

Spectral Clustering: Theory and Practice

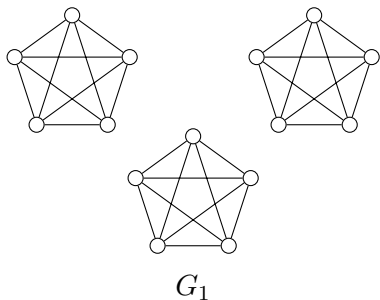
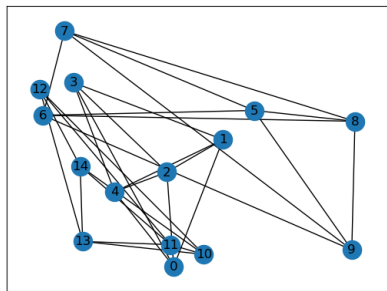
Jephian C.-H. Lin 林晉宏

Department of Applied Mathematics, National Sun Yat-sen University

November 12, 2025

MALA Seminar

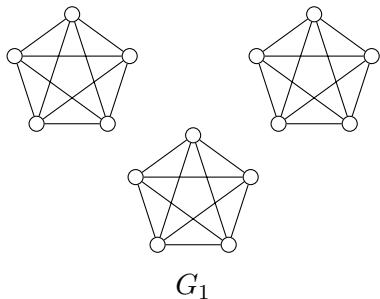
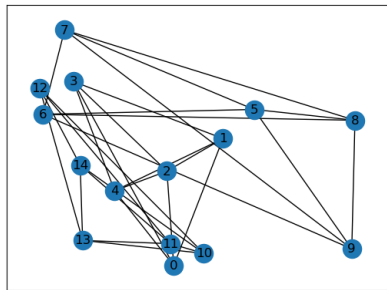
How to find the components?



☺ Breadth-first search

☺ Laplacian matrix

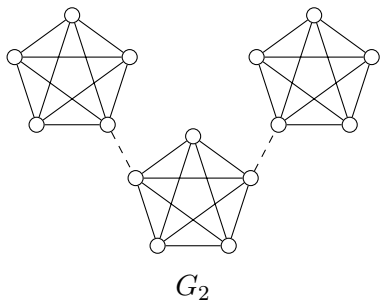
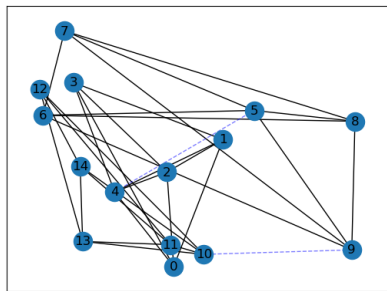
How to find the components?



☺ Breadth-first search

☺ Laplacian matrix

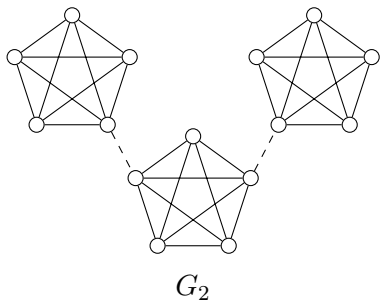
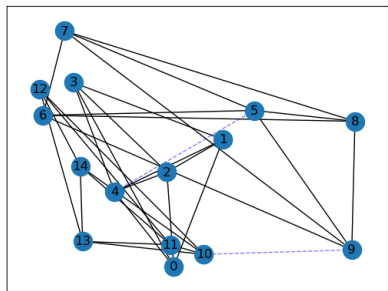
How to find the clusters?



☹ Breadth-first search

☺ Laplacian matrix

How to find the clusters?



☹ Breadth-first search

☺ Laplacian matrix



Miroslav Fiedler
1926–2015

Known for

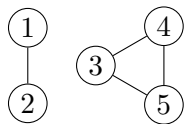
- algebraic connectivity,
- Fiedler vector,
- and more.

Have impact on

- graph partition,
- spectral clustering,
- image segmentation,
- and more.

