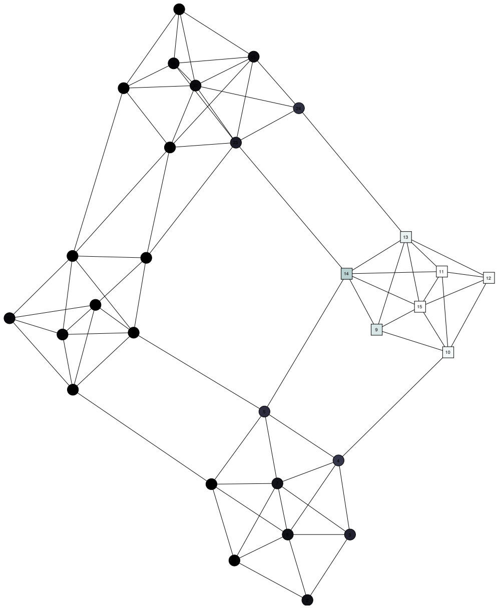
Communities, Spectral Clustering, and Random Walks

David Bindel

Department of Computer Science
Cornell University

26 Sep 2011


25
24
23
13
14
11
12
15
9
10
5
4
7
8
3

Basic setting

Informal approach:

Community = unusually tightly connected nodes in a network

Formal version:

Given graph $G = (V, E)$, seek subgraph $G' = (V', E')$:

1. Based on optimality properties (cut size, modularity, etc)
2. Based on dynamics on G (random walks and variants)

Two approaches unified by linear algebra!

(For today, all graphs are undirected, most are unweighted.)

Unusually tightly connected?

What constitutes “unusual” connectivity of a subgraph?

- High internal connectivity?
- Low external connectivity?

Basic notation

The adjacency matrix $A \in \{0,1\}^{n \times n}$ for G is

$$A_{ij} = \begin{cases} 1, & (i,j) \in E \\ 0, & \text{otherwise} \end{cases}$$

Also define

$$\begin{aligned} e &= \text{vector of } n \text{ ones} \\ d &= Ae = \text{degree vector} \\ D &= \operatorname{diag}(d) \\ L &= D - A = \text{graph Laplacian} \end{aligned}$$

Measuring subgraphs of G

Indicate $V' \subseteq V$ by $s \in \{0,1\}^n$. Can write many properties of the induced subgraph via quadratic forms:

$$\begin{aligned} s^T A s &= |E'| = \text{number of (directed) edges in subgraph} \\ s^T D s &= \text{number of (directed) edges incident on subgraph} \\ s^T L s &= s^T (D - A) s = \text{edges between } V' \text{ and } \bar{V}' \end{aligned}$$

Example: e indicates all of V, and

$$\begin{aligned} m &= e^T A e = e^T D e = \text{number of (directed) edges} \\ 0 &= e^T L e = \text{edges between } V \text{ and } \emptyset \end{aligned}$$

Configuration model

Configuration model for a random graph $\tilde{G} = (\tilde{V}, \tilde{E})$:

- The degree vector d is specified.
- $P\{(i,j) \in \tilde{E}\} = d_i d_j / m$.

Self-loops are allowed.

Expected adjacency matrix, degree vector and matrix, and Laplacian are

$$\begin{aligned}
\tilde{A} &= \frac{dd^T}{m} \\
\tilde{d} &= \tilde{A}e = d \\
\tilde{D} &= D \\
\tilde{L} &= D - \tilde{A}
\end{aligned}$$

Modularity

Define $B = A - \tilde{A} = \tilde{L} - L$. Then

$$\begin{aligned} s^T B s &= \text{unexpected } \textit{extra} \text{ edges in subgraph} \\ &= \text{unexpected } \textit{lack} \text{ of cut edges} \end{aligned}$$

If $s_1, \ldots, s_c$ indicate a partition into c sets,

$$Q := \frac{1}{2} \sum_{j=1}^{c} s_j^T B s_j = \textit{modularity} \text{ of partition}$$

Bisection by optimization

Idea: Find $s \in \{0,1\}^n$ such that $e^T s = n/2$ and

- $s^T L s$ is minimized (min cut) or
- $s^T B s$ is maximized (max modularity)

Equivalently: Find $\bar{s} \in \{-1/2, 1/2\}^n$ such that $e^T \bar{s} = 0$

- $\bar{s}^T L \bar{s} = s^T L s$ is minimized or
- $\bar{s}^T B \bar{s} = s^T B s$ is maximized

Oops — NP hard!

Spectral bisection

Relaxation makes the problem easier:

Hard:	minimize $\bar{s}^T L \bar{s}$	s.t.	$e^T \bar{s} = 0$,	$\bar{s} \in \{-1/2, 1/2\}^n$.
Easy:	minimize $v^T L v$	s.t.	$e^T v = 0$,	$v \in \mathbb{R}^n$, $\|v\|^2 = n/4$.

