Learning from Data
Lecture 9: Unsupervised Learning I

Yang Li **yangli@sz.tsinghua.edu.cn**

TBSI

May 10, 2024

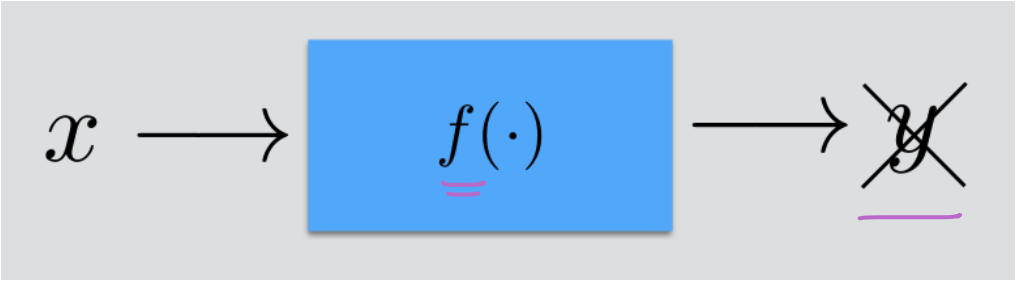
Today's Lecture

Unsupervised Learning

- Overview: the representation learning problem
- K-means clustering
- Spectral clustering

Unsupervised Learning Overview

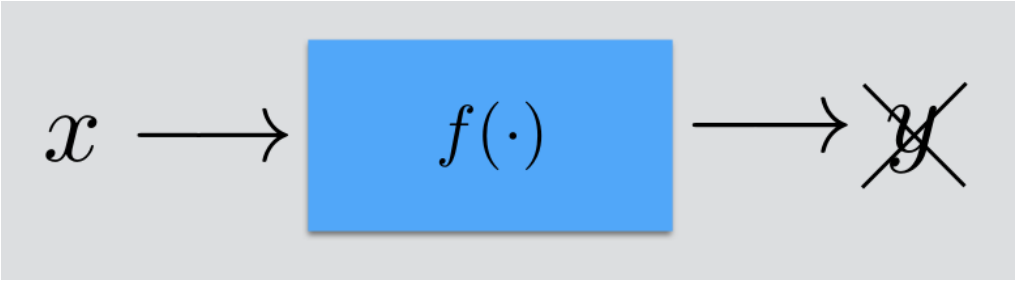
Unsupervised Learning



Similar to supervised learning, but without labels.

- Still want to learn the machine f
- Significantly harder in general

Unsupervised Learning



Similar to supervised learning, but without labels.

- Still want to learn the machine f
- Significantly harder in general

Unsupervised learning goal

Find **representations** of input feature x that can be used for reasoning, decision making, predicting things, comminicating etc.

The representation learning problem

(Y Bengio et. al. *Representation Learning: A Review and New Perspectives*, 2014)

Given input features x, find "simpler" features z that **preserve the same information** as x.

Example: Face recognition

100×100

100

100

$$\rightarrow x = \begin{bmatrix} 0.5 \\ 0 \\ \vdots \\ 0.3 \\ 1.0 \end{bmatrix} \Big\} 10^4 \rightarrow \qquad z = \begin{bmatrix} \vdots \end{bmatrix}$$

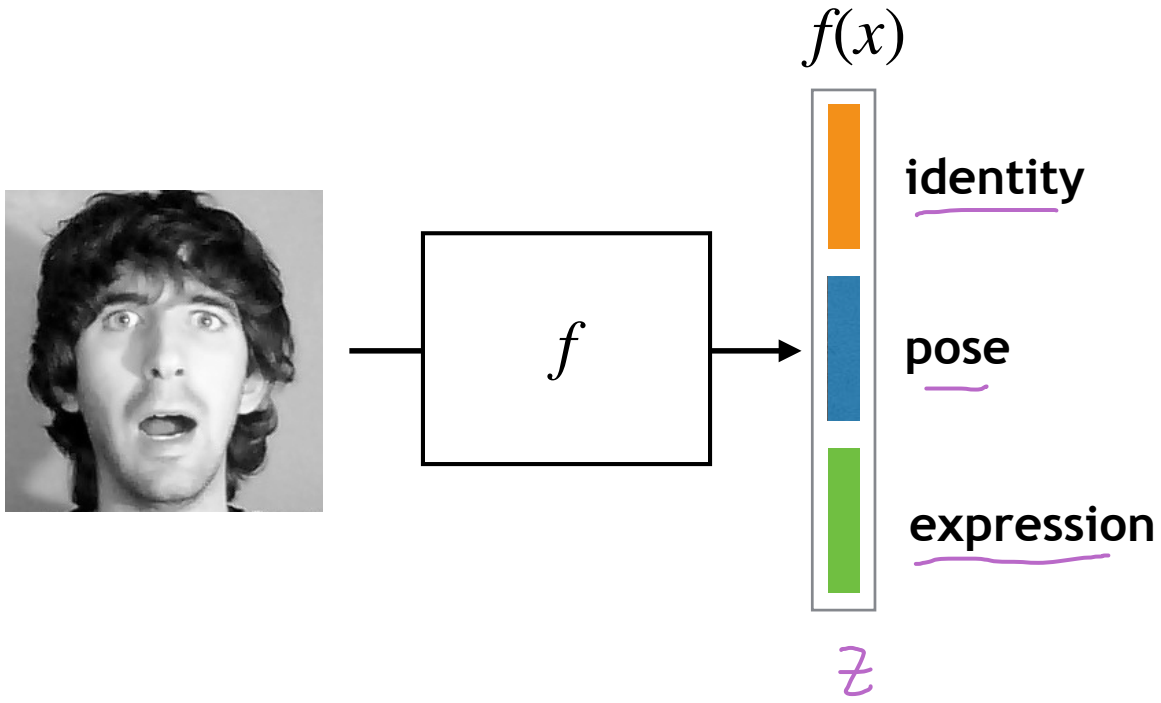
What information is in this picture? *identity, facial attributes, gender, age, sentiment, etc*

Characteristics of a good representation

$x^{(i)} \downarrow z^{(i)}$

$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$

- low dimensional: compress information to a smaller size → *reduce data size*
- sparse representation: most entries are zero for most data → *better interpretability*
- independent representations: disentangle the source of variations

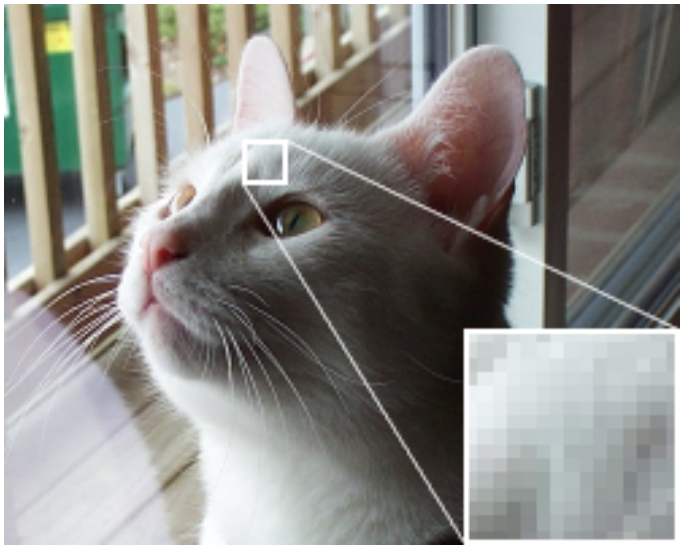


Uses of representation learning

- Data compression

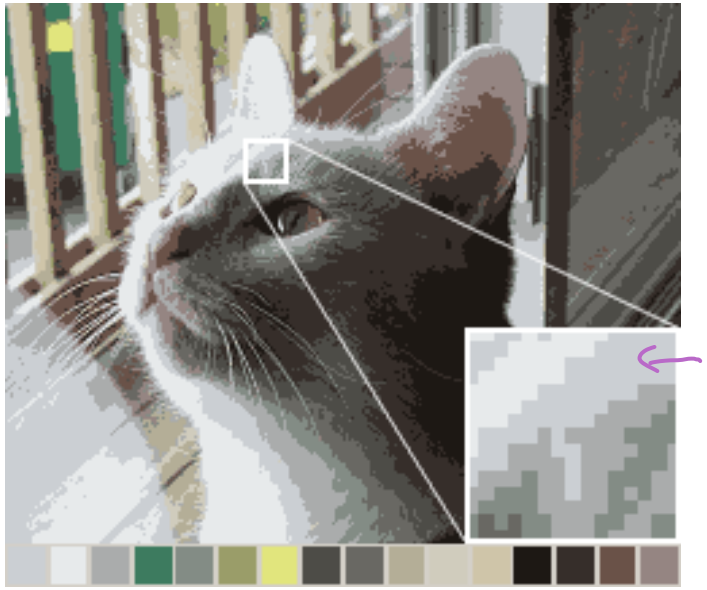
Example: Color image quantization. Each 24bit RGB color is reduced to a palette of 16 colors.