Source: MacTutor <https://mathshistory.st-andrews.ac.uk/Biographies/Fiedler/>

Laplacian matrix



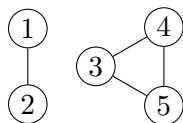
$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

Definition

Let G be a graph on n vertices. The **Laplacian matrix** of G is the $n \times n$ matrix $L(G) = [l_{i,j}]$ such that

$$l_{i,j} = \begin{cases} -1 & \text{if } \{i,j\} \in E(G), \\ \deg_G(i) & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Laplacian matrix



$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

Proposition

- 1 $L\mathbf{1} = \mathbf{0}$.
- 2 $\mathbf{x}^\top L\mathbf{x} = \sum_{\{i,j\} \in E} (x_i - x_j)^2$, which means L is PSD.
- 3 $L\mathbf{x} = \mathbf{0} \iff \mathbf{x}^\top L\mathbf{x} = 0$.

Example

For $G = K_2 \dot{\cup} K_3$,

$$\mathbf{x}^\top L\mathbf{x} = (x_1 - x_2)^2 + (x_3 - x_4)^2 + (x_4 - x_5)^2 + (x_3 - x_5)^2.$$

Count the number of components by the Laplacian matrix

Theorem (Fiedler 1973, Anderson and Morley 1971)

Let G be a graph and $L = L(G)$. Then $\text{null}(L)$ is the number of components of G , and

$$\ker(L) = \text{span}\{\phi_{X_1}, \dots, \phi_{X_k}\},$$

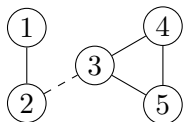
where X_1, \dots, X_k are the vertex sets of the components of G .

Example

For $G = K_2 \dot{\cup} K_3$,

$$\text{spec}(L) = \{0, 0, 2, 3, 3\} \text{ and } \ker(L) = \text{span} \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right\}.$$

Weighted Laplacian matrix



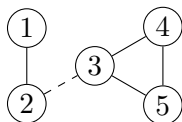
$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 1.1 & -0.1 & 0 & 0 \\ 0 & -0.1 & 2.1 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

Definition

Let G be a weighted graph on n vertices with weights $w_{i,j}$. The **weighted Laplacian matrix** of G is the $n \times n$ matrix $L(G) = [l_{i,j}]$ such that

$$l_{i,j} = \begin{cases} -w_{i,j} & \text{if } \{i,j\} \in E(G), \\ \sum_{k:k \sim i} w_{i,k} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Weighted Laplacian matrix



$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 1.1 & -0.1 & 0 & 0 \\ 0 & -0.1 & 2.1 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

Proposition

- 1 $L\mathbf{1} = \mathbf{0}$.
- 2 $\mathbf{x}^\top L\mathbf{x} = \sum_{\{i,j\} \in E} w_{i,j}(x_i - x_j)^2$, which means L is PSD.
- 3 $L\mathbf{x} = \mathbf{0} \iff \mathbf{x}^\top L\mathbf{x} = 0$.

Example

For $G = K_2 \dot{\cup} K_3 + \{2, 3\}$,

$$\mathbf{x}^\top L\mathbf{x} = (x_1 - x_2)^2 + (x_3 - x_4)^2 + (x_4 - x_5)^2 + (x_3 - x_5)^2 + 0.1(x_2 - x_3)^2.$$

Count the number of components by the Laplacian matrix

Theorem (Fiedler 1973, Anderson and Morley 1971)

Let G be a graph and $L = L(G)$. Then $\text{null}(L)$ is the number of components of G , and

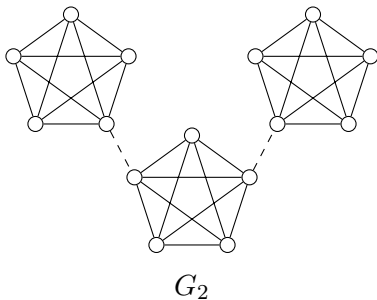
$$\ker(L) = \text{span}\{\phi_{X_1}, \dots, \phi_{X_k}\},$$

where X_1, \dots, X_k are the vertex sets of the components of G .