Now v is an eigenvector for second smallest eigenvalue of L.
Use sign pattern of v to partition $\implies$ spectral bisection.
Heuristic works well in practice (often with some refinement).

Same idea works for modularity.

Rayleigh quotients

Given matrices (K, M), the *generalized Rayleigh quotient* is

$$\rho_{K,M}(x) = \frac{x^T K x}{x^T M x}.$$

Can represent interesting subgraph properties:

$$\begin{aligned}
\rho_{A,I}(s) &= \text{mean internal degree in subgraph} \\
\rho_{L,I}(s) &= \text{edges cut between } V' \text{ and } \bar{V}'
\end{aligned}$$

$$\begin{aligned}
\rho_{A,D}(s) &= \text{fraction of incident edges internal to } V' \\
\rho_{L,D}(s) &= \text{fraction of incident edges cut}
\end{aligned}$$

$$\begin{aligned}
\rho_{B,I}(s) &= \text{mean ``surprising'' internal degree in subgraph} \\
\rho_{B,D}(s) &= \text{mean fraction of internal degree that is surprising} \\
\rho_{B,L}(s) &= \text{fraction of edge cuts that are surprising}
\end{aligned}$$

Rayleigh quotients and eigenvalues

Suppose M is positive definite. Basic connection:

$$\rho_{K,M} \text{ stationary at } x \quad \iff \quad Kx = \rho_{K,M}(x)Mx$$

Stationary points are (generalized) eigenvalues. Reasonable to compute (even though the optimization is nonconvex!).

Limits of Rayleigh quotients

But small variations kill us:

$$\max_{x \neq 0} \frac{x^T A x}{\|x\|_2^2} = \lambda_{\max}(A), \text{ but}$$

$$\max_{x \neq 0} \frac{x^T A x}{\|x\|_1^2} = 1 - \omega^{-1}$$

where ω is the max clique size (Motzkin-Strauss).

Rayleigh quotients and eigenproblems

For M pos. def., have the generalized eigendecomposition

$$W^T M W = I \text{ and } W^T K W = \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n).$$

For any x, the gen. RQ is a weighted average of eigenvalues

$$\rho_{K,M}(x) = \sum_{j=1}^{n} \lambda_j z_j^2,$$

where $z = W^{-1}x/\|W^{-1}x\|_2$. Therefore

1. $\lambda_{\max} = \max_{x \neq 0} \rho_{K,M}(x)$
2. If $\rho_{K,M}(s)$ is near $\lambda_{\max}$, most weight is on large eigenvalues. So s nearly lies in the invariant subspace associated with the large eigenvalues.

So look at invariant subspaces for extreme eigenvalues.

Another reason to look at subspaces

Spectrum of a $G_{n,p}$ graph:

- One large eigenvalue $\approx np$
- Other eigs between $\approx \pm\sqrt{n}p(1-p)/4$
- Adjacency matrix = $pee^T +$ "noise"

Composite model: $A \approx S \operatorname{diag}(\beta) S^T$, $S \in \{0,1\}^{n \times c}$

- Motivation: possibly-overlapping random graphs
- Columns of S are one basis for range space
- Want to go from some general basis back to S

Indicators from subspaces, take 1

U spans a small subspace (e.g. an invariant subspace)

1. If $\text{span}(u_1, \ldots, u_c) \approx \text{span}(s_1, \ldots, s_c)$ where $\{s_j\}$ indicate a partition, rows of U in the same partition are identical.

   Idea: Treat rows of U are *latent coordinates*. Cluster.

2. Suppose we have *some* indicator $s \approx Uy$. Then row $U(j,:)$
   - forms an acute angle with y when $s_j = 1$
   - is almost normal to y when $s_j = 0$.

   Clustering? What if sets overlap?

Indicators from subspaces, take 2

Suppose $s \approx Uy$ for some y, $s_i = 1$. Want to find s.
Try optimization (a linear program):

$$\begin{array}{rll} \text{minimize} & \|\tilde{s}\|_1 & \text{(proxy for sparsity of } \tilde{s}\text{)} \\ \text{s.t.} & \tilde{s} = Uy & \text{(}\tilde{s}\text{ in the right space)} \\ & \tilde{s}_i \geq 1 & \text{(“seed” constraint)} \\ & \tilde{s} \geq 0 & \text{(componentwise nonnegativity)} \end{array}$$

Recovers smallest set containing node i if

- $U = SY^{-1}$ exactly.
- Each set contains at least one element only in that set. (Frequently works if there is not “too much” overlap.)

What about noise? Generally need a thresholding strategy.