Original



(0-255,0-255,0-255)
24bit x 300 x 400

Compressed

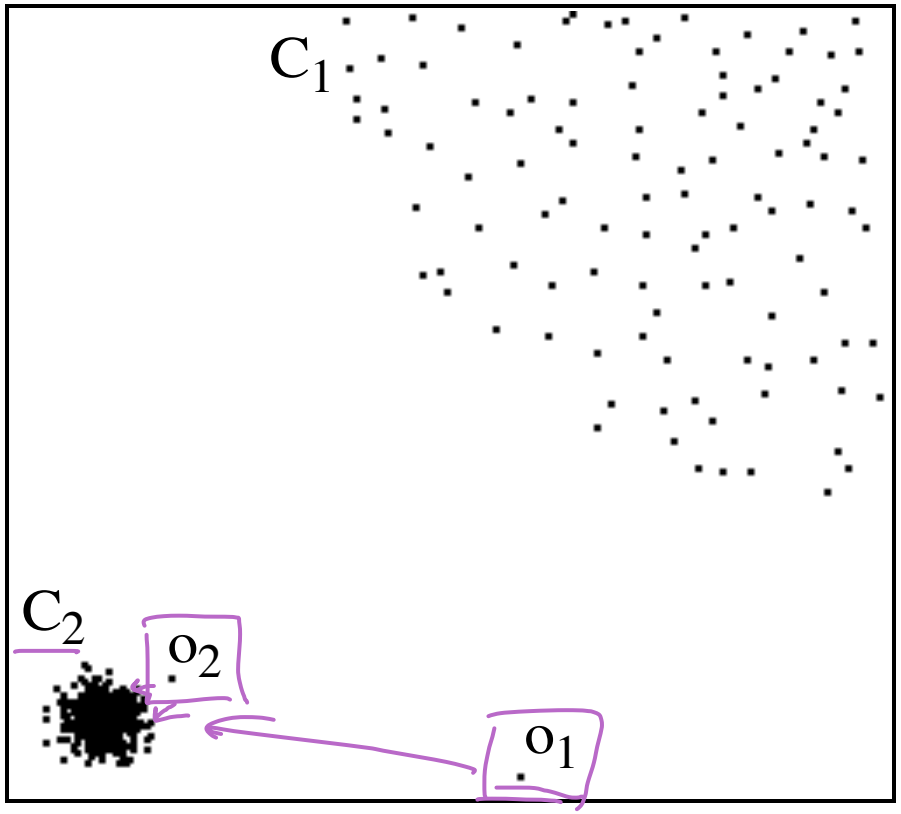


0-15 (R, G, B)
4bit x 300 x 400 + 16 x24bit
6 times smaller

Uses of representation learning

- Abnormality (outlier, novelty) detection

Example: local density-based outlier detection

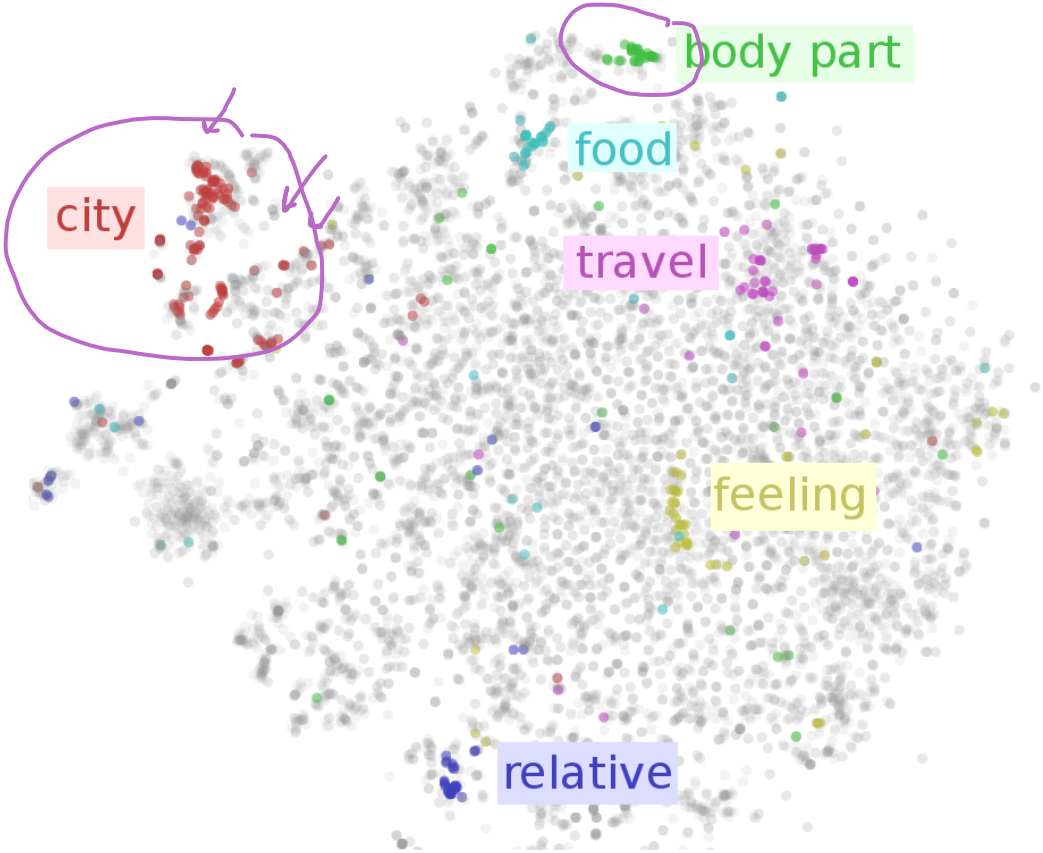


o_1 and o_2 are the detected outliers

Uses of representation learning

- Knowledge representation based on human perception

Exam~~ple: word embedding~~



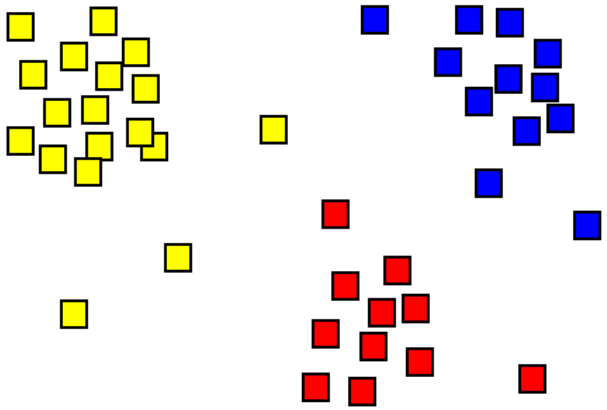
http://ruder.io/word-embeddings-1/

Each word is represented by a 2D vector. Words in the same semantic category are grouped together

K-Means Clustering

Clustering analysis

Given input features $\{x^{(1)}, \ldots, x^{(m)}\}$, group the data into a few *cohesive* "clusters".

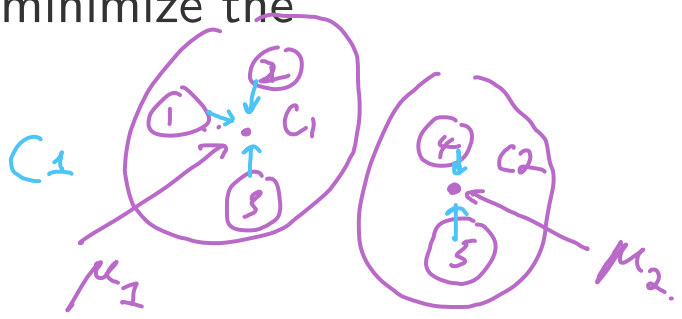


- Objects in the same cluster are more similar to each other than to those in other clusters

The k-means clustering problem

Given input data $X = \{x^{(1)}, \ldots, x^{(m)}\}$, $x^{(i)} \in \mathbb{R}^d$, **k-means clustering** partition the input into $k \leq m$ sets $C_1, \ldots, C_k$ to minimize the within-cluster sum of squares (WCSS).

$$\underset{C_1,\ldots,C_k}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{x \in C_j} \|x - \mu_j\|^2$$



$m = 5$.

$n = k$

find subsets $C_1, C_2 \subseteq X$

The k-means clustering problem

Given input data $\{x^{(1)}, \ldots, x^{(m)}\}$, $x^{(i)} \in \mathbb{R}^d$, **k-means clustering** partition the input into $k \leq m$ sets $C_1, \ldots, C_k$ to minimize the within-cluster sum of squares (WCSS).

$$\underset{C_1,\ldots,C_k}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{x \in C_j} \|x - \mu_j\|^2$$

Equivalent definitions:

- minimizing the within-cluster variance: $\sum_{j=1}^{k} |C_j| \operatorname{Var}(C_j)$

$$\operatorname{Var}(C_j) = \frac{1}{|C_j|} \cdot \sum_{x \in C_j} \|x - \mu_j\|^2$$

$$WCV = \sum_{j=1}^{k} \cancel{|C_j|} \cdot \frac{1}{\cancel{|C_j|}} \sum_{x \in C_j} \|x - \mu_j\|^2$$

$$= \sum_{j=1}^{k} \sum_{x \in C_j} \|x - \mu_j\|^2 = WCSS$$

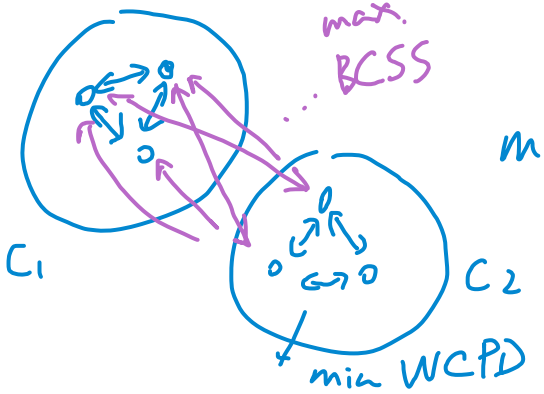
The k-means clustering problem

Given input data $\{x^{(1)}, \ldots, x^{(m)}\}$, $x^{(i)} \in \mathbb{R}^d$, **k-means clustering** partition the input into $k \leq m$ sets $C_1, \ldots, C_k$ to minimize the within-cluster sum of squares (WCSS).

$$\operatorname*{argmin}_{C_1,\ldots,C_k} \sum_{j=1}^{k} \sum_{x \in C_j} \|x - \mu_j\|^2$$

Equivalent definitions:

- minimizing the within-cluster variance: $\sum_{j=1}^{k} |C_j| \operatorname{Var}(C_j)$
- minimizing the pairwise squared deviation between points in the same cluster: *(homework)*



$$\min_{C} \sum_{i=1}^{k} \frac{1}{2|C_i|} \sum_{x,x' \in C_i} \|x - x'\|^2$$

The k-means clustering problem

Given input data $\{x^{(1)}, \ldots, x^{(m)}\}$, $x^{(i)} \in \mathbb{R}^d$, **k-means clustering** partition the input into $k \leq m$ sets $C_1, \ldots, C_k$ to minimize the within-cluster sum of squares (WCSS).

$$\operatorname*{argmin}_{C_1,\ldots,C_k} \sum_{j=1}^{k} \sum_{x \in C_j} \|x - \mu_j\|^2$$

Equivalent definitions:

- minimizing the within-cluster variance: $\sum_{j=1}^{k} |C_j| \operatorname{Var}(C_j)$
- minimizing the pairwise squared deviation between points in the same cluster: *(homework)*

$$\sum_{i=1}^{k} \frac{1}{2|C_i|} \sum_{x,x' \in C_i} \|x - x'\|^2$$

- maximizing between-cluster sum of squares (BCSS) *(homework)*

K-Means Clustering Algorithm

- Optimal k-means clustering is NP-hard in Euclidean space.
- Often solved via a heuristic, iterative algorithm

K-Means Clustering Algorithm

- Optimal k-means clustering is NP-hard in Euclidean space.
- Often solved via a heuristic, iterative algorithm

Lloyd's Algorithm (1957,1982)

Let $c^{(i)} \in \{1, \ldots, k\}$ be the cluster label for $x^{(i)}$

```
Initialize cluster centroids μ_1,...μ_k ∈ R^n randomly
Repeat until convergence{
  For every i,
    c^(i) := argmin_j ||x^(i) − μ_j||^2

  For each j
        μ_j := (Σ_{i=1}^m 1{c^(i)=j} x^(i)) / (Σ_{i=1}^m 1{c^(i)=j})
}
```

Demo:http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html

Lloyd, Stuart P. (1982). "Least squares quantization in PCM". IEEE Transactions on Information Theory

K-Means Clustering Algorithm

- Optimal k-means clustering is NP-hard in Euclidean space.
- Often solved via a heuristic, iterative algorithm

Lloyd's Algorithm (1957,1982)

Let $c^{(i)} \in \{1, \ldots, k\}$ be the cluster label for $x^{(i)}$

```
Initialize cluster centroids μ_1,...μ_k ∈ R^n randomly
Repeat until convergence{
  For every i,
    c^(i) := argmin_j ‖x^(i) − μ_j‖^2   ← assign x^(i) to the cluster
                                          with the closest centroid
  For each j
        μ_j := Σ_{i=1}^m 1{c^(i)=j} x^(i) / Σ_{i=1}^m 1{c^(i)=j}
}
```

$c^{(i)} \in \{0, 1, \ldots k\}$.

Demo:http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html

Lloyd, Stuart P. (1982). "Least squares quantization in PCM". IEEE Transactions on Information Theory

K-Means Clustering Algorithm

- Optimal k-means clustering is NP-hard in Euclidean space.
- Often solved via a heuristic, iterative algorithm

Lloyd's Algorithm (1957,1982)

Let $c^{(i)} \in \{1, \ldots, k\}$ be the cluster label for $x^{(i)}$

```
Initialize cluster centroids μ_1,...μ_k ∈ R^n randomly
Repeat until convergence{
  For every i,
    c^(i) := argmin_j ||x^(i) − μ_j||^2   ← assign x^(i) to the cluster
                                           with the closest centroid
  For each j
      μ_j := Σ_{i=1}^m 1{c^(i)=j} x^(i) / Σ_{i=1}^m 1{c^(i)=j}   ← update centroid
}
```

Demo:http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html

Lloyd, Stuart P. (1982). "Least squares quantization in PCM". IEEE Transactions on Information Theory

