_i \|A_{ii}\|_F$.
- Use oversampling for the number of eigenpairs r from child clusters (e.g., $1.2k$).
- Convergence check is time consuming — test convergence after a few iterations of block Lanczos.
- 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

Number of Clusters

Trade-off in choosing the number of clusters c:

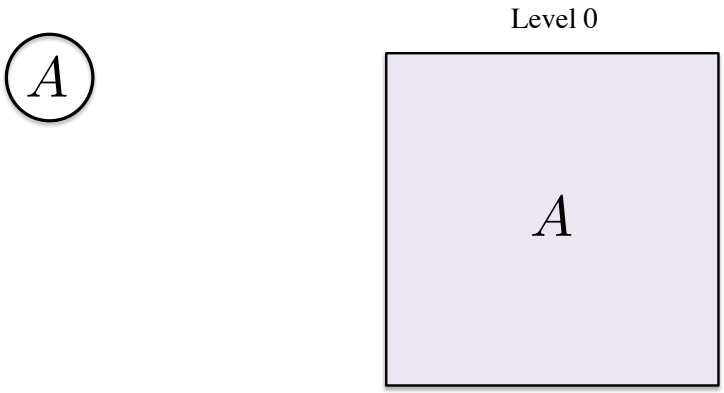
- Small c will give:
  - Larger clusters — increases time to compute eigenpairs of A_{ii}
  - Smaller $\|\Delta\|$ — faster convergence of single-level MSEIGS
- Large c will give:
  - Smaller clusters — faster to compute eigenpairs of A_{ii}
  - Larger $\|\Delta\|$ — slower convergence of single-level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

$\Rightarrow$ Use hierarchical clustering:

- Further partition A_{ii} into smaller clusters until each cluster is small enough.
- Recursively apply single-level MSEIGS from lower to upper levels.

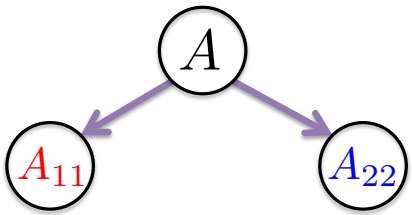
Multi-Scale Spectral Decomposition (MSEIGS)

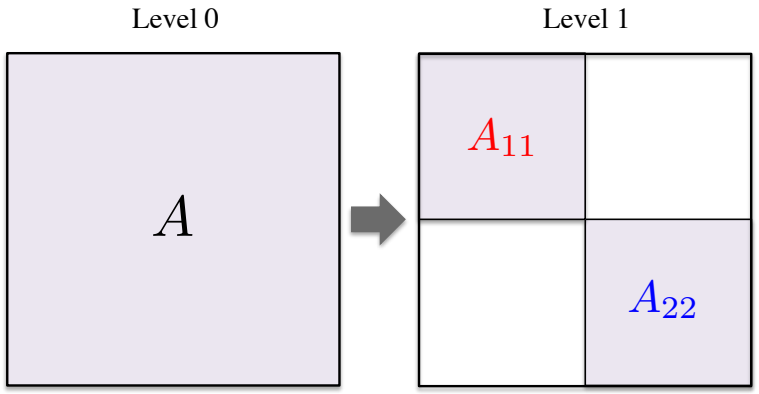
1. Construct hierarchical clustering of A.
2.
3.
4.



Multi-Scale Spectral Decomposition (MSEIGS)