Indicators from subspaces, take 3

Alternate optimization (box-constrained quadratic program):

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\tilde{s}^T P \tilde{s} + \tau \|\tilde{s}\|_1 \\ \text{s.t.} \quad & \tilde{s}_i \geq 1 \\ & \tilde{s} \geq 0 \end{aligned}$$

Recover LP with $P = I - UU^T$ and $\tau \to 0$ (assuming $U^T U = I$).

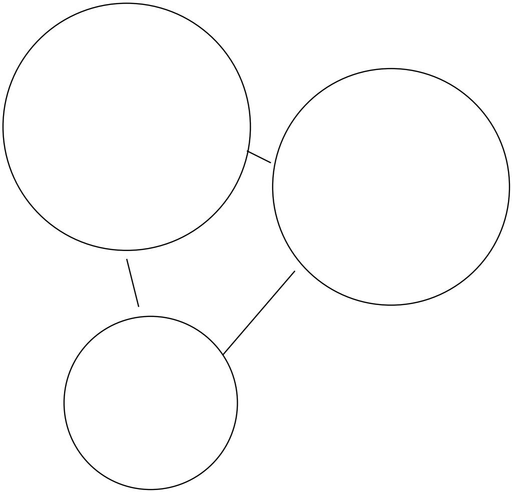
- Can let P be more general semidefinite matrix (e.g. $P = L$)
- Size of τ controls sparsity (can automate choice)

Summary so far

Two pieces to spectral community detection:

- Pull out an invariant subspace
- Mine the subspace for community structure

The random walker



The random walker, take 1

Lazy random walk on a graph:

$$p_{k+1} = \hat{T} p_k = \frac{1}{2}(I + T) p_k \quad \rightarrow \quad p_\infty = d/m$$

where $T = AD^{-1}$ is a transition matrix (column stochastic).

Idea: extract community structure from random walk dynamics

- Start at a node i and take a few steps
- Rapidly explore the local community (only one?)
- Probability “leaks” into adjoining communities (slowly?)

The random walker, take 2

If a random walk *starts* at known i, goes k steps:

$$p_k(j) = P\{\text{end at } j \mid \text{start at } i\} = e_j^T \hat{T}^k e_i$$

If *end* at known j, uniform prior on starting point, then

$$q_k(i) = P\{\text{start at } i \mid \text{end at } j\} = \frac{1}{Z_{j,k}} e_j^T \hat{T}^k e_i$$

Idea: extract structure from how fast we forget starting points

Day 1: David came up with a funny joke!

Day 2: There's a joke going around the CS department.

Day 3: I read this bad joke while browsing the web...

Simon-Ando theory

Markov chain with loosely-coupled subchains. Dynamics are:

- Rapid *local* mixing: after a few steps

  $$p_k \approx \sum_{j=1}^{c} \alpha_{j,k} p_\infty^{(j)}$$

  where $p_\infty^{(j)}$ is a local equilibrium for the jth subchain
- Slow equilibration: $\alpha_{j,k} \to \alpha_{j,\infty}$.

Alternately, rapid local mixing looks like:

$$\phi_k \approx \sum_{j=1}^{c} \gamma_{j,k} s_j$$

where s_j is an indicator for nodes in one subchain.

Simon-Ando theory

In chemistry:
transient dynamics = transitions among metastable states.

In network analysis:
transient dynamics = transitions among communities?

But what if mixing happens so fast we miss the transient?

Random walks and spectrum

Write

$$\begin{aligned}\hat{T} &= \frac{1}{2}(I+T)\\ T &= AD^{-1} = D^{1/2}\bar{A}D^{-1/2}\\ \bar{A} &= D^{-1/2}AD^{-1/2}\end{aligned}$$

Have an eigendecomposition $\bar{A} = Q\Lambda Q^T$. Then

$$\begin{aligned}p_k &= (D^{1/2}Q)(I+\Lambda)^k(Q^T D^{-1/2})p_0\\ \phi_k &= (D^{-1/2}Q)(I+\Lambda)^k(Q^T D^{1/2})\phi_0/Z_{j,k}\end{aligned}$$

Spectral picture of Simon-Ando

$$
\begin{aligned}
\phi_k &= (D^{-1/2}Q)(I+\Lambda)^k(Q^T D^{1/2})\phi_0/Z_{j,k} \\
&= \frac{1}{Z_{j,k}}\sum_{j=1}^{n}(1+\lambda_j)^k c_j(D^{-1/2}\phi_j) \\
&\approx \frac{1}{Z_{j,k}}\sum_{j=1}^{c}(1+\lambda_j)^k c_j(D^{-1/2}q_j)
\end{aligned}
$$

- Gap in the spectrum between λ_c and λ_{c+1}
- After a few steps, $(1+\lambda_{c+1})^k$ negligible, but not $(1+\lambda_c)^k$. So q_k lies approximately in span of $D^{-1/2}q_1, \ldots, D^{-1/2}q_c$.
- Treat as a perturbation of decoupled case where subchain indicator vectors are eigenvectors for unit eigenvalues. $D^{-1/2}q_j \approx$ linear combo of indicators, $j = 1, \ldots, c$.