K-Means clustering discussion

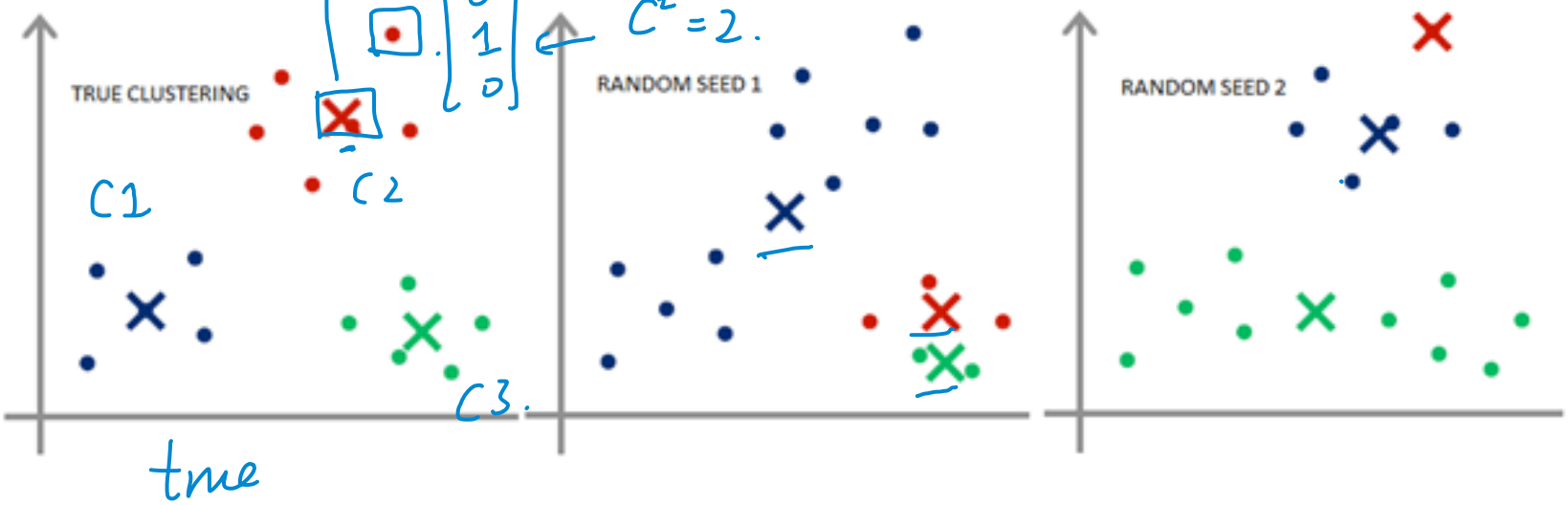
- K-Means learns a k-dimensional *sparse* representation.
  i.e. $x^{(i)}$ is transformed into a "one-hot" vector $z^{(i)} \in \mathbb{R}^k$:

$$z_j^{(i)} = \begin{cases} 1 & \text{if } c^{(i)} = j \\ 0 & \text{otherwise} \end{cases}$$

$\mu_2.$

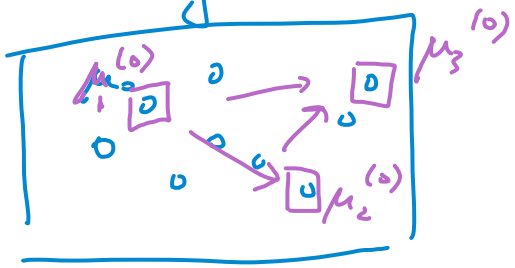
- Only converges to a local minimum: initialization matters!

$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ $c^i = 2.$



Practical considerations

- Replicate clustering trails and choose the result with the smallest WCSS
- How to initialize centroids μ_j's ?
  - Uniformly random sampling ☹
  - Distance-based sampling e.g. kmeans++ [Arthur & Vassilvitskii SODA 2007] ☺ → farthest point sampling
- How to choose k?
  - Cross validation (~~later lecture~~)
  - G-Means [Hamerly & Elkan, NIPS 2004]
- How to improve k-means efficiency?
  - Elkan's algorithm [Elkan, ICML 2003]
  - Mini-batch k-means [D. Sculley, WWW 2010]

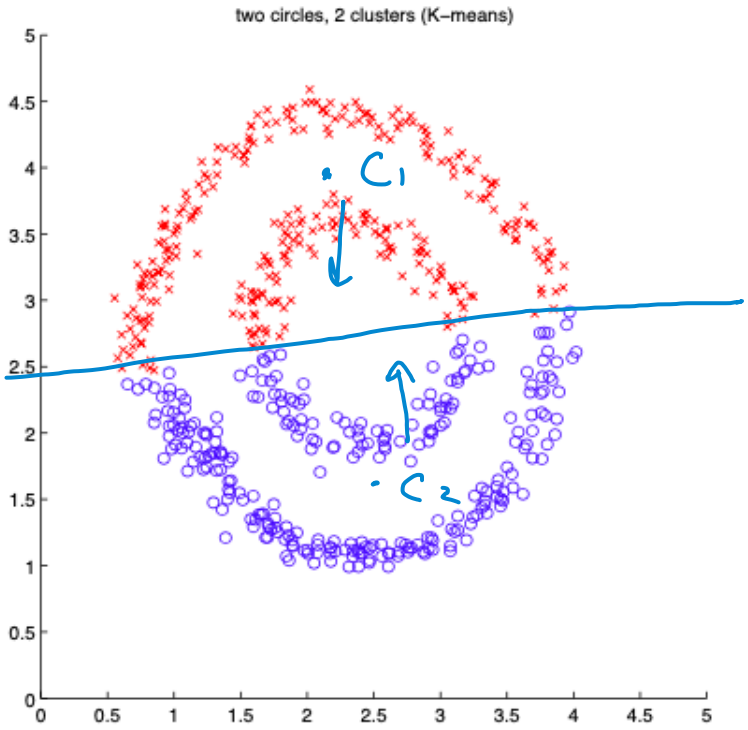


Spectral Graph Theory

- Graph Terminologies and Similarity Graphs
- Spectral Theory
- Spectral Clustering

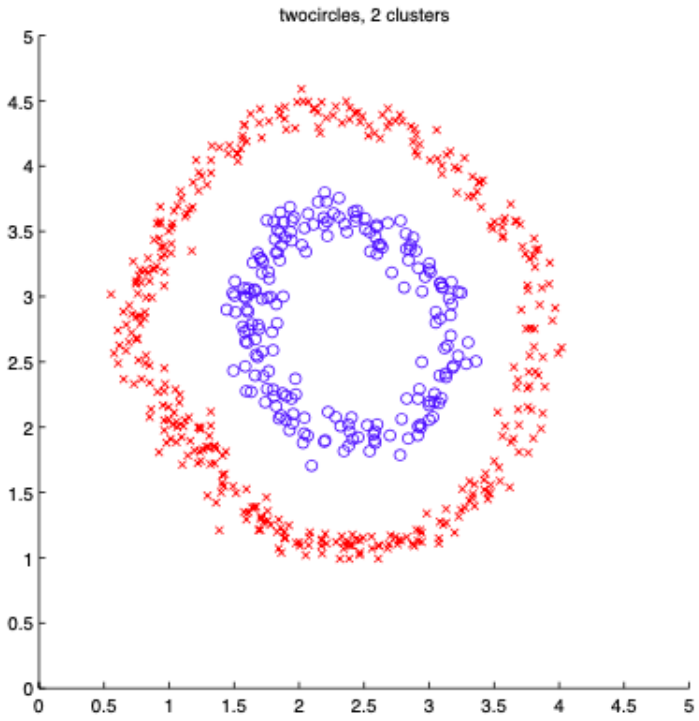
K-Means vs Spectral Clustering

K-Means



(i)

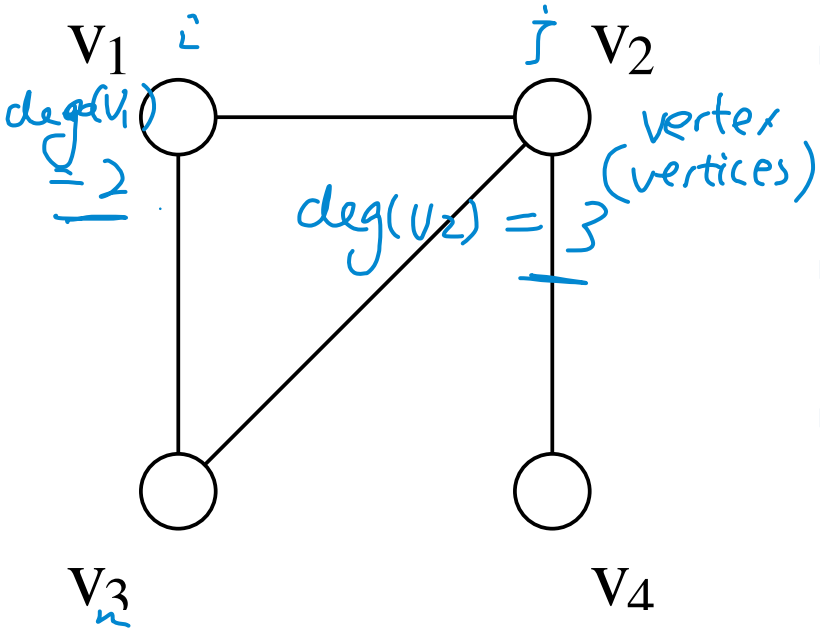
Spectral Clustering



(e)

[Shi & Malik 00; Ng, Jordan, Weiss NIPS 01]

Graph Terminologies



$$\sum_{j=1}^{n} w_{ij}$$

$$W = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{matrix} = 2 \\ = 3 \\ \\ \\ \end{matrix}$$

A

symmetric!

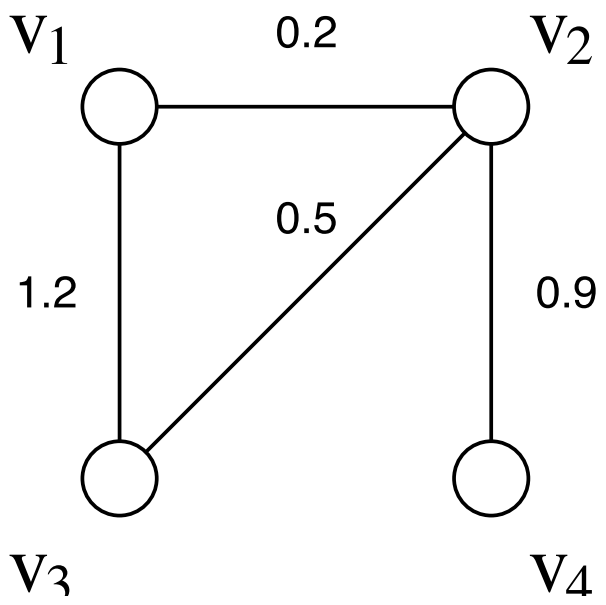
- An **undirect graph** $G = (V, E)$ consists of nodes $V = \{v_1, \ldots, v_n\}$ and edges $E = \{e_1, \ldots, e_m\}$
- Edge e_{ij} connects v_i and v_j if they are **adjacent** or neighbors.
- Adjacency matrix

$$W_{ij} = \begin{cases} 1 & \text{if there is an edge } e_{ij} \\ 0 & \text{otherwise} \end{cases}$$

- **Degree** d_i of node v_i is the number of neighbors of v_i.

$$d_i = \sum_{j=1}^{n} w_{ij}$$

Graph Terminologies



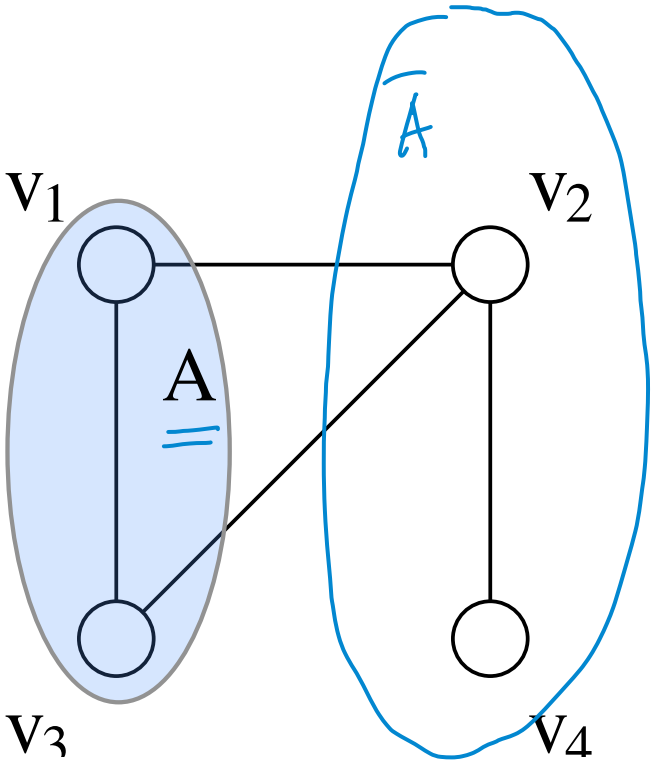
$$W = \begin{bmatrix} 0 & 0.2 & 1.2 & 0 \\ 0.2 & 0 & 0.5 & 0.9 \\ 1.2 & 0.5 & 0 & 0 \\ 0 & 0.9 & 0 & 0 \end{bmatrix} \begin{matrix} =1.4 \\ 1.6 \\ 1.7 \\ 0.9 \end{matrix}$$