1. Construct hierarchical clustering of A.
2.
3.
4.

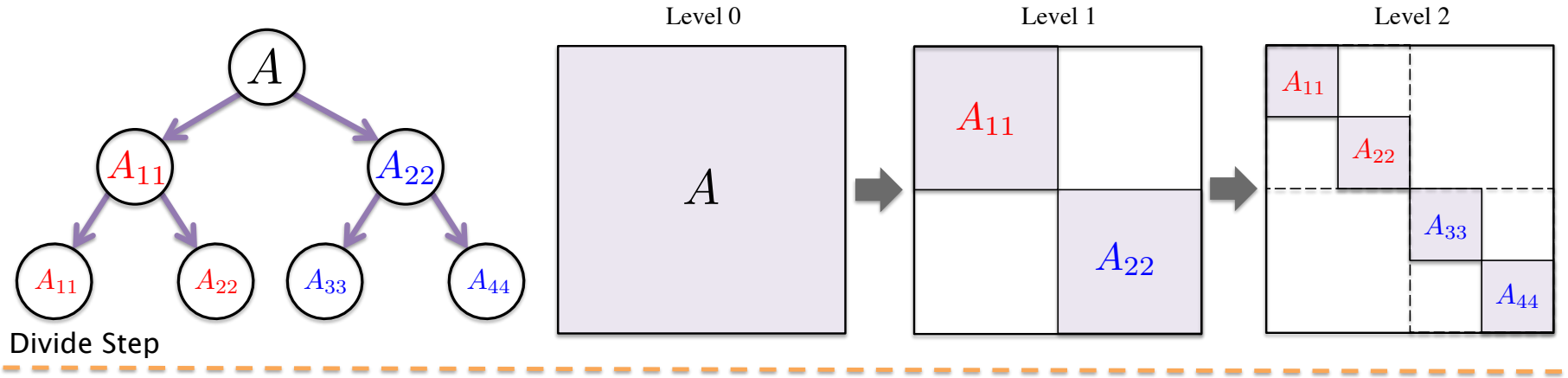




Divide Step

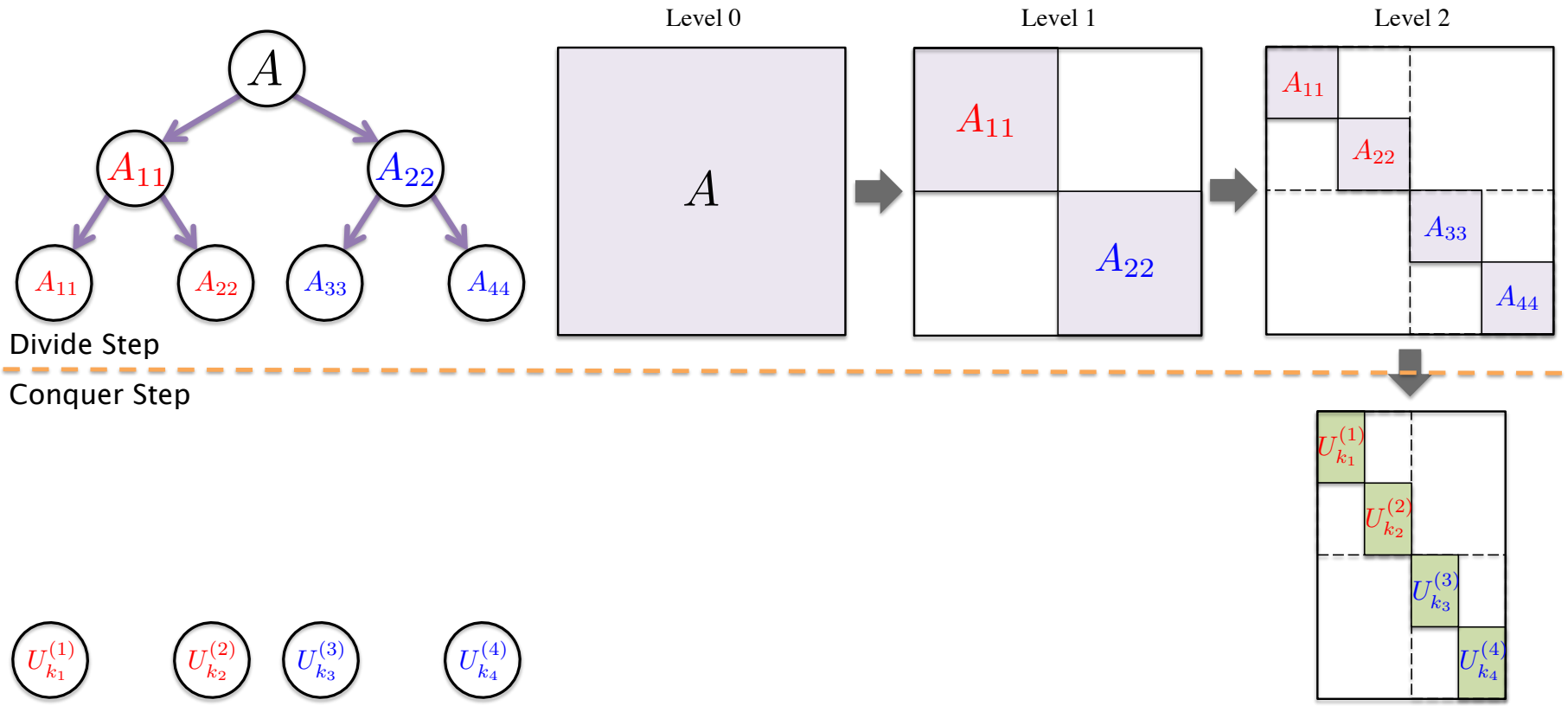
Multi-Scale Spectral Decomposition (MSEIGS)