Summary so far

Two pieces to spectral community detection:

- Pull out an invariant subspace
- Mine the subspace for community structure

Motivation: optimization or random walk dynamics.

But...

- What about when n and c are both large?
- What if there is no clear spectral gap?

Would like an alternative to invariant subspaces!

Eigenvectors to Ritz vectors

Eigenvectors are stationary points of Rayleigh quotients.
Find stationary points in a subspace $\implies$ *Ritz* vectors.

Usual approach to large-scale eigenproblems:

1. Generate a basis for a *Krylov subspace*

$$\mathcal{K}_k(A, x_0) = \text{span}\{x_0, Ax_0, A^2 x_0, \dots, A^{k-1} x_0\}$$

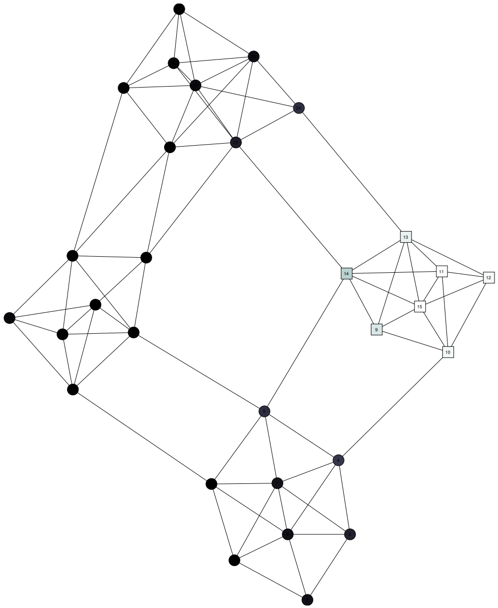
2. Ritz values rapidly approximate extreme eigenvalues
3. Ritz vectors approximate extreme eigenvectors

Idea: Instead of searching invariant subspace, search in a space spanned by a few scaled Ritz vectors. Pulls out dynamics of *short* random walks (vs long).

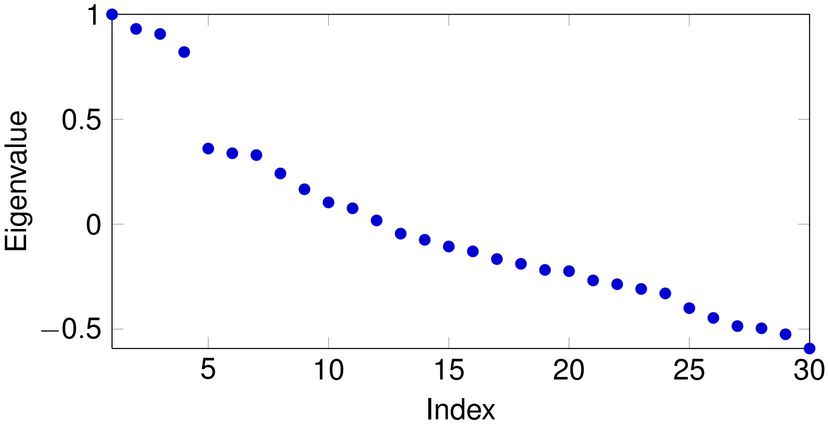
Current favorite method

1. Pick “seed” nodes $j_1, j_2, \ldots$
2. Take short random walks (length k) from each seed
3. Extract a few Ritz vectors (fewer than k) from span$\{\phi_0, \phi_1, \ldots, \phi_{k-1}\}$.
4. Use quadratic programming to find approximate indicators in subspace space spanned by all Ritz vectors.
5. Possibly add more seeds and return to step 1.
6. Threshold to get initial indicator approximation.
7. Greedily optimize angle between indicator and space.