Example

For $G = K_2 \dot{\cup} K_3 + \{2, 3\}$,

$$\text{spec}(L) = \{0, 0.08, 2.05, 3, 3.07\} \text{ and } \ker(L) = \text{span} \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right\}.$$



$$\text{spec}(L) = \{0, 0.02, 0.06, 5, \dots\} \text{ and } \ker(L) = \{\mathbf{1}\}.$$

Just a small perturbation: first few eigvals and eigvecs

$$\{0, 0, 0\} \rightarrow \{0, 0.02, 0.06\}$$

$$\begin{bmatrix} 0.45 & 0 & 0 \\ 0.45 & 0 & 0 \\ 0.45 & 0 & 0 \\ 0.45 & 0 & 0 \\ 0.45 & 0 & 0 \\ 0 & 0.45 & 0 \\ 0 & 0.45 & 0 \\ 0 & 0.45 & 0 \\ 0 & 0.45 & 0 \\ 0 & 0.45 & 0 \\ 0 & 0 & 0.45 \\ 0 & 0 & 0.45 \\ 0 & 0 & 0.45 \\ 0 & 0 & 0.45 \\ 0 & 0 & 0.45 \end{bmatrix} \rightarrow \begin{bmatrix} 0.26 & 0.32 & -0.18 \\ 0.26 & 0.32 & -0.18 \\ 0.26 & 0.32 & -0.18 \\ 0.26 & 0.32 & -0.18 \\ 0.26 & 0.31 & -0.17 \\ 0.26 & 0.01 & 0.36 \\ 0.26 & -0.00 & 0.37 \\ 0.26 & -0.00 & 0.37 \\ 0.26 & -0.00 & 0.37 \\ 0.26 & -0.01 & 0.36 \\ 0.26 & -0.31 & -0.17 \\ 0.26 & -0.32 & -0.18 \\ 0.26 & -0.32 & -0.18 \\ 0.26 & -0.32 & -0.18 \\ 0.26 & -0.32 & -0.18 \end{bmatrix}$$

Count the number of clusters by the Laplacian matrix

Theorem

Let G be a weighted graph and $L = L(G)$. Then *the number of zeroish eigenvalues* suggests the number of clusters of G , and vertices in the same cluster share *similar values in each eigenvector*.

Example

For $G = K_2 \dot{\cup} K_3 + \{2, 3\}$,

$$\text{spec}(L) = \{0, 0.08, 2.05, 3, 3.07\} \text{ and } 0, 0.08 \rightarrow \begin{bmatrix} 0.45 \\ 0.45 \\ 0.45 \\ 0.45 \\ 0.45 \end{bmatrix}, \begin{bmatrix} 0.57 \\ 0.52 \\ 0.35 \\ 0.37 \\ 0.37 \end{bmatrix}$$

Spectral embedding algorithm

Algorithm

Input: a **weighted graph** G on n vertices and a **targeted dimension** d

Output: an $n \times d$ matrix Y

Steps: ① $L \leftarrow L(G)$.

② Find the first d eigenvalues $\lambda_1 \leq \dots \leq \lambda_d$ and the corresponding eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_d$.

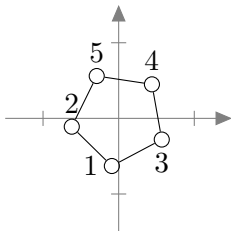
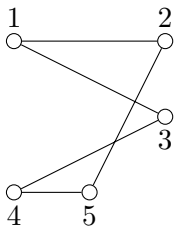
③ $Y \leftarrow$ the matrix composed of columns $\mathbf{u}_1, \dots, \mathbf{u}_d$.

④ Let $\mathbf{y}_1, \dots, \mathbf{y}_n$ be the rows of Y . Define the embedding $f : V(G) \rightarrow \mathbb{R}^d$ by $i \mapsto \mathbf{y}_i$.