1. Construct hierarchical clustering of A.
2.
3.
4.



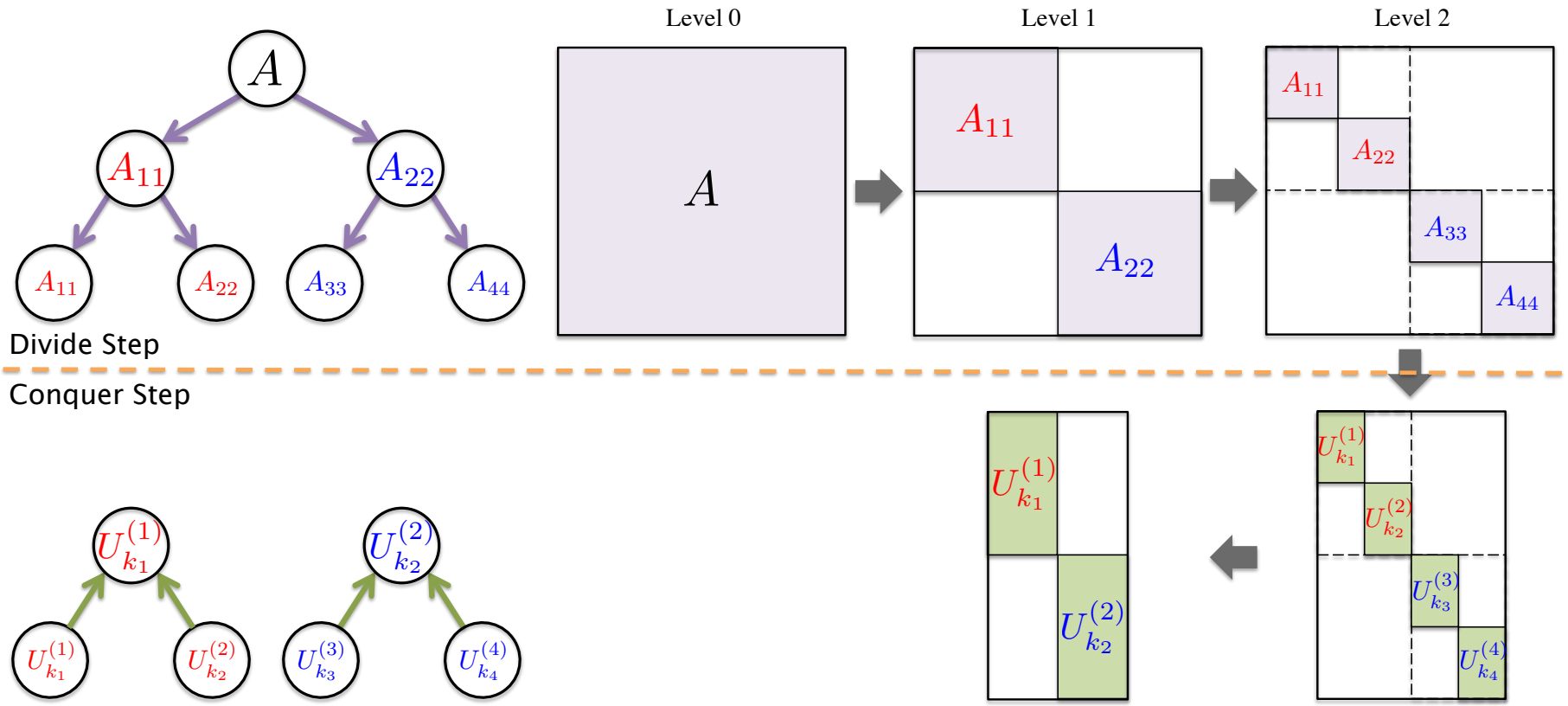
Multi-Scale Spectral Decomposition (MSEIGS)