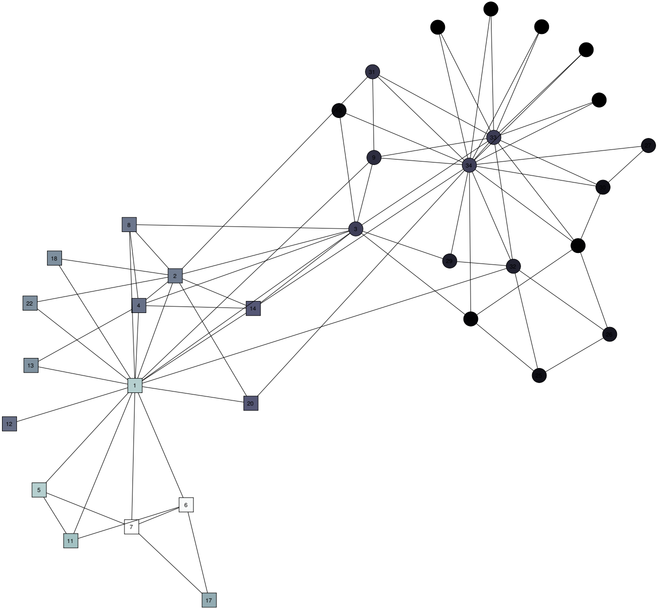
Wang test graph



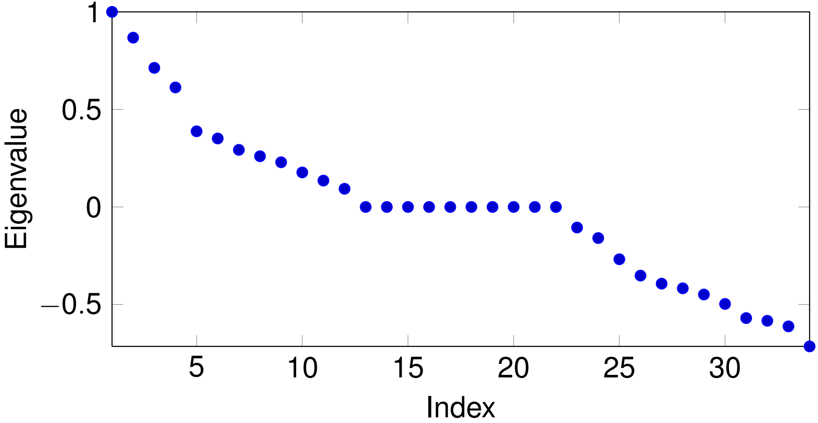
Spectrum for Wang test graph



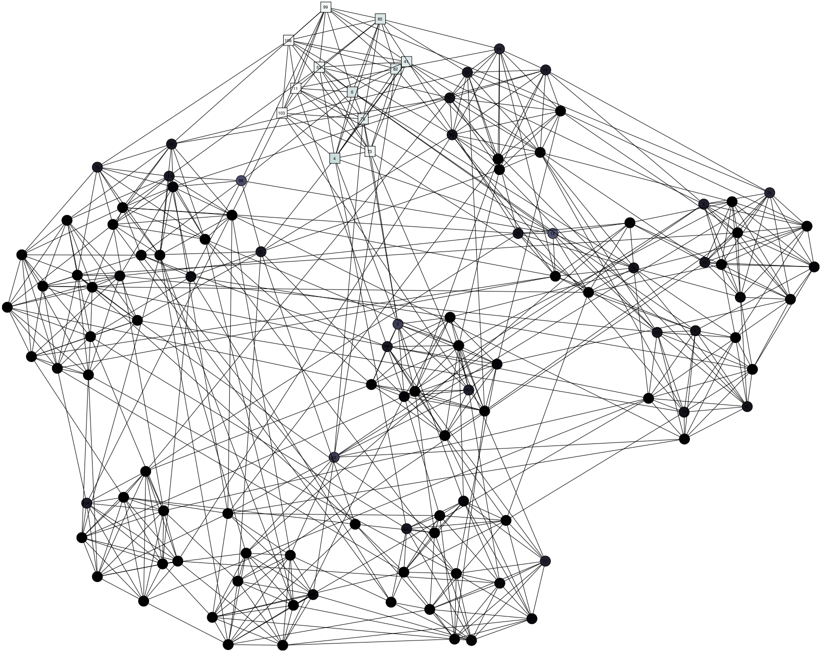
Zachary Karate graph



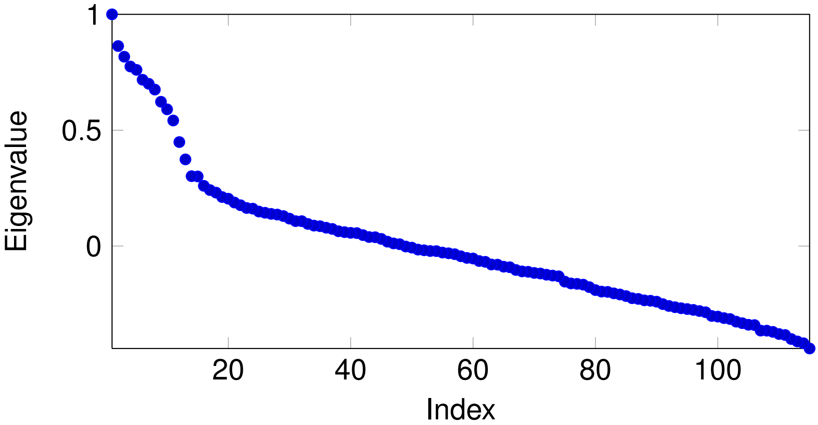
Spectrum for Karate



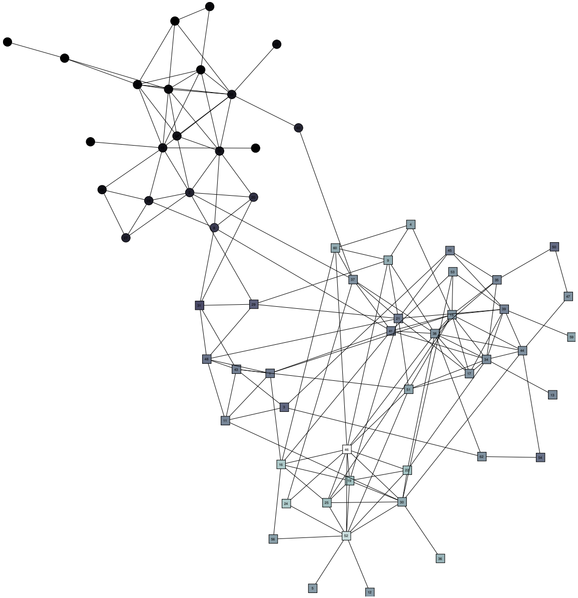
Football graph