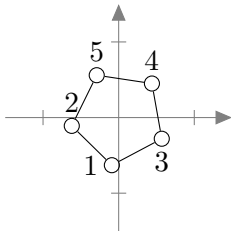
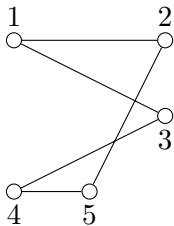
Remark

① Since $\mathbf{u}_1 = \frac{1}{\sqrt{n}} \mathbf{1}$, people often take $\lambda_2 < \dots < \lambda_{d+1}$ and their eigenvectors instead.

② Main idea: The embedding try to **put adjacent vertices together**—the stronger the weight, the closer they are.



$$L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & 0 & 0 & -1 \\ -1 & 0 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix} \rightarrow Y = \begin{bmatrix} -0.09 & -0.63 \\ -0.62 & -0.11 \\ 0.57 & -0.28 \\ 0.44 & 0.45 \\ -0.29 & 0.56 \end{bmatrix}$$



The spectral embedding algorithm occurs in

- graph drawing (Hall 1970, Koren 2005),
- graph partitioning (Pothén, Simon, and Liou 1990),
- graph ordering (Juvan and Mohar 1992),
- spectral clustering (Shi and Malik 2000),
- Laplacian eigenmap (Belkin and Niyogi 2003),
- and more.

How to draw a graph properly?

Problem

Given a weighted graph G on n vertices and a target dimension d , find an $n \times d$ matrix Y such that

$$\begin{aligned} & \text{minimize} && \text{tr}(Y^\top LY) = \sum_{\{i,j\} \in E(G)} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \\ & \text{subject to} && \mathbf{1}^\top Y = \mathbf{0}^\top \text{ and} \\ & && Y^\top Y = I. \end{aligned}$$

Intuition:

- $\text{tr}(Y^\top LY)$: the potential energy of a spring-mass system.
- $\mathbf{1}^\top Y = \mathbf{0}^\top$: centered at the origin.
- $Y^\top Y = I$: normalized each coordinate.

Spectral embedding algorithm generates the answer!

How to draw a graph properly?

Problem

Given a weighted graph G on n vertices and a target dimension d , find an $n \times d$ matrix Y such that

$$\begin{aligned} & \text{minimize} && \text{tr}(Y^\top LY) = \sum_{\{i,j\} \in E(G)} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \\ & \text{subject to} && \mathbf{1}^\top Y = \mathbf{0}^\top \text{ and} \\ & && Y^\top Y = I. \end{aligned}$$

Intuition:

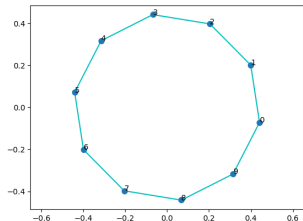
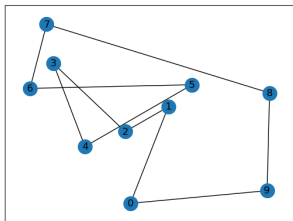
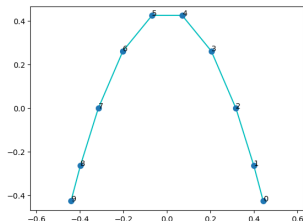
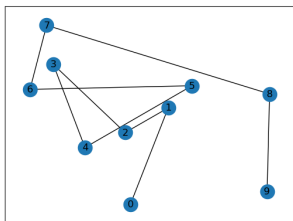
- $\text{tr}(Y^\top LY)$: the potential energy of a spring-mass system.
- $\mathbf{1}^\top Y = \mathbf{0}^\top$: centered at the origin.
- $Y^\top Y = I$: normalized each coordinate.

Spectral embedding algorithm generates the answer!