1. Construct hierarchical clustering of A.
2. Compute approximation at the bottom level (leaf clusters).
3.
4.



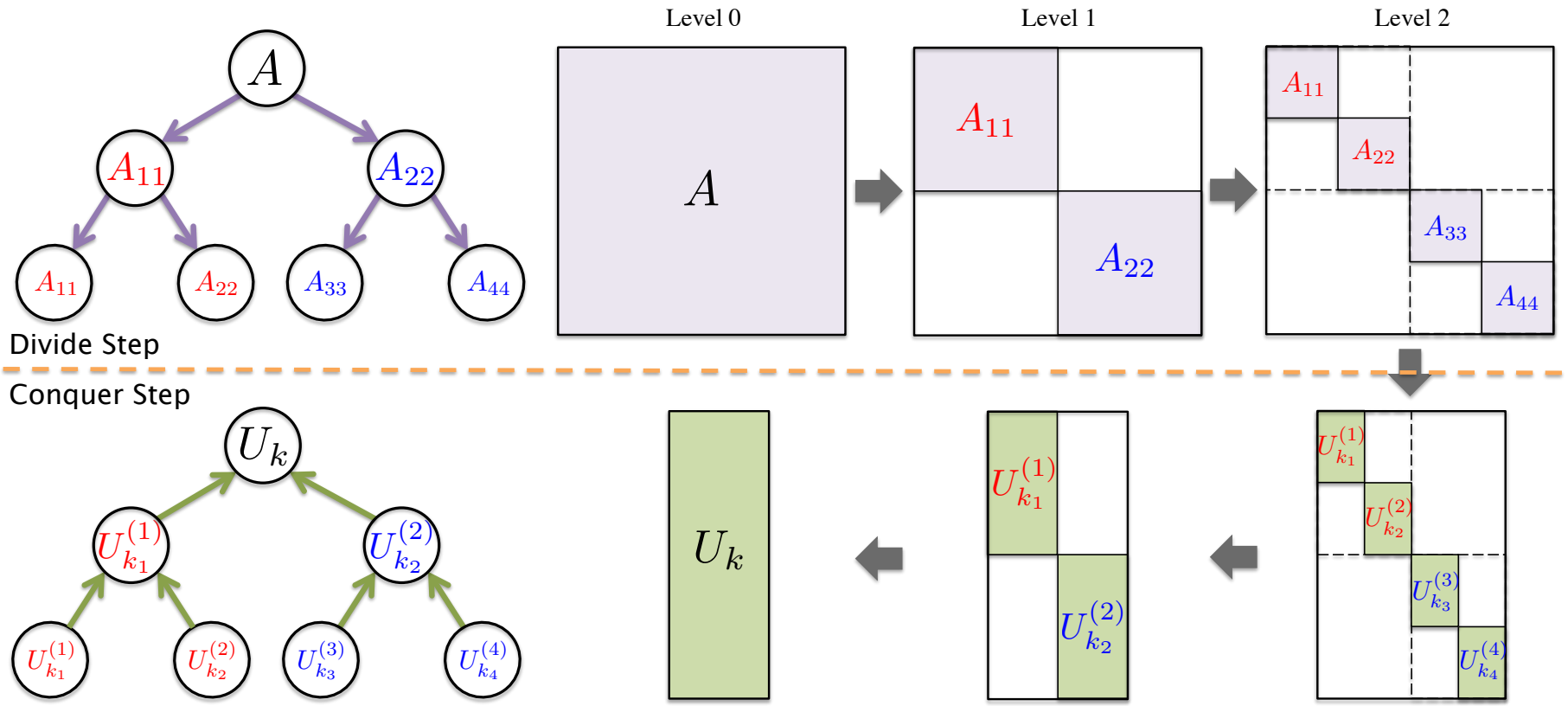
Multi-Scale Spectral Decomposition (MSEIGS)

