Introduction to Spectral Graph Theory and Graph Clustering

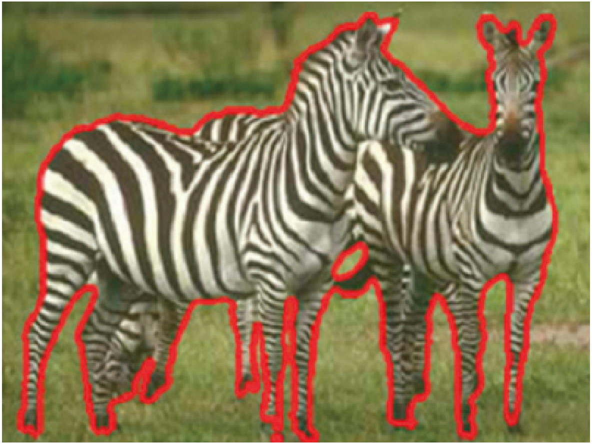
Chengming Jiang

ECS 231 Spring 2016

University of California, Davis

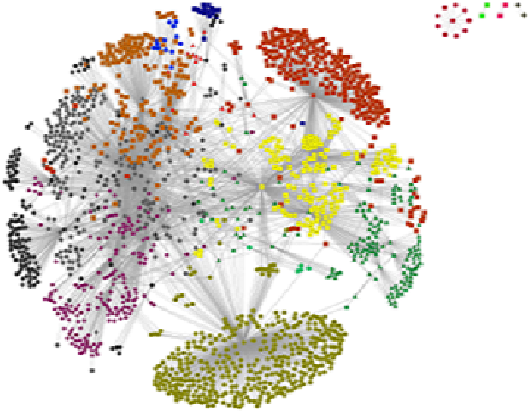
Motivation

Image partitioning in computer vision



Motivation

Community detection in network analysis



Outline

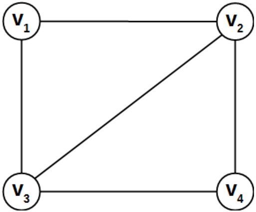
I. Graph and graph Laplacian
   - Graph
   - Weighted graph
   - Graph Laplacian

II. Graph clustering
   - Graph clustering
   - Normalized cut
   - Spectral clustering

I.1 Graph

An (undirected) **graph** is a pair $G = (V, E)$, where

- $\{v_i\}$ is a set of vertices;
- E is a subset of $V \times V$.



Remarks:

- An edge is a pair $\{u, v\}$ with $u \neq v$ (no self-loop);
- There is at most one edge from u to v (simple graph).

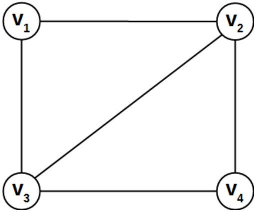
I.1 Graph

- For every vertex $v \in V$, the **degree** $d(v)$ of v is the number of edges adjacent to v:

$$d(v) = |\{u \in V | \{u, v\} \in E\}|.$$

- Let $d_i = d(v_i)$, the **degree matrix**

$$D = D(G) = \mathrm{diag}(d_1, \ldots, d_n).$$

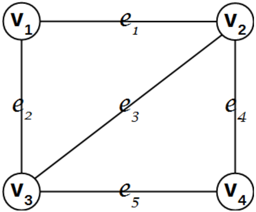


$$D = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}.$$

I.1 Graph

- Given a graph $G = (V, E)$, with $|V| = n$ and $|E| = m$, the **incidence matrix** $\tilde{D}(G)$ of G is an $n \times m$ matrix with

$$\tilde{d}_{ij} = \left\{ \begin{array}{ll} 1, & \text{if } \exists k \text{ s.t. } e_j = \{v_i, v_k\} \\ 0, & \text{otherwise} \end{array} \right) .$$