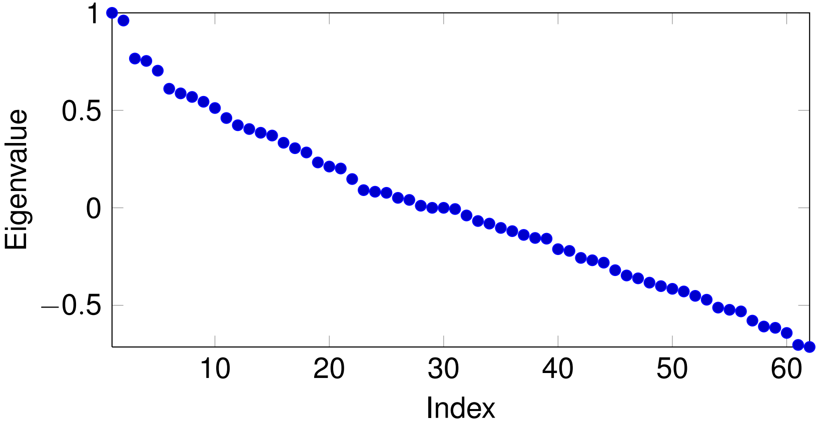
Spectrum for Football



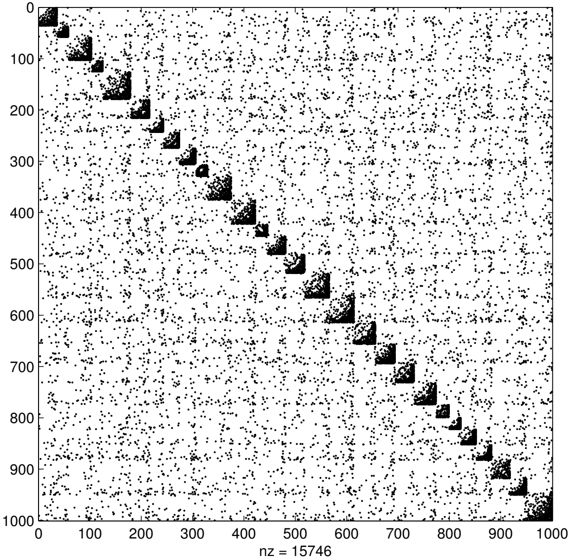
Dolphin graph



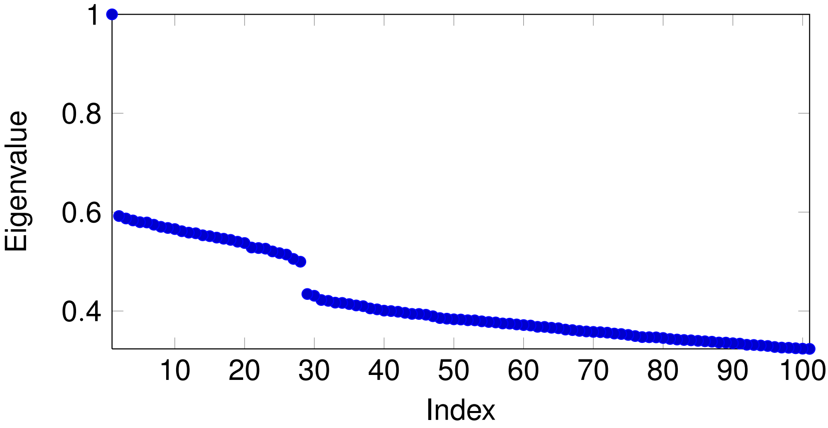
Spectrum for Dolphin



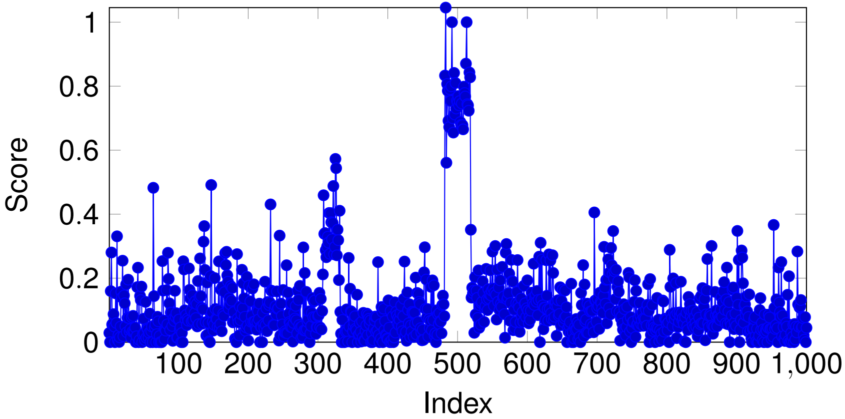
Non-overlapping synthetic benchmark ($\mu = 0.5$)



Spectrum for synthetic benchmark