1. Construct hierarchical clustering of A.
2. Compute approximation at the bottom level (leaf clusters).
3. Compute approximation of parent cluster using child cluster's subspace.
4.



Multi-Scale Spectral Decomposition (MSEIGS)

1. Construct hierarchical clustering of A.
2. Compute approximation at the bottom level (leaf clusters).
3. Compute approximation of parent cluster using child cluster's subspace.
4. Top level outputs (approximate) eigenpairs U_k and Λ_k of A.



Early-Termination Strategy

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

Fast early termination strategy (MSEIGS-Early):

- Approximate the eigenpairs of A by terminating MSEIGS at a certain level ℓ.
- From the top-r eigenpairs of each cluster, select the top-k eigenpairs from all c_ℓ clusters as an approximation of A.

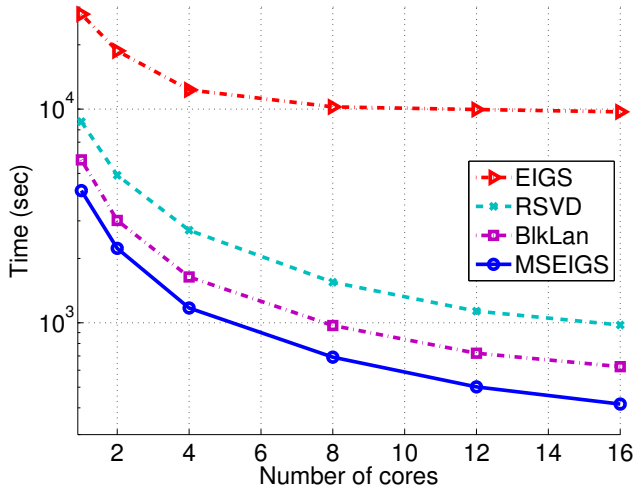
Experimental Results

Multi-core Shared Memory

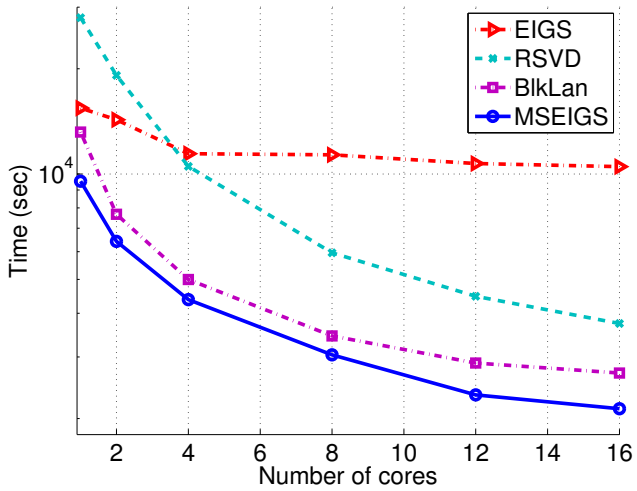
- All methods implemented using C/C++, OpenMP, Intel MKL
- Datasets:

Dataset	CondMat	Amazon	RoadCA	LiveJournal	Friendster	SDWeb
nodes	21,263	334,843	1,965,206	3,997,962	10.00M	82.29M
nonzeros	182,628	1,851,744	5,533,214	69,362,378	83.67M	3.68B
rank k	100	100	200	500	100	50