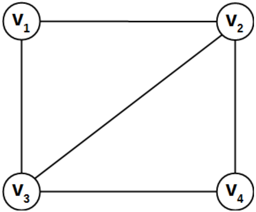


$$\tilde{D}(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

I.1 Graph

- Given a graph $G = (V, E)$, with $|V| = n$ and $|E| = m$, the **adjacency matrix** $A(G)$ of G is a symmetric $n \times n$ matrix with

$$a_{ij} = \begin{cases} 1, & \text{if } \{v_i, v_j\} \in E \\ 0, & \text{otherwise} \end{cases} .$$



$$A(G) = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

I.2 Weighted graph

A **weighted graph** is a pair $G = (V, W)$ where

- $V = \{v_i\}$ is a set of vertices and $|V| = n$;
- $W \in \mathbb{R}^{n \times n}$ is called *weight matrix* with

$$w_{ij} = \begin{cases} w_{ji} \geq 0 & i \neq j \\ 0 & i = j \end{cases} .$$

The **underlying graph** of G is $\widehat{G} = (V, E)$ with

$$E = \{\{v_i, v_j\} | w_{ij} > 0\}.$$

- If $w_{ij} \in \{0, 1\}$, $W = A$, the adjacency matrix of $\widehat{G}$.
- Since $w_{ii} = 0$, there is no self-loops in $\widehat{G}$.

I.2 Weighted graph

- For every vertex $v_i \in V$, the **degree** $d(v_i)$ of v_i is the sum of the weights of the edges adjacent to v_i:

$$d(v_i) = \sum_{j=1}^{n} w_{ij}.$$

- Let $d_i = d(v_i)$, the **degree matrix**

$$D = D(G) = \text{diag}(d_1, \ldots, d_n).$$

Remark:
Let $\boldsymbol{d} = \text{diag(D)}$ and denote $\mathbf{1} = (1, \ldots, 1)^T$, then $\boldsymbol{d} = W\mathbf{1}$.

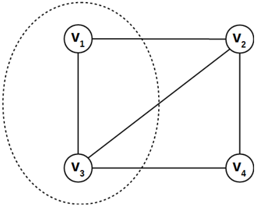
I.2 Weighted graph

- Given a subset of vertices $A \subseteq V$, we define the **volume** $\text{vol}(A)$ by

$$\text{vol}(A) = \sum_{v_i \in A} d(v_i) = \sum_{v_i \in A} \sum_{j=1}^{n} w_{ij}.$$

Remarks:

- If $\text{vol}(A) = 0$, all the vertices in A are isolated.
- Example:



If $A = \{v_1, v_3\}$, then

$$\begin{aligned}\text{vol}(A) &= d(v_1) + d(v_3)\\ &= (w_{12} + w_{13}) + \\ &\quad (w_{31} + w_{32} + w_{34})\end{aligned}$$

I.2 Weighted graph

- Given two subsets of vertices $A, B \subseteq V$, we define the **links** links(A, B) by

$$\text{links}(A, B) = \sum_{v_i \in A, v_j \in B} w_{ij}.$$

Remarks:

- A and B are not necessarily distinct;
- Since W is symmetric, links$(A, B) = $ links(B, A);
- vol$(A) = $ links(A, V).

I.2 Weighted graph

- The quantity $\mathrm{cut}(A)$ is defined by

$$\mathrm{cut}(A) = \mathrm{links}(A, V - A).$$

- The quantity $\mathrm{assoc}(A)$ is defined by

$$\mathrm{assoc}(A) = \mathrm{links}(A, A).$$

Remarks:

- $\mathrm{cut}(A)$ measures how many links escape from A;
- $\mathrm{assoc}(A)$ measures how many links stay within A;
- $\mathrm{cut}(A) + \mathrm{assoc}(A) = \mathrm{vol}(A)$.

I.3 Graph Laplacian

Given a weighted graph $G = (V, W)$, the (graph) **Laplacian** L of G is defined by

$$L = D - W.$$

where D is the degree matrix of G.

Remark

- $D = \text{diag}(W \cdot \mathbf{1})$.

I.3 Graph Laplacian

Properties of Laplacian

1. $x^T L x = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(x_i - x_j)^2$ for $\forall x \in \mathbb{R}^n$.
2. $L \geq 0$ if $w_{ij} \geq 0$ for all i, j;
3. $L \cdot \mathbf{1} = \mathbf{0}$;
4. If the underlying graph of G is connected, then

$$0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \ldots \leq \lambda_n;$$

5. If the underlying graph of G is connected, then the dimension of the nullspace of L is 1.

I.3 Graph Laplacian

Proofs:

Property 1. Since $L = D - W$, we have

$$\begin{aligned}
x^T L x &= x^T D x - x^T W x \\
&= \sum_{i=1}^{n} d_i x_i^2 - \sum_{i,j=1}^{n} w_{ij} x_i x_j \\
&= \frac{1}{2} \Big(\sum_{i}^{n} d_i x_i^2 - 2 \sum_{i,j=1}^{n} w_{ij} x_i x_j + \sum_{j=1}^{n} d_j x_j^2 \Big) \\
&= \frac{1}{2} \Big(\sum_{i,j=1}^{n} w_{ij} x_i^2 - 2 \sum_{i,j=1}^{n} w_{ij} x_i x_j + \sum_{i,j=1}^{n} w_{ij} x_j^2 \Big) \\
&= \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (x_i - x_j)^2.
\end{aligned}$$

I.3 Graph Laplacian

Property 2.

- Since $L^T = D - W^T = D - W = L$, L is symmetric.
- Since $x^T Lx = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(x_i - x_j)^2$ and $w_{ij} \geq 0$ for all i, j, we have $x^T Lx \geq 0$.

Property 3.

$$L \cdot \mathbf{1} = (D - W)\mathbf{1} = D\mathbf{1} - W\mathbf{1} = \boldsymbol{d} - \boldsymbol{d} = \mathbf{0}.$$

Property 4 and **Property 5** skip for now, see §2.2 of [Gallier'14].

Outline

I. Graph and graph Laplacian
- Graph
- Weighted graph
- Graph Laplacian

II. Graph clustering
- Graph clustering
- Normalized cut
- Spectral clustering

II.1 Graph clustering

k-**way partitioning:** given a weighted graph $G = (V, W)$, find a partition $A_1, A_2, \ldots, A_k$ of V, such that

- $A_1 \cup A_2 \cup \ldots \cup A_k = V$;
- $A_1 \cap A_2 \cap \ldots \cap A_k = \varnothing$;
- for any i and j, the edges between (A_i, A_j) have low weight and the edges within A_i have high weight.

If $k = 2$, it is a *two-way partitioning*.

II.1 Graph clustering

- Recall: (two-way) cut:

$$\mathsf{cut}(A) = \mathsf{links}(A, \bar{A}) = \sum_{v_i \in A, v_j \in \bar{A}} w_{ij}$$

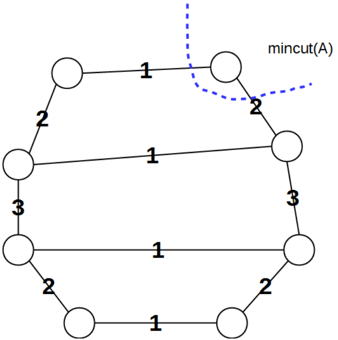
where $\bar{A} = V - A$.

II.1 Graph clustering problems

The **mincut** is defined by

$$\min \mathsf{cut}(A) = \min \sum_{v_i \in A, v_j \in \bar{A}} w_{ij}.$$

In practice, the mincut easily yields unbalanced partitions.



$$\min \mathsf{cut}(A) = 1 + 2 = 3;$$

II.2 Normalized cut

The **normalized cut**[1] is defined by

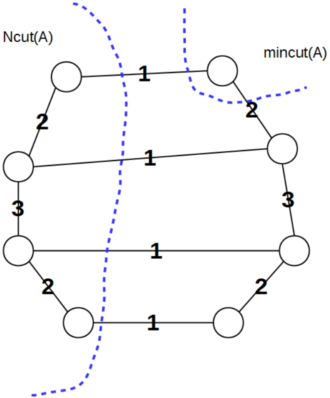
$$\mathsf{Ncut}(A) = \frac{\mathsf{cut}(A)}{\mathsf{vol}(A)} + \frac{\mathsf{cut}(\bar{A})}{\mathsf{vol}(\bar{A})}.$$

[1] Jianbo Shi and Jitendra Malik, 2000

II.2 Normalized cut

Minimal Ncut:

$$\min \mathsf{Ncut}(A).$$



$$\min \mathsf{Ncut}(A) = \frac{4}{3+6+6+3} + \frac{4}{3+6+6+3} = \frac{4}{9}.$$

II.2 Normalized cut

1. Let $x = (x_1, \ldots, x_n)$ be the *indicator vector*, such that

$$x_i = \begin{cases} 1 & \text{if } v_i \in A \\ -1 & \text{if } v_i \in \bar{A} \end{cases} .$$