- **Weigted undirected graph** $G = (V, E, W)$
- Edge weight $w_{ij} \in \mathbb{R}_{\geq 0}$ between v_i and v_j
  edge (v_i, v_j) exists iff $w_{ij} > 0$
- **Weighted adjacency matrix** $W = [w_{ij}]$
- Vertex degree $d_i = \sum_{j=1}^{n} w_{ij}$
- **Degree matrix** $D = diag(d_1, \ldots, d_n)$

$$D = \begin{bmatrix} 1.4 & & & \\ & 1.6 & & \\ & & 1.7 & \\ & & & 0.9 \end{bmatrix}$$

Graph Terminologies

$$\mathbf{1}_A = \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad \mathbf{1}_{\bar{A}} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$



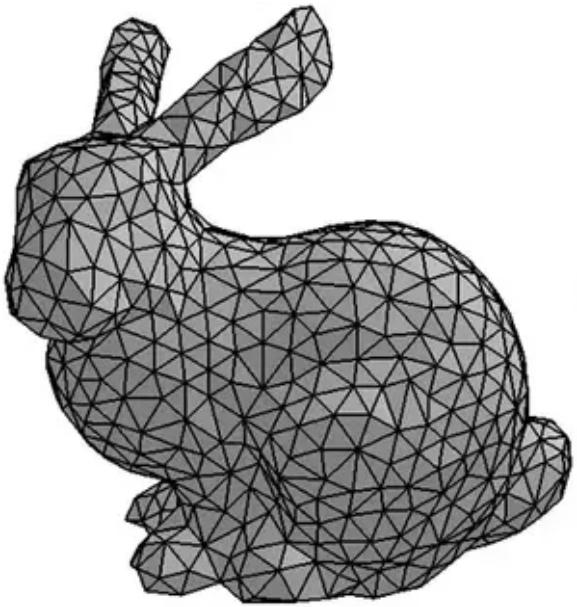
- Given vertex subset $A \subset V$, let $\bar{A} = V \backslash A$ be the complement of A in the graph
- Subset indicator function $\mathbf{1}_A \in \mathbb{R}^n$:

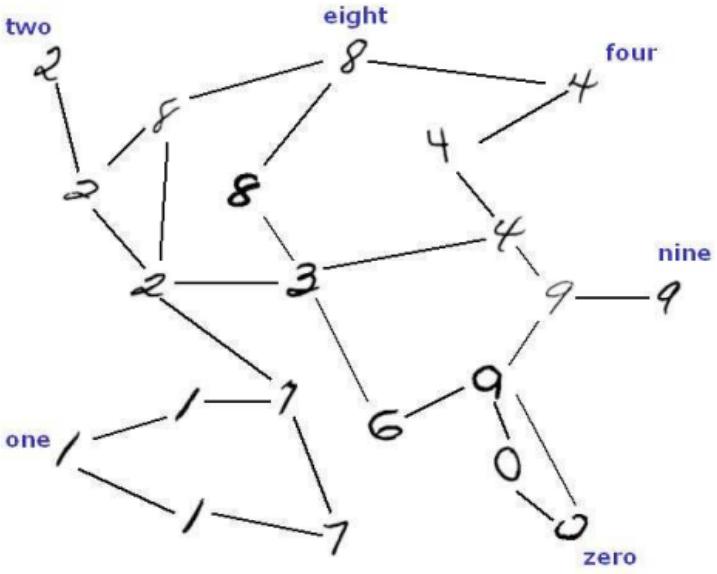
$$1_A\{i\} = \begin{cases} 1 & \text{if } v_i \in A \\ 0 & \text{if } v_i \notin A \end{cases}$$

- Sets $A_1, \ldots, A_k$ form a **partition** of the graph if $A_i \cap A_j = \varnothing$ for all $i \neq j$ and $A_1 \cup \cdots \cup A_k = V$

Represent data using a graph

Some data are naturally represented by a graph e.g. social networks, 3D mesh etc





Use graph to represent similarity in data

Clustering from a graph point of view

- Given data points $x^{(1)}, \ldots, x^{(n)}$ and **similarity measure** $s_{ij} \geq 0$ for all $x^{(i)}, x^{(j)}$
- A typical **similarity graph** $G = (V, E)$ is
  - $v_i \leftrightarrow x^{(i)}$
  - v_i and v_j are connected if $s_{ij} \geq \delta$ for some threshold δ
- **Clustering**: Divide data into groups such that points in the same group are similar and points in different groups are dissimilar
- **Spectral Clustering (informal)**: *Find a partition of G such that edges between the same group have high weights and edges between different groups have very low weights.*

similarity

Building similarity graphs from data

ϵ-neighborhood

Add edges to all points inside a ball of radius f centered at v

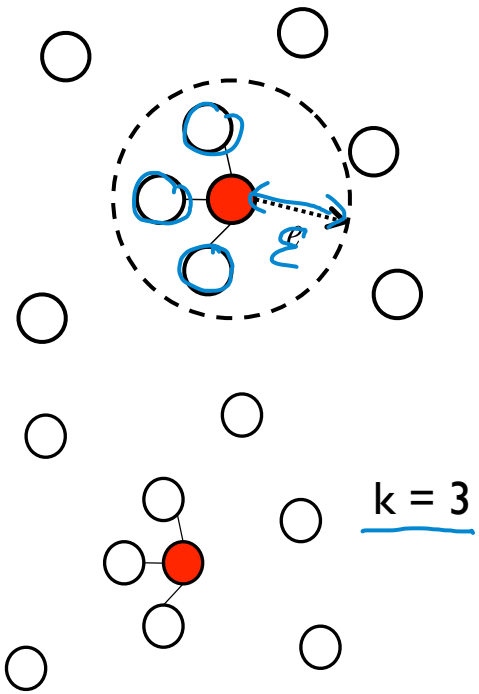
k-Nearest Neighbors

Add edges between v's k-nearest neighbors.

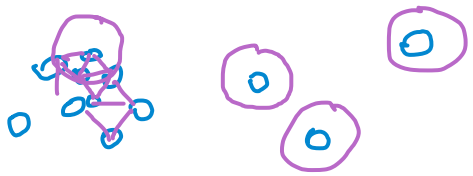
Fully connected graph

Often, Gaussian similarity is used

$$W_{i,j} = \exp\left(-\frac{\|x^{(i)} - x^{(j)}\|_2^2}{2\sigma^2}\right) \text{ for } i, j = 1, \ldots, m$$



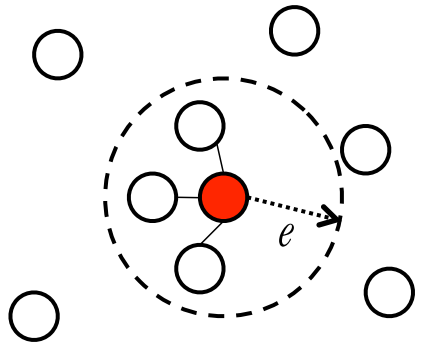
Building similarity graphs from data



ϵ-neighborhood

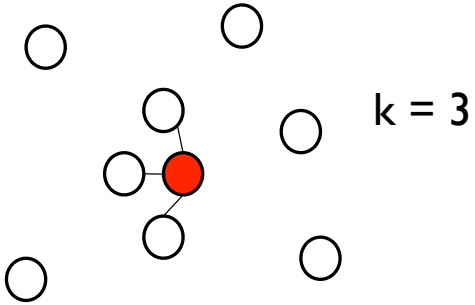
Add edges to all points inside a ball of radius f centered at v

Drawbacks: sensitive to ε, edge weights are on similar scale



k-Nearest Neighbors

Add edges between v's k-nearest neighbors.



Fully connected graph

Often, Gaussian similarity is used

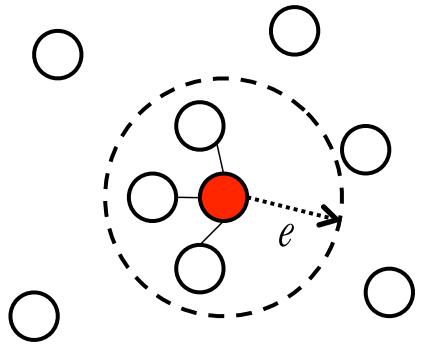
$$W_{i,j} = \exp\left(-\frac{\|x^{(i)} - x^{(j)}\|_2^2}{2\sigma^2}\right) \text{ for } i,j = 1,\ldots,m$$

Building similarity graphs from data

ϵ-neighborhood

Add edges to all points inside a ball of radius f centered at v

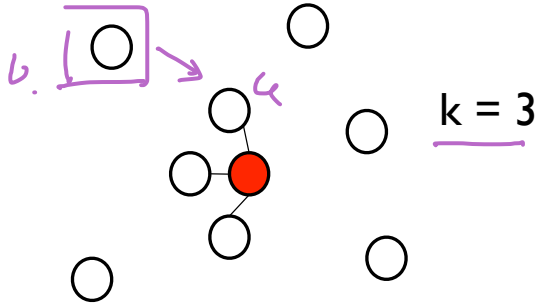
Drawbacks: sensitive to e, edge weights are on similar scale



k-Nearest Neighbors

Add edges between v's k-nearest neighbors.

Drawbacks: may result in asymmetric and irregular graph



1-KNN

Fully connected graph

Often, Gaussian similarity is used

$$W_{i,j} = \exp\left(-\frac{\|x^{(i)} - x^{(j)}\|_2^2}{2\sigma^2}\right) \text{ for } i, j = 1, \ldots, m$$

- "or"
- "and"

$$e_{ij} = \begin{cases} 1 & v_i \in N_k(v_j) \ \boxed{\text{or}} \rightarrow \text{"and"} \ v_j \in N_k(v_i) \\ 0 & \text{o.w.} \end{cases}$$

Building similarity graphs from data

ϵ-neighborhood

Add edges to all points inside a ball of radius f centered at v

Drawbacks: sensitive to e, edge weights are on similar scale

k-Nearest Neighbors

Add edges between v's k-nearest neighbors.

