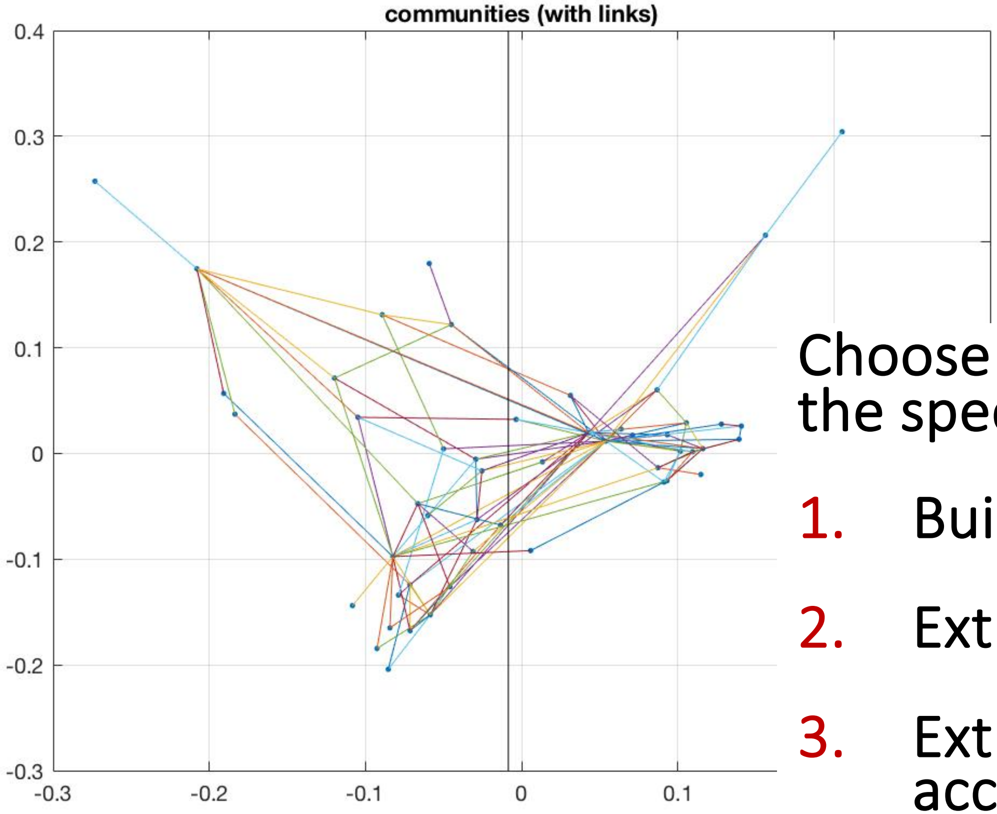
Network Science

Lab #5
Community detection
Spectral approach

Timetable

- ❑ Lab 1 – Fri Oct 12
  Scale free properties
- ❑ Lab 2 – Fri Oct 19
  Albert-Baràbasi model
- ❑ Lab 3 – Fri Oct 26
  Assortativity
- ❑ Lab 4 – Fri Nov 16
  Ranking
- ❑ Lab 5 – Fri Nov 23
  Community detection – Spectral
- ❑ Lab 6 – Fri Nov 30
  Community detection – PageRank-Nibble
- ❑ Lab 7 – Fri Dec 7
  Gephi

Lab 5 – Community detection



eigenvectors

$v_i = D^{-½} x_i$

provide a more stable representation

ASSIGNMENT a

Choose one of the datasets, then apply the spectral algorithm:

1. Build the normalized Laplacian
2. Extract/plot eigenvalues
3. Extract/show the nodes coordinates according to Fiedler`s and the following eigenvector (appropriately scale eigenvectors)
4. Identify communities according to Fiedler’s eigenvector