2. Then
   $(\mathbf{1}+x)^T D(\mathbf{1}+x) = 4\sum_{v_i \in A} d_i = 4 \cdot \mathrm{vol}(A)$;
   $(\mathbf{1}+x)^T W(\mathbf{1}+x) = 4\sum_{v_i \in A, v_j \in A} w_{ij} = 4 \cdot \mathrm{assoc}(A)$.
   $(\mathbf{1}+x)^T L(\mathbf{1}+x) = 4 \cdot (\mathrm{vol}(A) - \mathrm{assoc}(A)) = 4 \cdot \mathrm{cut}(A)$;
   and
   $(\mathbf{1}-x)^T D(\mathbf{1}-x) = 4\sum_{v_i \in \bar{A}} d_i = 4 \cdot \mathrm{vol}(\bar{A})$;
   $(\mathbf{1}-x)^T W(\mathbf{1}-x) = 4\sum_{v_i \in \bar{A}, v_j \in \bar{A}} w_{ij} = 4 \cdot \mathrm{assoc}(\bar{A})$.
   $(\mathbf{1}-x)^T L(\mathbf{1}-x) = 4 \cdot (\mathrm{vol}(\bar{A}) - \mathrm{assoc}(\bar{A})) = 4 \cdot \mathrm{cut}(\bar{A})$.
   (Please verify it after class.)

II.2 Normalized cut

3. Ncut(A) can now be written as

$$\begin{aligned}\text{Ncut}(A) &= \frac{1}{4}\left(\frac{(\mathbf{1}+x)^T L(\mathbf{1}+x)}{k\mathbf{1}^T D\mathbf{1}} + \frac{(\mathbf{1}-x)^T L(\mathbf{1}-x)}{(1-k)\mathbf{1}^T D\mathbf{1}}\right) \\ &= \frac{1}{4}\cdot\frac{((\mathbf{1}+x)-b(\mathbf{1}-x))^T L((\mathbf{1}+x)-b(\mathbf{1}-x))}{b\mathbf{1}^T D\mathbf{1}}.\end{aligned}$$

   where $k = \text{vol}(A)/\text{vol}(V)$, $b = k/(1-k)$ and $\text{vol}(V) = \mathbf{1}^T D\mathbf{1}$.

4. Let $y = (\mathbf{1}+x) - b(\mathbf{1}-x)$, we have

$$\text{Ncut}(A) = \frac{1}{4}\cdot\frac{y^T L y}{b\mathbf{1}^T D\mathbf{1}}$$

   where

$$y_i = \begin{cases} 2 & \text{if } v_i \in A \\ -2b & \text{if } v_i \in \bar{A} \end{cases}.$$

II.2 Normalized cut

5. Since $b = k/(1-k) = \mathsf{vol}(A)/\mathsf{vol}(\bar{A})$, we have

$$\frac{1}{4}y^T D y = \sum_{v_i \in A} d_i + b^2 \sum_{v_i \in \bar{A}} d_i = \mathsf{vol}(A) + b^2 \mathsf{vol}(\bar{A})$$
$$= b(\mathsf{vol}(\bar{A}) + \mathsf{vol}(A)) = b\mathbf{1}^T D\mathbf{1}.$$

In addition,

$$y^T D\mathbf{1} = y^T \boldsymbol{d} = 2 \cdot \sum_{v_i \in A} d_i - 2b \cdot \sum_{v_i \in \bar{A}} d_i$$
$$= 2 \cdot \mathsf{vol}(A) - 2b \cdot \mathsf{vol}(\bar{A}) = 0$$

II.2 Normalized cut

6. The **minimal normalized cut** is to solve the following **binary optimization**:

$$
\begin{aligned}
y = \arg\min_{y} \quad & \frac{y^T L y}{y^T D y} \\
s.t. \quad & y(i) \in \{2, -2b\} \\
& y^T D \mathbf{1} = 0
\end{aligned}
\tag{1}
$$

7. Relaxation

$$
\begin{aligned}
y = \arg\min_{y} \quad & \frac{y^T L y}{y^T D y} \\
s.t. \quad & y \in \mathbb{R}^n \\
& y^T D \mathbf{1} = 0
\end{aligned}
\tag{2}
$$

II.2 Normalized cut

Variational principle

- Let $A, B \in \mathbb{R}^{n \times n}$, $A^T = A$, $B^T = B > 0$ and $\lambda_1 \leq \lambda_2 \leq \ldots \lambda_n$ be the eigenvalues of $Au = \lambda Bu$ with corresponding eigenvectors $u_1, u_2, \ldots, u_n$,
- then

$$\min_x \frac{x^T Ax}{x^T Bx} = \lambda_1 \ , \quad \arg\min_x \frac{x^T Ax}{x^T Bx} = u_1$$

  and

$$\min_{x^T Bu_1 = 0} \frac{x^T Ax}{x^T Bx} = \lambda_2 \ , \quad \arg \min_{x^T Bu_1 = 0} \frac{x^T Ax}{x^T Bx} = u_2.$$

- More general form exists.