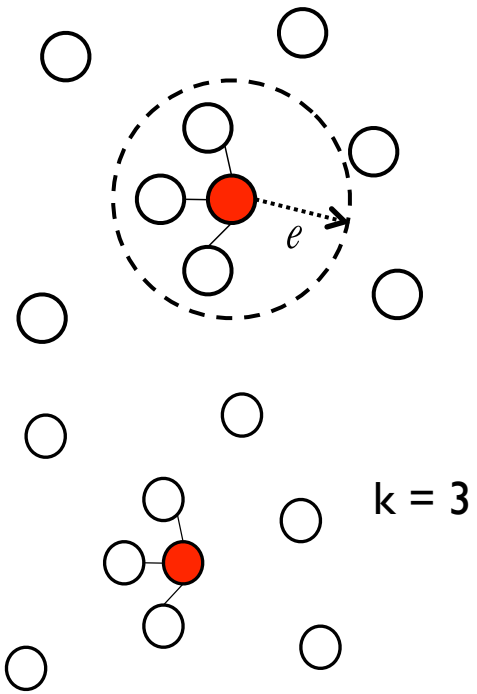
Drawbacks: may result in asymmetric and irregular graph

Fully connected graph

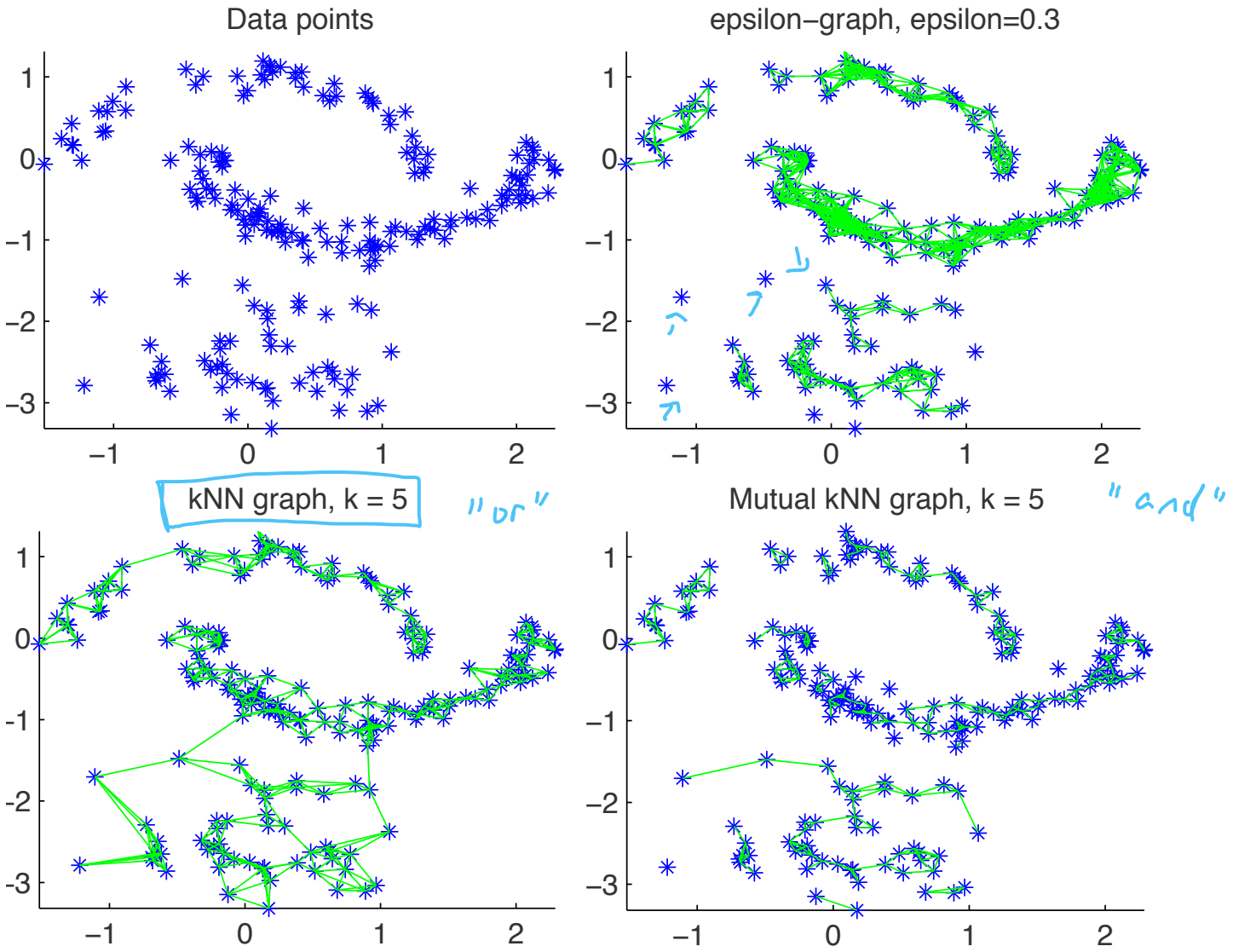
Often, Gaussian similarity is used

$$W_{i,j} = \exp\left(-\frac{\|x^{(i)} - x^{(j)}\|_2^2}{2\sigma^2}\right) \text{ for } i, j = 1, \ldots, m$$

Drawbacks: W is not sparse



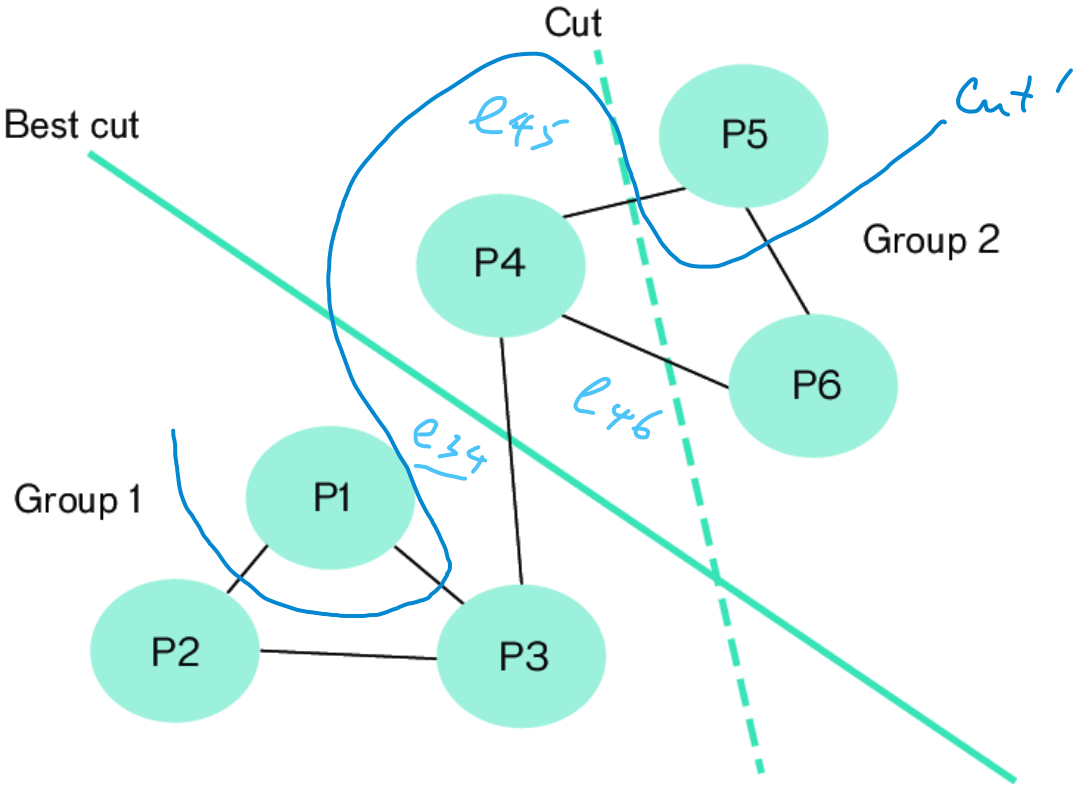
Similarity graphs examples



Spectral Clustering as Graph Partitioning

Find a partition of the graph such that

- Edges between groups have low weights
- Edges within each group have high weights



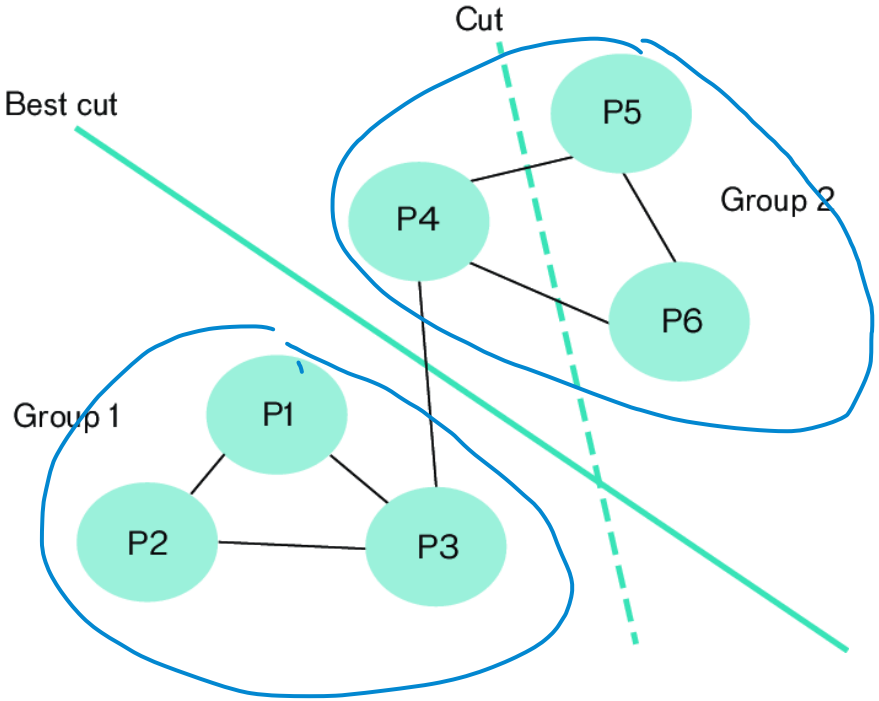
Graph Cut Formulation

Case $k = 2$:

- Given partition $A, \bar{A}$, define a cut as the total weight of edges between A and $\bar{A}$:

$$cut(A, \bar{A}) := \sum_{i \in A, j \in \bar{A}} w_{ij}$$

- Example: $cut(\{p_1, p_2, p_3\}, \{p_4, p_5, p_6\}) = 1$, $cut(\{p_1, p_2, p_3, p_4\}, \{p_5, p_6\}) = 2$



Graph Cut Formulations

Case $k > 2$:

- Given partition $A_1, \ldots, A_k$, define a cut as the total edges weights between groups:

$$cut(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} cut(A_i, \bar{A}_i)$$

$k=2$.

$cut(A_1, A_2)$

$= \frac{1}{2}\left(cut(A_1, \boxed{\bar{A}_1}) + cut(A_2, \bar{A}_2)\right)$

A_2

A_1

$= \frac{1}{2} \cdot 2 \cdot cut(A_1, A_2)$

$= cut(A_1, A_2)$

$\bar{A}_1$

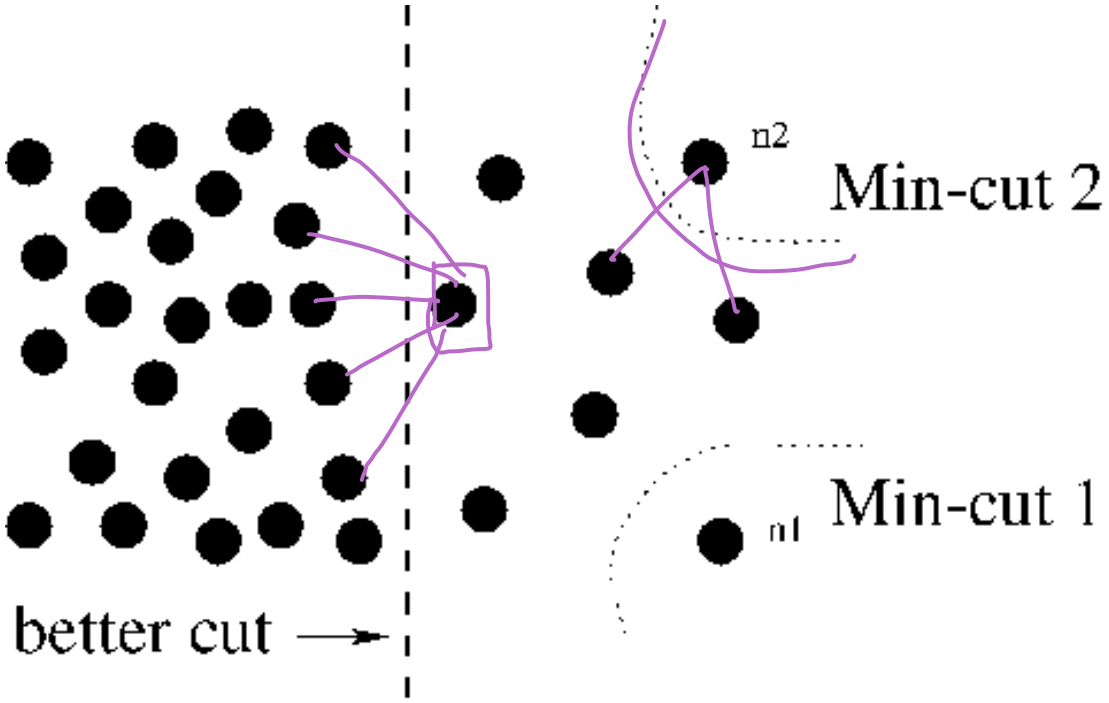
Graph Cut Formulations

Case $k > 2$:

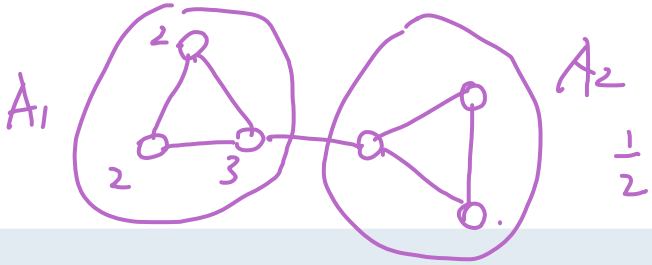
- Given partition $A_1, \ldots, A_k$, define a cut as the total edges weights between groups:

$$cut(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} cut(A_i, \bar{A}_i)$$

Minimizing cut directly tends to favor small isolated clusters.