Lab 5 – Community detection

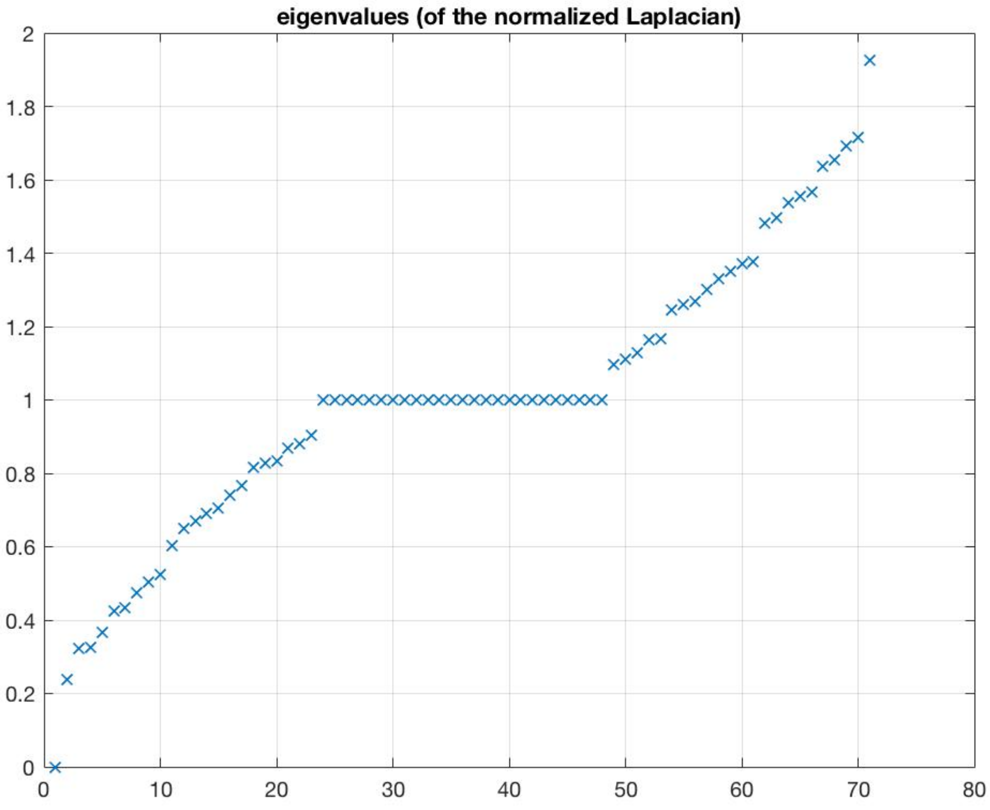
$$\boldsymbol{L_1 = I - D^{-½} \cdot A \cdot D^{-½}}$$

Normalized Laplacian matrix

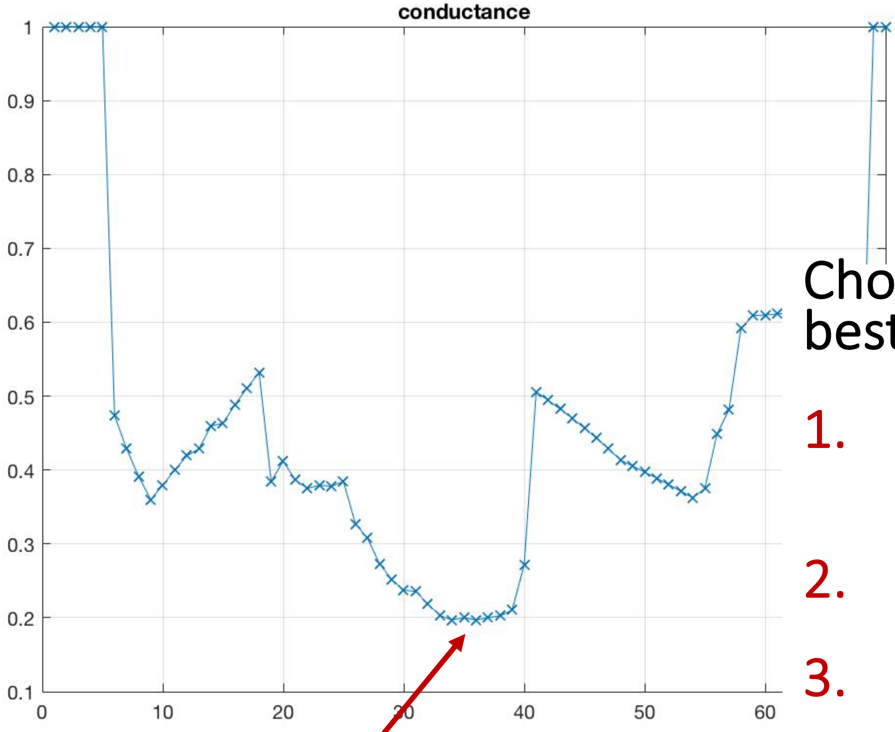
degree matrix

D = diag(***d***)