II.2 Normalized cut

- For the matrix pair (L, D), it is known that $(\lambda_1, y_1) = (0, \mathbf{1})$.
- Therefore, by the variational principle, the relaxed minimal Ncut problem (2) is equivalent to finding the second smallest eigenpair (λ_2, y_2) of

$$Ly = \lambda Dy \tag{3}$$

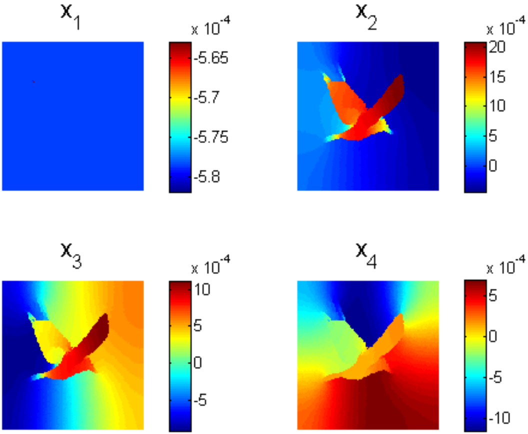
- L is extremely sparse and D is diagonal;
- Precision requirement for eigenvectors is low, say $\mathcal{O}(10^{-4})$.

II.2 Normalized cut

Image segmentation: original graph

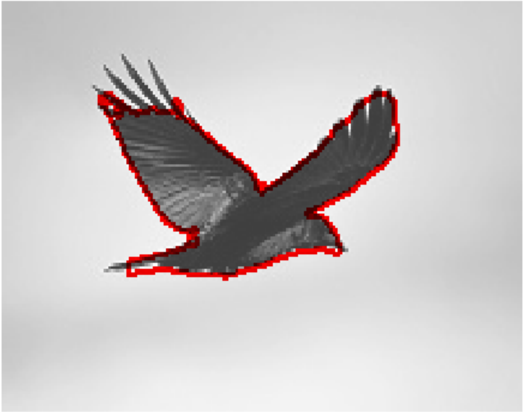
II.2 Normalized cut

Image segmentation: heatmap of eigenvectors



II.2 Normalized cut

Image segmentation $\min$ **Ncut**



Ncut remaining issues

- Once the indicator vector is computed, how to search the splitting point that the resulting partition has the minimal Ncut(A) value?
- How to use the extreme eigenvectors to do the k-way partitioning?

The above two problems are addressed in spectral clustering algorithm.

II.3 Spectral clustering

Spectral clustering algorithm [Ng et al, 2002]

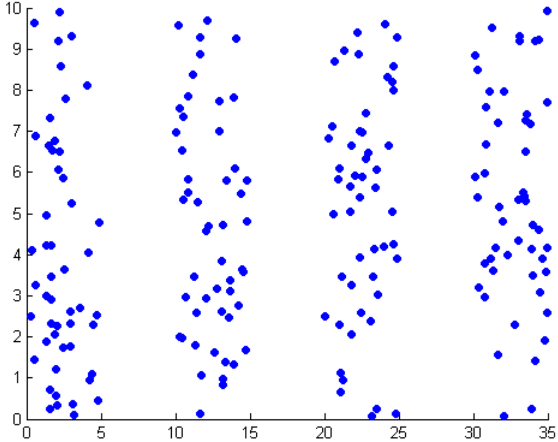
Given a weighted graph $G = (V, W)$,

1. *compute the normalized Laplacian* $L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$*;*
2. *find k eigenvectors $X = [x_1, \ldots, x_k]$ corresponding to the k smallest eigenvalues of L;*
3. *form $Y \in \mathbb{R}^{n \times k}$ by normalizing each row of X such that $Y(i,:) = X(i,;)/\|X(i,:)\|$;*
4. *treat each $Y(i,:)$ as a point, cluster them into k clusters via K-means with label $c_i = \{1, \ldots, k\}$.*

The label c_i indicates the cluster that v_i belongs to.