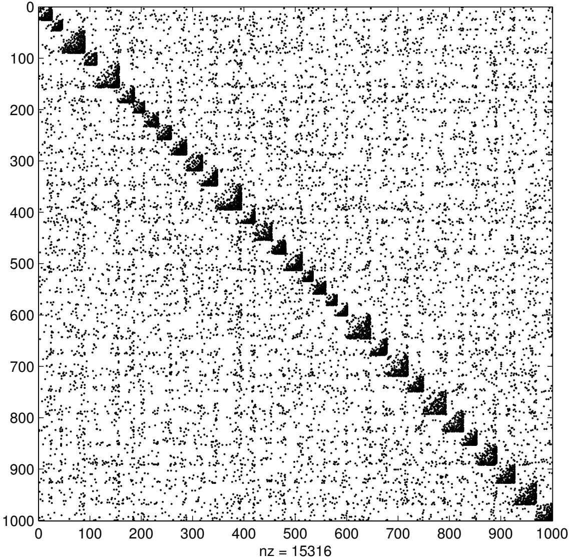


Score vector

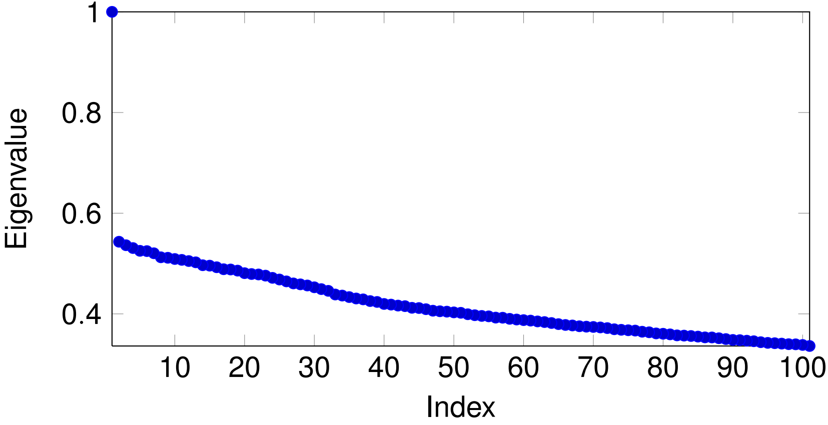


Score vector for the two-node seed of 492 and 513 in the first LFR benchmark graph. Ten steps, three Ritz vectors.

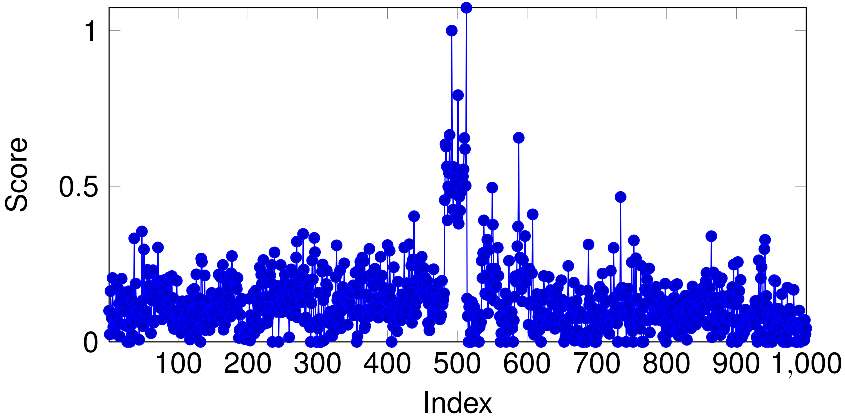
Non-overlapping synthetic benchmark ($\mu = 0.6$)