Number of cores vs. Time to compute similar approximations:



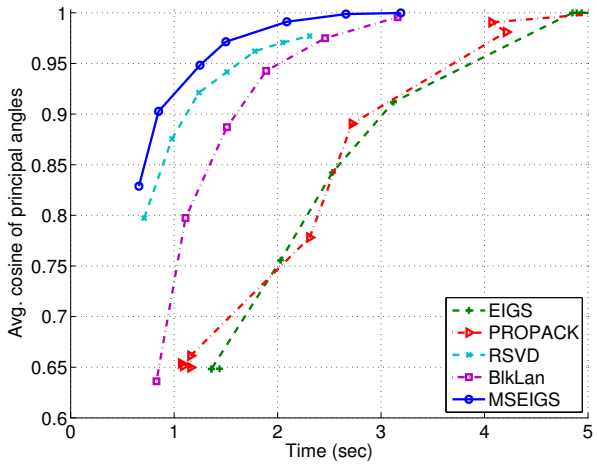
(c) LiveJournal



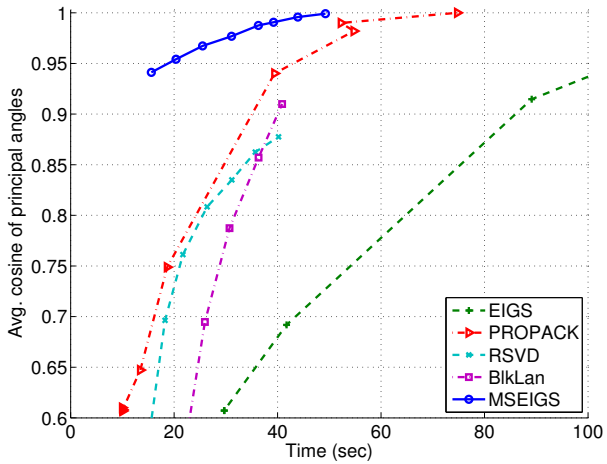
(d) SDWeb

Approximation Results

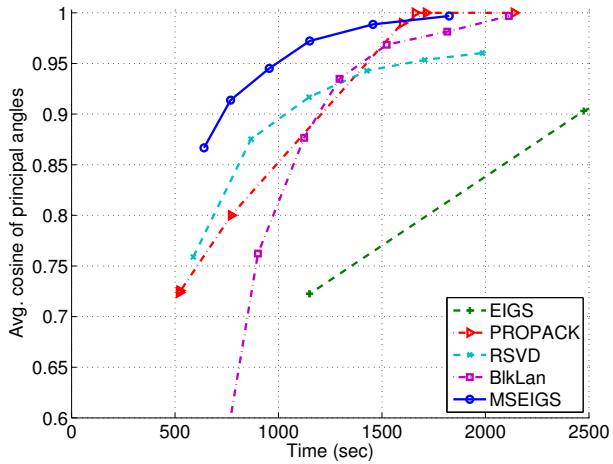
Time vs. Average cosine of principal angles:



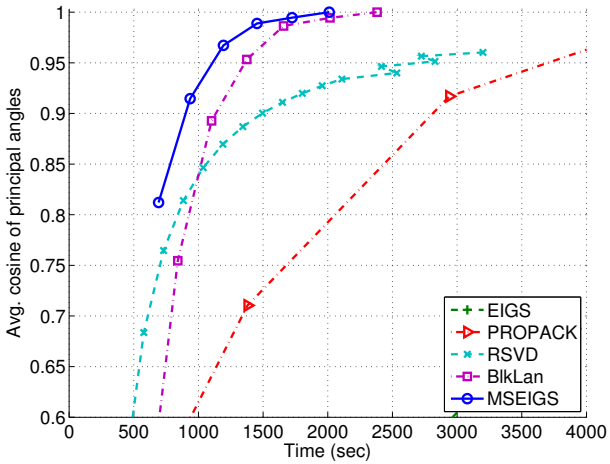
(e) CondMat



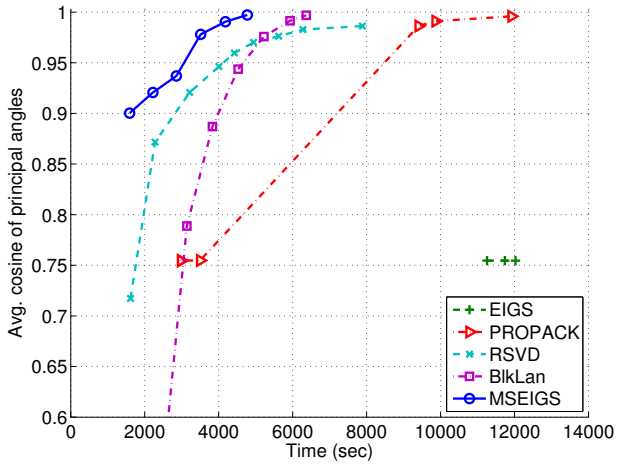
(f) Amazon



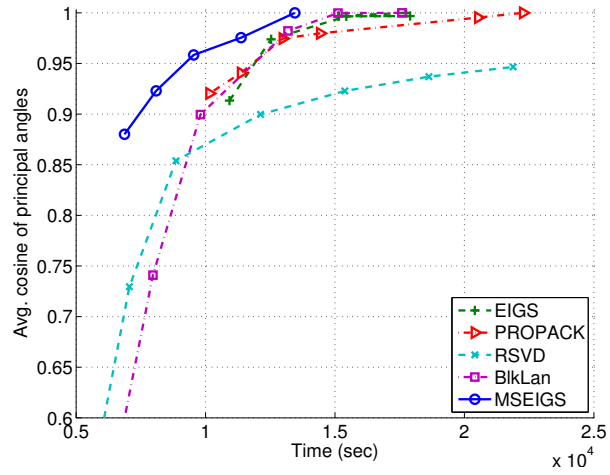
(g) FriendsterSub



(h) RoadCA



(i) LiveJournal



(j) SDWeb

Label Propagation in Semi-supervised Learning

Label Propagation: $F(t+1) = \alpha SF(t) + (1-\alpha)Y$

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

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

Method	Aloi ($k = 1500$)		Delicious ($k = 1000$)		
	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

Recommender Systems with Side-Information

Inductive Matrix Completion (IMC): given a user-item matrix R

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

where $\mathbf{x}_i$ and $\mathbf{y}_j$ 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	RCL@20	eig-time	RCL@20	eig-time	RCL@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 the clustering structure of the graph — easily parallelizable.
- Effective for two important applications: label propagation and inductive matrix completion.
- Future work — dealing with graphs that cannot fit into memory.

Block Lanczos

Given an $n \times n$ symmetric matrix A, compute rank-k approximation $A \approx U_k \Lambda_k U_k^{\mathsf{T}}$.

1: Initialize $B_0 = 0$; $Q_0 = 0$; $Q_1 = \Omega$.
2: **for** $j = 1, 2, \ldots$ **do**
3: $\quad R = AQ_j - Q_{j-1}B_{j-1}^{\mathsf{T}}$ (Let R be orthogonal to Q_{j-1})
4: $\quad D_j = Q_j^{\mathsf{T}} R$ (Obtain D_j by projecting R onto Q_j)
5: $\quad R = R - Q_j D_j$ (Let R be orthogonal to Q_j)
6: $\quad Q_{j+1}B_j = \mathrm{qr}(R)$ (QR-factorization of R)
7: $\quad$ Form the block tri-diagonal matrix T_j
8: $\quad T_j \approx V_k \Sigma_k V_k^{\mathsf{T}}$ (Eigendecomposition of T_j)
9: $\quad$ If residuals $\|A\mathbf{u}_i - \lambda_i \mathbf{u}_i\|$ are sufficiently small, then stop and output $U_k = [Q_1 Q_2 \cdots Q_j] V_k$ and $\Lambda_k = \Sigma_k$ as the approximate eigenpairs.
10: **end for**