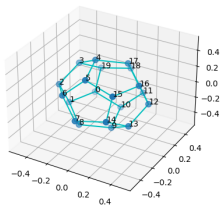
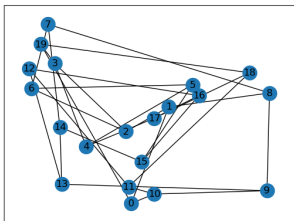
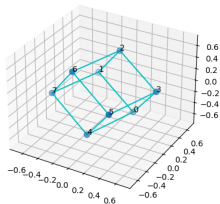
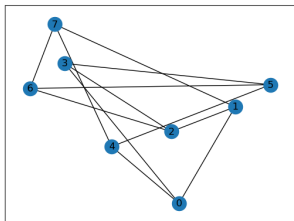
Some examples



More examples of spectral embedding



More examples of spectral embedding

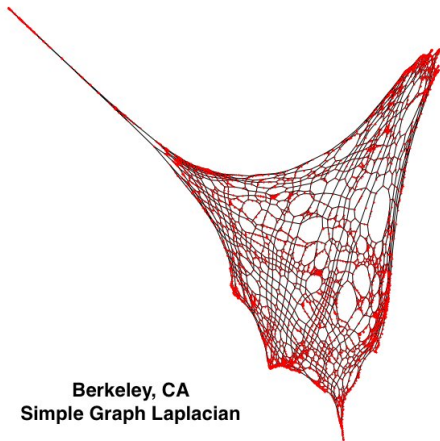


Example of spectral clustering



Source: Kuan Butts's blog <https://kuanbutts.com/2017/10/21/spectral-cluster-berkeley/>

Example of spectral clustering



Source: Kuan Butts's blog <https://kuanbutts.com/2017/10/21/spectral-cluster-berkeley/>

Example of spectral clustering



Source: Kuan Butts's blog <https://kuanbutts.com/2017/10/21/spectral-cluster-berkeley/>

Different choices of the weight assignment

Gaussian similarity function

$$e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}$$

Cosine similarity $c_{i,j} = \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}$

$$c_{i,j} + 1 \quad \text{or} \quad \frac{1}{\sqrt{2}} \sqrt{c_{i,j} + 1}$$

Pearson correlation coefficient $r_{i,j}$

$$r_{ij} + 1 \quad \text{or} \quad \frac{1}{\sqrt{2}} \sqrt{r_{ij} + 1}$$

In the spectral embedding algorithm, how does the weight assignment change the embedding output?

Different choices of the weight assignment

Gaussian similarity function

$$e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}$$

Cosine similarity $c_{i,j} = \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}$

$$c_{i,j} + 1 \quad \text{or} \quad \frac{1}{\sqrt{2}} \sqrt{c_{i,j} + 1}$$

Pearson correlation coefficient $r_{i,j}$

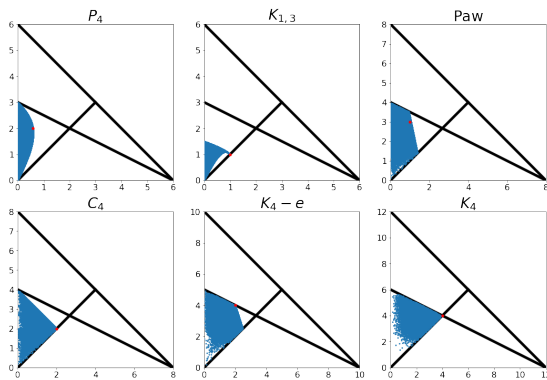
$$r_{ij} + 1 \quad \text{or} \quad \frac{1}{\sqrt{2}} \sqrt{r_{ij} + 1}$$

In the spectral embedding algorithm, how does the weight assignment change the embedding output?

The inverse problems

Let $\mathcal{S}_L(G)$ be the set of all weighted Laplacian matrix of G .

- 1 Characterize the spectra of matrices in $\mathcal{S}_L(G)$.
- 2 Characterize the eigenvector, especially the Fiedler vector, of matrices in $\mathcal{S}_L(G)$.
- 3 What about the normalized Laplacian matrices $D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$?



You need / will learn . . .