Adjacency matrix (undirected/symmetric)



Lab 5 – Community detection



minimum conductance value

ASSIGNMENT b

Choose the best community according to the best conductance value:

1. Order nodes according to their score in Fiedler`s eigenvector
2. Sweep across nodes
3. Identify the best community as the one providing the minimum conductance value
4. Compare the conductance value to Cheeger’s bound $(2\lambda_{N-1})^{½}$
5. Have you found a good community?

Computing the sweep

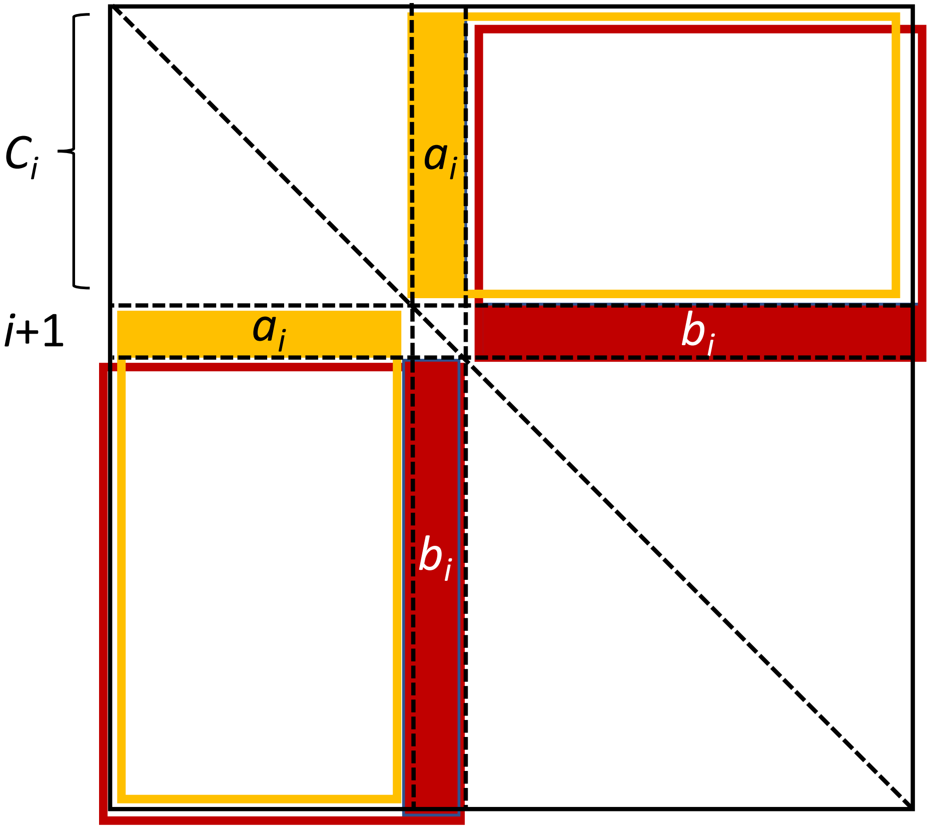
Algorithm

- ❑ Let $C_i = \{1,2,\dots,i\}$
- ❑ Node degree is $d_i = a_i + b_i$
- ❑ Association update

  $\mathrm{assoc}(C_{i+1}) = \mathrm{assoc}(C_i) + d_i$

- ❑ Cut update

  $\mathrm{cut}(C_{i+1}) = \mathrm{cut}(C_i) - a_i + b_i$



Goodness of fit = Conductance

✓ $\phi(C_i) = \mathrm{cut}(C_i) / \min(\mathrm{assoc}(C_i), D\text{-}\mathrm{assoc}(C_i))$

Lab 5 – MatLab hints

1. tic: starts the counter
2. toc: reads the counter

3. spdiags(ones(N,1),0,N,N) : sparse identity matrix
4. eigs: extracts (ordered) eigenvalues

5. cumsum: evaluates a cumulative sum
6. triu: extracts the upper triangular matrix
7. tril: extracts the lower triangular matrix