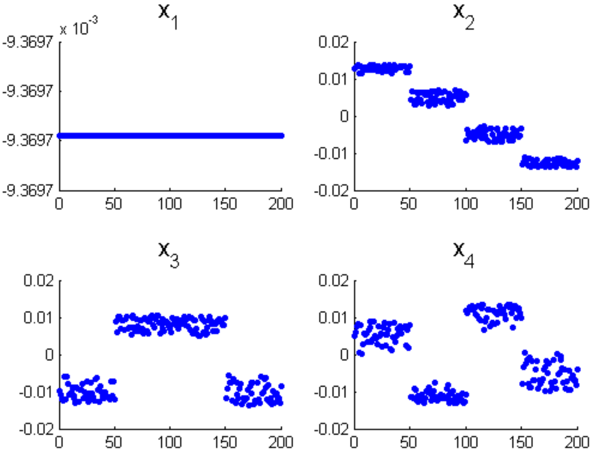
II.3 Spectral clustering

Synthetic example: original data



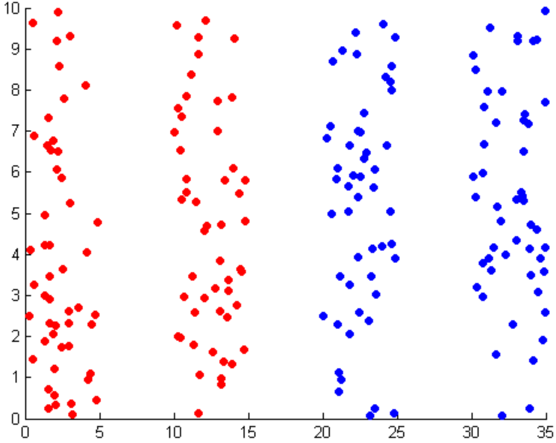
II.3 Spectral clustering

Synthetic example: computed eigenvectors



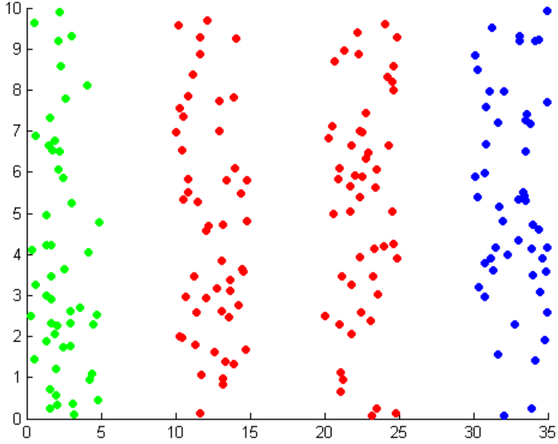
II.3 Spectral clustering

Synthetic example: 2-way clustering



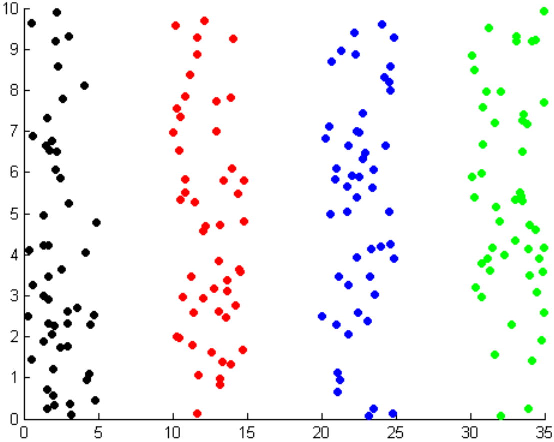
II.3 Spectral clustering

Synthetic example: 3-way clustering



II.3 Spectral clustering

Synthetic example: 4-way clustering



Further reading

1. Jean Gallier, *Notes on elementary spectral graph theory applications to graph clustering using normalized cuts*, 2013.
2. Jianbo Shi and Jitendra Malik, *Normalized cuts and image segmentation*, 2000.
3. Andrew Y Ng, Michael I. Jordan and Yair Weiss, *On spectral clustering: Analysis and an algorithm*, 2001