Spectrum for synthetic benchmark



Score vector

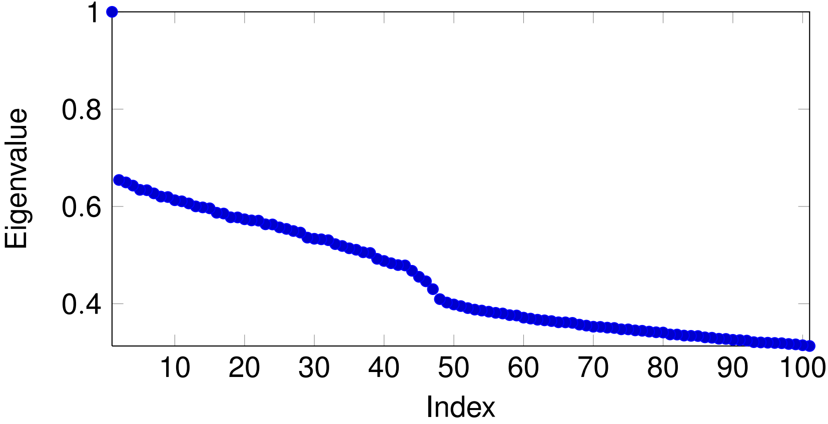


Score vector for the two-node seed of 492 and 513 in the first LFR benchmark graph. Ten steps, three Ritz vectors.

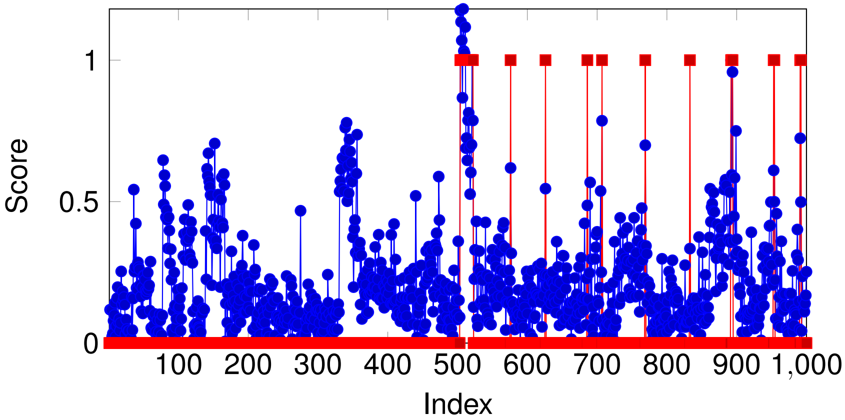
Overlapping synthetic benchmark ($\mu = 0.3$)

- 1000 nodes
- 47 communities
- 500 nodes belong to two communities

Spectrum for synthetic benchmark

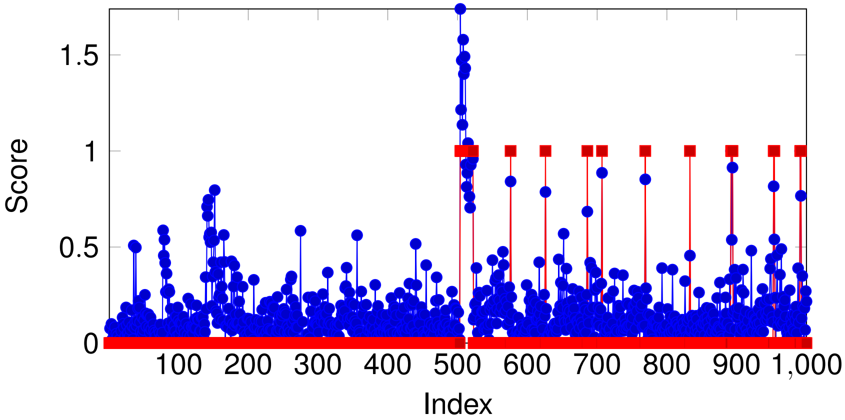


Score vector



Score vector for the two-node seed of 521 and 892.
The desired indicator is in red.

Score vector



Score vector for the two-node seed of 521 and 892 + twelve reseeds. The desired indicator is in red.

Conclusions

Classic spectral methods use eigenvectors to find communities, but:

- We don't need to stop at partitioning!
  - Overlap is okay
  - Key is how we mine the subspace
- We don't need to stop at eigenvectors!
  - Can also use *Ritz* vectors
  - Computation is cheap: short random walks