Balanced Graph Cut



$$\frac{1}{2}\left(\frac{cut(A_1,\bar{A}_1)}{|A_1|}+\frac{cut(A_2,\bar{A}_2)}{|A_2|}\right)$$

$$=\frac{1}{2}\left(\frac{1}{3}+\frac{1}{3}\right).$$

$$=\frac{1}{12}$$

RatioCut and NCut

Find a k-way partition of graph G ($A_i \cup \cdots \cup A_k = V, A_i \cap A_j = \varnothing$) that minimizes:

$$RatioCut(A_1,\ldots,A_k) = \frac{1}{2}\sum_{i=1}^{k}\frac{cut(A_i,\bar{A}_i)}{|A_i|} \quad \text{[Hagen \& Kahng,1992]}$$

Normalized.

$$NCut(A_1,\ldots,A_k) = \frac{1}{2}\sum_{i=1}^{k}\frac{cut(A_i,\bar{A}_i)}{vol(A_i)},$$

$$vol(A_i) = \sum_{i\in A, j\in V} w_{ij} \quad \text{[Shi \& Malik ,2000]}$$

$$=\sum_{x^i\in A} d(x^i)$$

$$\frac{1}{2}\left(\frac{1}{7}+\frac{1}{7}\right)$$

$$=\frac{2}{14}=\frac{1}{7}.$$

Balanced Graph Cut

RatioCut and NCut

Find a k-way partition of graph G ($A_i \cup \cdots \cup A_k = V, A_i \cap A_j = \varnothing$) that minimizes:

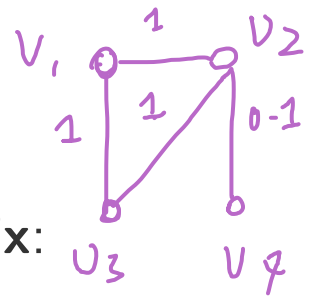
$$RatioCut(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{cut(A_i, \bar{A}_i)}{|A_i|} \quad \text{[Hagen \& Kahng,1992]}$$

$$NCut(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{cut(A_i, \bar{A}_i)}{vol(A_i)},$$

$$vol(A_i) = \sum_{i \in A, j \in V} w_{ij} \quad \text{[Shi \& Malik ,2000]}$$

Both RatioCut and NormalizeCut can be **approximated** *by spectral method.*

Graph Laplacian



Unnormalized graph laplacian matrix:

$$L = D - W = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2.1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0.1 \\ 1 & 1 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2.1 & -1 & -0.1 \\ -1 & -1 & 2 & 0 \\ 0 & -0.1 & 0 & 0.1 \end{bmatrix}$$

Properties of L

1. For every $f \in \mathbb{R}^n$, $f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(f_i^2 + f_j^2 - 2f_if_j) = \frac{1}{2}\Big(\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}}_{d_i} f_i^2 + \underbrace{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}}_{\sum_{j=1}^{n} f_j^2 \sum_{i=1} w_{ij}} f_j^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j\Big)$$

$$= \frac{1}{2}\Big(\sum_{i=1}^{n} d_i f_i^2 + \sum_{j=1}^{n} d_j f_j^2\Big) - \frac{1}{2}\cdot 2 \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j$$

$$= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j$$

$$= f^T D f - f^T W f = f^T (D - W) f = f^T L f$$

Graph Laplacian

Unnormalized graph laplacian matrix:

$$L = D - W$$

Properties of L

1. For every $f \in \mathbb{R}^n$, $f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$

Graph Laplacian

Unnormalized graph laplacian matrix:

$$L = D - W$$

Properties of L

1. For every $f \in \mathbb{R}^n$, $f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$ $\geq 0.$ $\geq 0.$ $\geq 0.$
2. L is symmetric and positive semi-definite

A Review on Eigenvalue Problem

The Eigenvalue Problem

Nonzero vector $u \in \mathbb{R}^n$ is an **eigenvector** of matrix $A \in \mathbb{R}^{n \times n}$ if

$$Au = \lambda u$$

for some $\lambda \in \mathbb{R}$. We call λ the **eigenvalue** corresponding to u.

- A has at most n distinct eigenvalues

$\in \mathbb{R}^{n \times n}$

$$A = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \begin{bmatrix} - & u_1 & - \\ - & u_2 & - \\ & \vdots & \\ - & u_n & - \end{bmatrix}$$

symmetric A

Eigenvalue Decomposition

Let $U = [u_1, \ldots, u_n]$ be the matrix of n linearly independent eigenvectors of A and $\Lambda = diag([\lambda_1, \ldots, \lambda_n])$, then

$$A = U \Lambda U^{-1}$$

- If A is symmetric, A can be decomposed as $A = U \Lambda U^T$ where U is an orthogonal matrix ($U^T U = I$).

Rayleigh-Ritz Theorem

Theorem 1

Given symmetric matrix $A \in \mathbb{R}^{n\times n}$, the solution to the minimization problem is the smallest eigen vector of A

$$\min_{x\in\mathbb{R}^n} \; x^T A x \qquad (1)$$
$$s.t. \quad \|x\|^2 = 1$$

Proof.

$L(x) = x^T A x + \beta(x^T x - 1)$

$\frac{\partial L(x)}{\partial x} = 2Ax + 2\beta x = 0$

$Ax = -\beta x$

eigenvalue eigenvector

Since we want to minimize $x^T A x = (A^T x)^T x = (Ax)^T x = (-\beta x)^T x$.

(by symmetry of A)

Since $\|x\|^2 = x^T x = 1$, $x^T A x = -\beta$ is the smallest eigenvalue.

Rayleigh-Ritz Theorem

Theorem 2

Given symmetric matrix $A \in \mathbb{R}^{n\times n}$, the solution to the minimization problem is the smallest eigen vector of A

$$\min_{x\in\mathbb{R}^n} \quad x^T A x \qquad (2)$$
$$s.t. \quad \|x\|^2 = 1$$

- An equivalent form of (2) is minimizing the **Rayleigh quotient** $\frac{x^T A x}{x^T x}$

$$\min_{x\neq 0\in\mathbb{R}^n} \frac{x^T A x}{x^T x}$$

① $\frac{x^T A x}{x^T x}$ is scale invariant.

Let $x' = cx$. $\frac{x'^T A x'}{x'^T x'} = \frac{x^T A x}{x^T x}$

② Let $x' = cx$ such that $\|x'\|^2 = 1$.

Then $\min_{x'\neq 0} \frac{x'^T A x'}{x'^T x'} = \min_{x'\neq 0} \frac{x'^T A x'}{1} = \min_{x'\neq 0} x'^T A x'$ s.t. $\|x'\| = 1$.

Rayleigh-Ritz Theorem

Theorem 2

Given symmetric matrix $A \in \mathbb{R}^{n \times n}$, the solution to the minimization problem is the smallest eigen vector of A

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & x^T A x \\ s.t. \quad & \underline{\|x\|^2 = 1} \end{aligned} \tag{2}$$

- An equivalent form of (2) is minimizing the **Rayleigh quotient** $\frac{x^T A x}{x^T x}$

$$\min_{x \neq 0 \in \mathbb{R}^n} \frac{x^T A x}{x^T x}$$

- Rayleigh quotient $\frac{x^T A x}{x^T x}$ is scale invariant.

Rayleigh-Ritz Theorem

$$\min_{x} \; x^T A x \quad \text{s.t.} \quad \|x\|^2 = 1. \quad \Longrightarrow \quad \min_{x_1,\dots,x_k} \sum_{i=1}^{k} x_i^T A x_i \quad \text{s.t.} \quad x_i^T x_j = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{if } i \neq j \end{cases}$$

Generalization to multiple vectors:

Theorem 3

Given symmetric matrix $A \in \mathbb{R}^{n\times n}$, $x = [x_1, \dots, x_k], x_i \in \mathbb{R}^n$ $(k \le n)$, the solution to the minimization problem are k smallest eigenvectors of A:

$$\begin{aligned} \min_{X \in \mathbb{R}^{n\times k}} \quad & tr(X^T A X) \\ s.t. \quad & X^T X = \mathbf{I}_k \end{aligned} \tag{3}$$

Graph Laplacian

Unnormalized graph laplacian matrix:

$$L = D - W$$

Properties of L

1. For every $f \in \mathbb{R}^n$, $f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$
2. L is symmetric and positive semi-definite
3. The smallest eigenvalue of L is 0 with eigenvector $\mathbf{1}$

1) $L \cdot 1 = (D - W)1 = D1 - W1 = D - D = 0 = 0 \cdot 1 \Rightarrow$ 0 is an eigenvalue corresponding to eigenvector 1

2) L is PSD. eigenvalues are non-negative

Therefore, 0 is the smallest eigenvalue of L.

Graph Laplacian

Unnormalized graph laplacian matrix:

$$L = D - W$$

Properties of L

1. For every $f \in \mathbb{R}^n$, $f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$
2. L is symmetric and positive semi-definite
3. The smallest eigenvalue of L is 0 with eigenvector $\mathbf{1}$

Graph Laplacian

Unnormalized graph laplacian matrix:

$$L = D - W$$

Properties of L

1. For every $f \in \mathbb{R}^n$, $f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$
2. L is symmetric and positive semi-definite
3. The smallest eigenvalue of L is 0 with eigenvector $\mathbf{1}$
4. L has n real eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

Graph Laplacian

Proposition 1

Let G be an undirected graph with non-negative weights W.

- *The multiplicity k of eigenvalue 0 of L is the number of connected components $A_1, \ldots, A_k$ in G.*
- *The eigenspace of eigenvalue 0 is spanned by vectors $\mathbf{1}_{A_1}, \ldots, \mathbf{1}_{A_k}$*

(Normalized) Graph Laplacian

Normalized graph laplacian (Chung 1997) [1]:

$$L_{rw} = D^{-1}L = I - D^{-1}W$$

Properties of L_{rw}

- λ is an eigenvalue of L_{rw} with eigenvector v if and only if λ, v solve the generalized eigenproblem $Lv = \lambda Dv$
- 0 is an eigenvalue of L with eigenvector $\mathbf{1}$
- L_{rw} is positive semi-definite and has n non-negative eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

[1] "rw" comes from its interpertation as "random walk". Another definition of normalized graph Laplacian is $D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$

(Normalized) Graph Laplacian

Normalized graph laplacian (Chung 1997) [1]:

$$L_{rw} = D^{-1}L = I - D^{-1}W$$

Properties of L_{rw}

- λ is an eigenvalue of L_{rw} with eigenvector v if and only if λ, v solve the generalized eigenproblem $Lv = \lambda Dv$
- 0 is an eigenvalue of L with eigenvector $\mathbf{1}$
- L_{rw} is positive semi-definite and has n non-negative eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

Proposition 2

Let G be an undirected graph with non-negative weights W, the multiplicity k of eigenvalue 0 of L_{rw} is the number of connected components $A_1, \ldots, A_k$ in G.
The eigenspace of eigenvalue 0 is spanned by vectors $\mathbf{1}_{A_1}, \ldots, \mathbf{1}_{A_k}$

[1] "rw" comes from its interpertation as "random walk". Another definition of normalized graph Laplacian is $D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$

Solving graph cut

Define $f \in \{0,1\}^n$ to be the indicator function for partition $A \subset V$:

$$f_i := \{\mathbf{1}_A\}_i = \begin{cases} 1 & v_i \in A \\ 0 & v_i \in \bar{A} \end{cases}$$

We have that $||f||^2 = |A|$.
$Cut(A, \bar{A})$ can be written as a function of f and graph Laplacian L:

$$\begin{aligned} f^T L f &= \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \\ &= \frac{1}{2}\left(\sum_{v_i \in A, v_j \in \bar{A}} w_{ij} + \sum_{v_i \in \bar{A}, v_j \in A} w_{ij} \right) = \sum_{v_i \in A, v_j \in \bar{A}} w_{ij} = cut(A, \bar{A}) \end{aligned}$$

Let $f_{(1)}, \ldots, f_{(k)}$ be k indicator functions $\mathbf{1}_{A_i}, \ldots, \mathbf{1}_{A_k}$. They are mutually orthogonal (i.e. $f_{(i)}^T f_{(j)} = 0$ for all $i \neq j$).