Foundation:

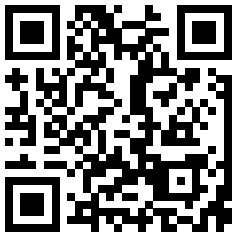
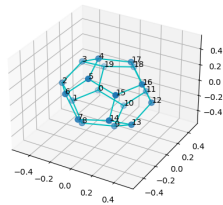
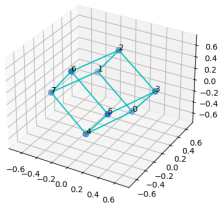
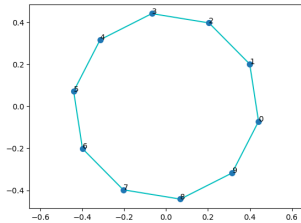
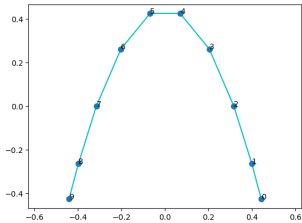
- Graph theory
- Matrix theory
- Programming

Advanced skills:

- Structural graph theory: Graph minor theorem, forbidden structures, . . .
- Matrix analysis: Inverse/implicit function theorem, manifold of matrices, . . .
- Real algebraic geometry: Gröbner basis, solving system of polynomial equations, . . .

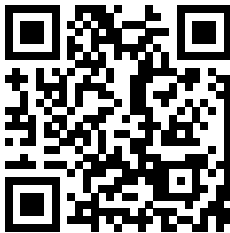
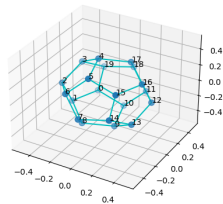
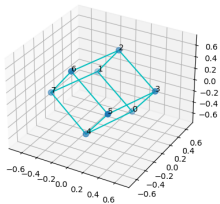
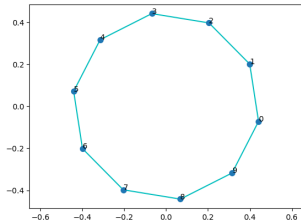
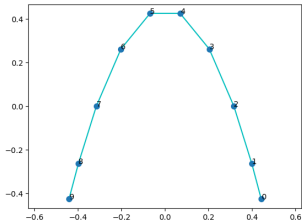
Applications:

- Graph drawing, graph partitioning, spectral clustering, . . .
- Learning on graphs, graph convolution neural network, . . .



Jephian Lin





Thank you!




Jephian Lin


Thank you!


References I


-  W. N. Anderson, Jr. and R. D. Morley.
Eigenvalues of the Laplacian of a graph.
University of Maryland Technical Report, TR-71-45, 1971.
-  M. Belkin and P. Niyogi.
Laplacian eigenmaps for dimensionality reduction and data representation.
Neural Computation, 15:1373–1396, 2003.
-  S. M. Fallat, H. Gupta, and J. C.-H. Lin.
Inverse eigenvalue problem for Laplacian matrices of a graph.
SIAM J. Matrix Anal. Appl., 46:1866–1886, 2025.
-  M. Fiedler.
Algebraic connectivity of graphs.
Czechoslovak Math. J., 23:298–305, 1973.

References II

 M. Fiedler.
Eigenvalues of acyclic matrices.
Czechoslovak Math. J., 25:607–618, 1975.

 M. Fiedler.
A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory.
Czechoslovak Math. J., 25:619–633, 1975.

 K. M. Hall.
An r -dimensional quadratic placement algorithm.
Management Science, 17:219–229, 1970.

 M. Juvan and B. Mohar.
Optimal linear labelings and eigenvalues of graphs.
Discrete Appl. Math., 36:153–168, 1992.

References III



Y. Koren.

Drawing graphs by eigenvectors: Theory and practice.

Computers & Mathematics with Applications, 49:1867–1888, 2005.



A. Pothen, H. D. Simon, and K.-P. Liou.

Partitioning sparse matrices with eigenvectors of graphs.

SIAM J. Matrix Anal. Appl., 11:430–452, 1990.



J. Shi and J. Malik.

Normalized cuts and image segmentation.

IEEE Transactions on Pattern Analysis and Machine Intelligence,
22:888–905, 2000.