Solving graph cut

Recall the definition of RatioCut:

$$\min_{A_1,\ldots,A_k} \sum_i^k \frac{cut(A_i, \bar{A}_i)}{|A_i|} \tag{4}$$

$$\Longrightarrow \min_{A_1,\ldots,A_k} \sum_i^k \frac{f_{(i)}^T L f_{(i)}}{f_{(i)}^T f_{(i)}} \tag{5}$$

Relax the $f_{(i)}$'s to be real vectors:

$$\min_{f_{(1)},\ldots,f_{(k)} \in \mathbb{R}^n} \sum_i^k \frac{f_{(i)}^T L f_{(i)}}{f_{(i)}^T f_{(i)}} \tag{6}$$

$$s.t.\ f_{(i)}^T f_{(j)} = 0, \text{ for all } i \neq j$$

Solving graph cut

Since rescaling $f_{(i)}$ by constants does not change the objective, (3) is equivalent to

$$\min_{f_{(1)},\ldots,f_{(k)}\in\mathbb{R}^n} \sum_i^k f_{(i)}^T L f_{(i)} \tag{7}$$
$$s.t.\ f_{(i)}^T f_{(j)} = 0,\ \text{for all } i \neq j$$
$$f_{(i)}^T f_{(i)} = 1,\ \text{for all} i = 1,\ldots,k$$

Let $F = \left[f_{(1)} \ldots f_{(k)}\right]$, (5) can be written in matrix notation:

$$\min_{F\in\mathbb{R}^n} \operatorname{tr}(F^T L F)$$
$$s.t.\ F^T F = I$$

- By Theorem 3, optimal solution F^* is the first k eigenvectors of L.
- To get discrete cluster labels, we can apply k-means clustering on the rows of F^*.

Spectral Clustering Algorithm

Unormalized spectral clustering

Input: data points $x^{(1)}, \ldots, x^{(n)}$ and cluster size k

- Build a graph connecting $x^{(1)}, \ldots, x^{(n)}$ with weight W
- Compute first k eigenvectors $V = [v_1, \ldots, v_k]$ of L
- Define $y_i \in \mathbb{R}^k$ as the ith row of V, cluser $y_1, \ldots, y_n$ into k clusters $C_1, \ldots, C_k$ using k-means

Output: $A_1, \ldots, A_k$ where $A_i = \{j | y_j = C_i\}$

- Unormalized spectral clustering is relaxed solution to the RatioCut problem.

Spectral Clustering Algorithm

Normalized spectral clustering (Ng, Shi and Malik 2000)

Input: data points $x^{(1)}, \ldots, x^{(n)}$ and cluster size k

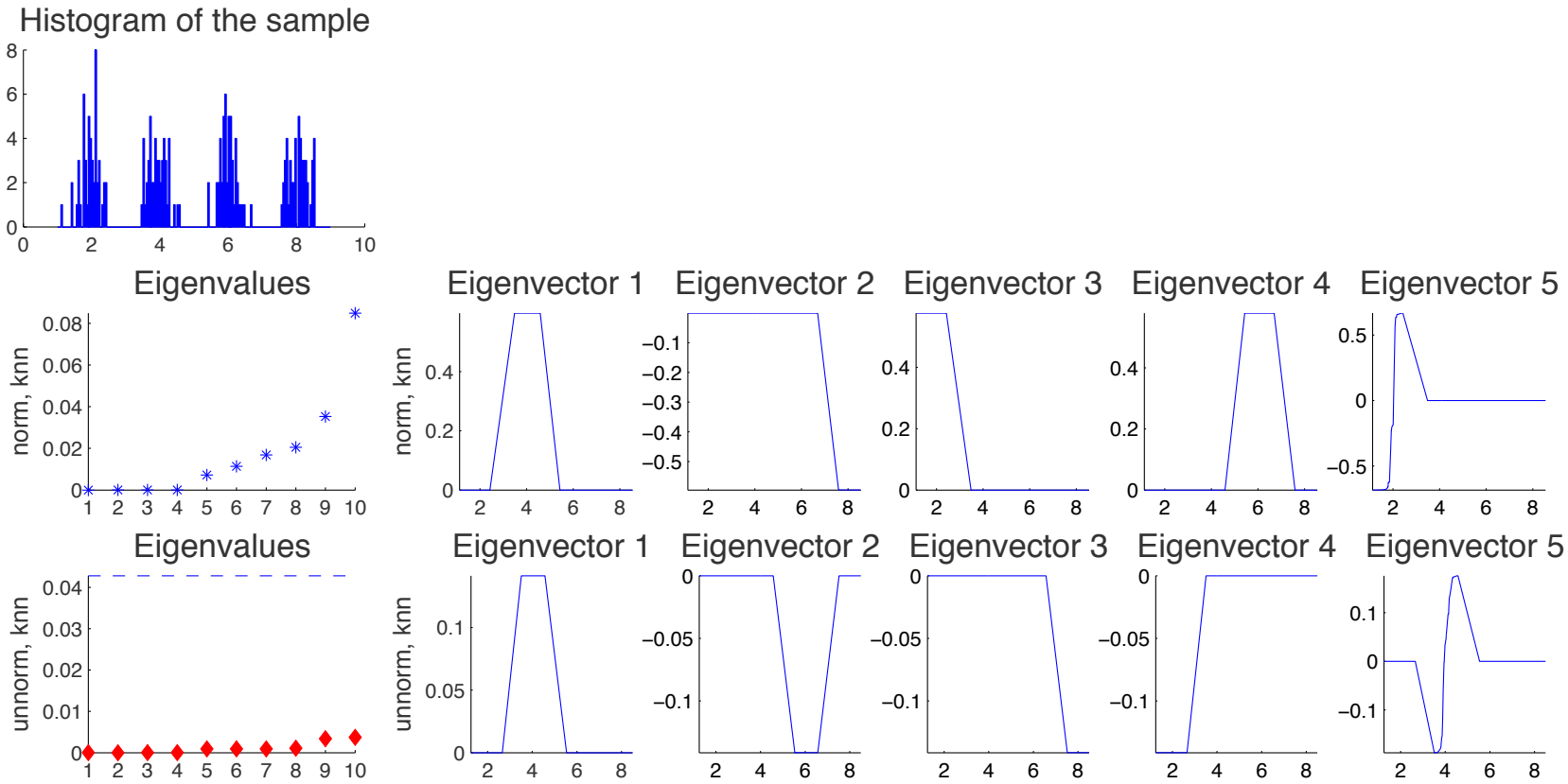
- Build a graph connecting $x^{(1)}, \ldots, x^{(n)}$ with weight W
- Compute first k eigenvectors $V = [v_1, \ldots, v_k]$ of generalized eigen problem $Lv = \lambda Dv$
- Define $y_i \in \mathbb{R}^k$ as the ith row of V, cluser $y_1, \ldots, y_n$ into k clusters $C_1, \ldots, C_k$ using k-means

Output: $A_1, \ldots, A_k$ where $A_i = \{j | y_j = C_i\}$

- Normalized spectral clustering (L_{rw}) is a relaxed solution to the NCut problem.

Toy Example

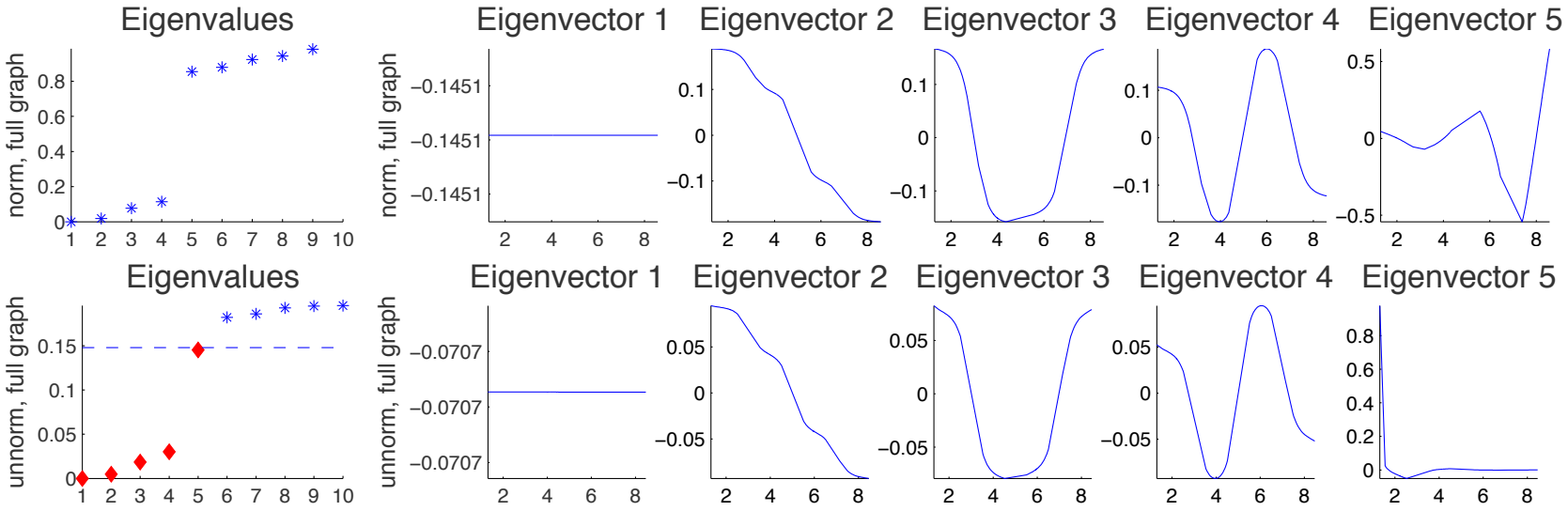
- 200 data points sampled from 4 Gaussian distributions
- KNN similarity graph ($k = 10$)



First 4 eigenvalues are 0 with eigenvectors 1_{A_i} , $i = 1, \ldots, 4$

Toy Example

- Fully connected graph with Gaussian similarity graph ($\sigma = 1$)



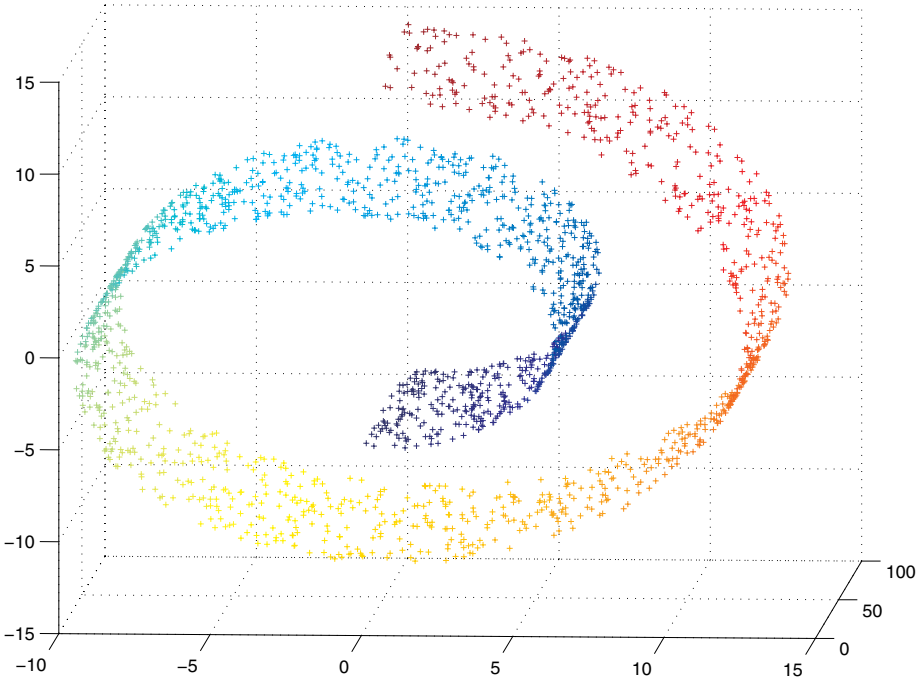
First eigenvector is **1** since the graph has only 1 connected component

Spectral Embedding

Also known as Laplacian Eigenmaps [Belkin et. al., 2003]:

- Learn a k-dimensional embedding $Y = \begin{bmatrix} -y_1- \\ \vdots \\ -y_m- \end{bmatrix} \in \mathbb{R}^{n\times k}$

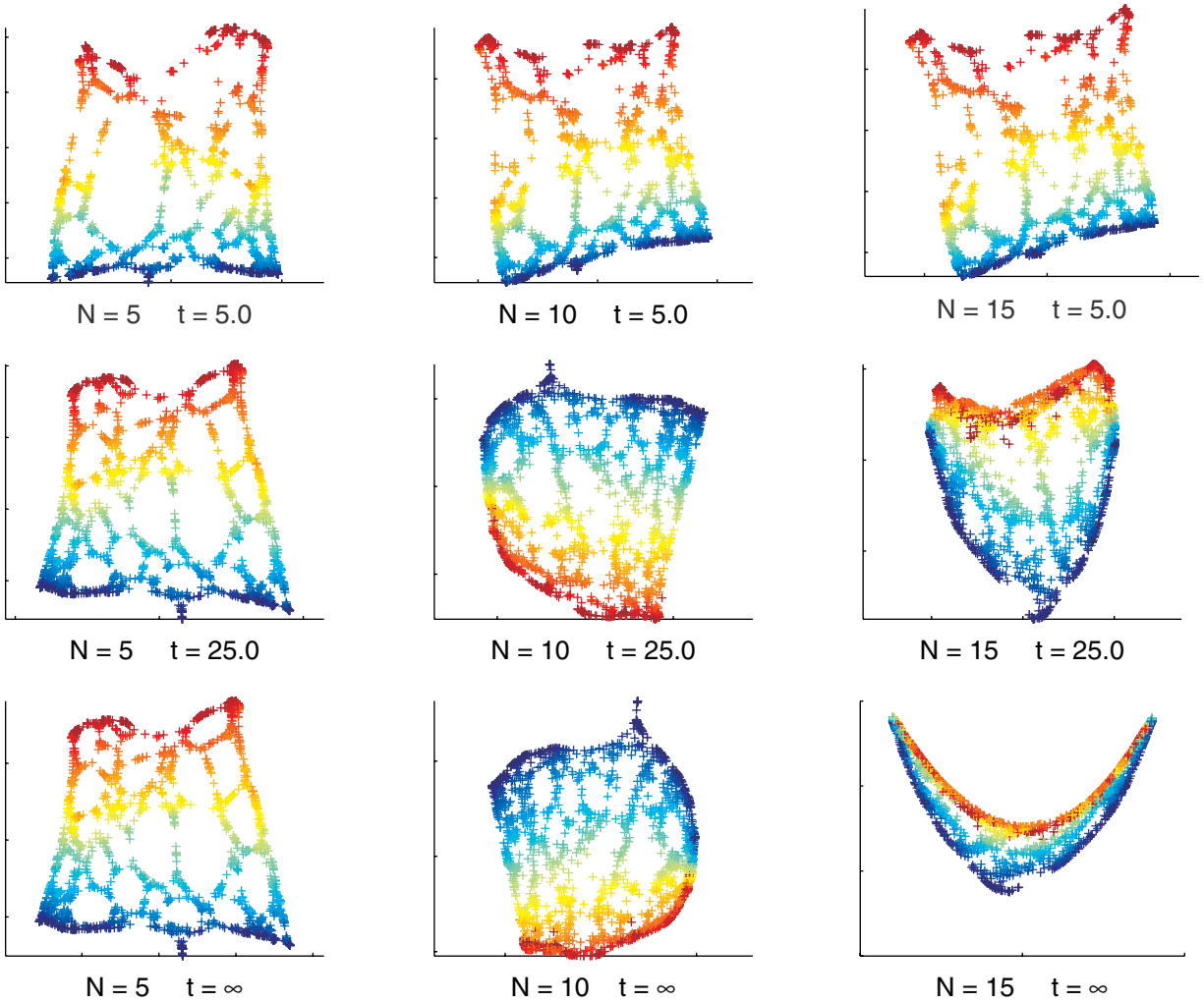
$$\min_{\substack{Y^T DY = I \\ Y^T D\mathbf{1} = 0}} \frac{1}{2} \sum_{ij} w_{ij} \|y_i - y_j\|^2$$



Spectral Embedding

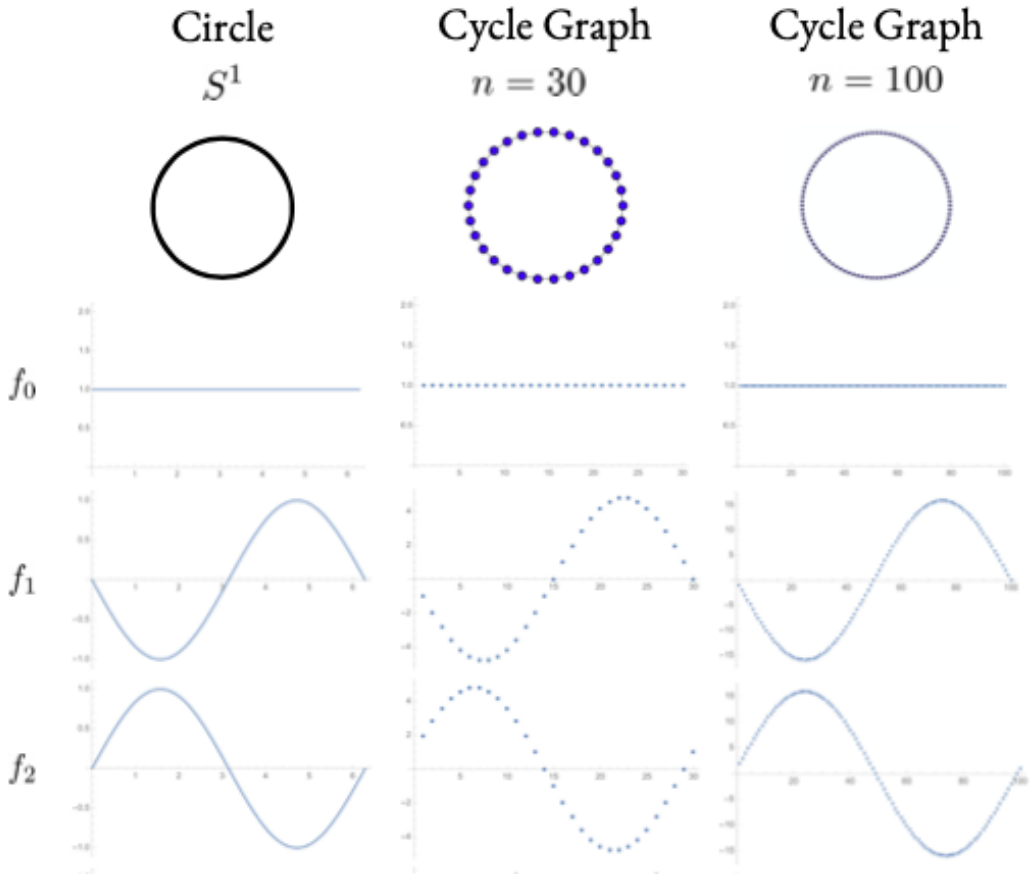
Example: 2D embedding results:

- N: number of neighbors in kNN graph
- t: hyperparameter in the similarity function $W_{i,j} = \exp(\frac{||x_i - x_j||^2}{t})$



Spectral Embedding

Example: Embedding results on a 2D cycle graph

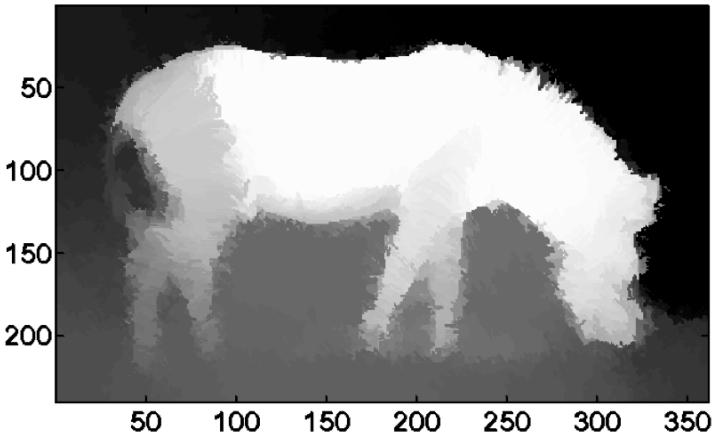


Also studied in graph signal processing and differential geometry

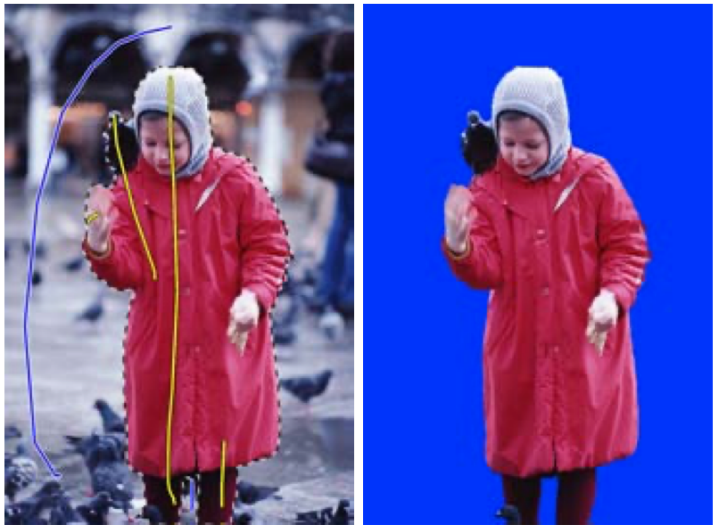
Additional topics of graph Laplacian methods

Graph spectra can be used as topological features for supervised and unsupervised learning

- Laplacian eigenmaps for dimension reduction and visualization
- Unsupervised segmentation
- Graph-based semi-supervised learning (manifold regularization)



Unsupervised segmentation using NCut [Shi & Malik, 2000]



Lazy Snapping (semi-supervised graph cut) [Li et. al. 2004]

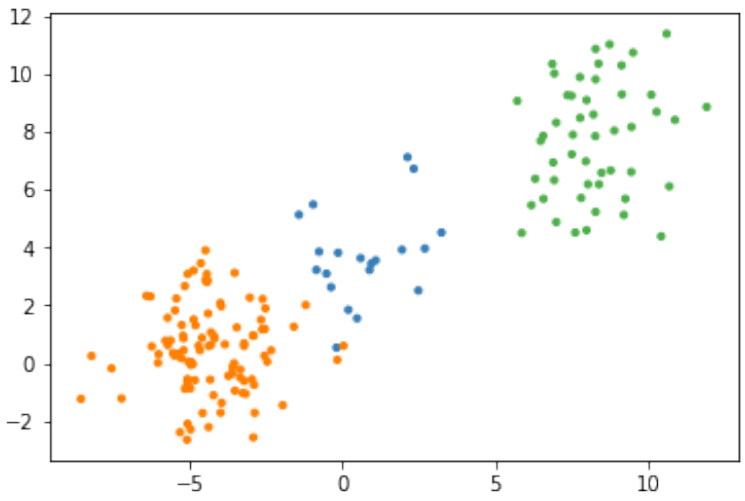
Summary

Representation learning

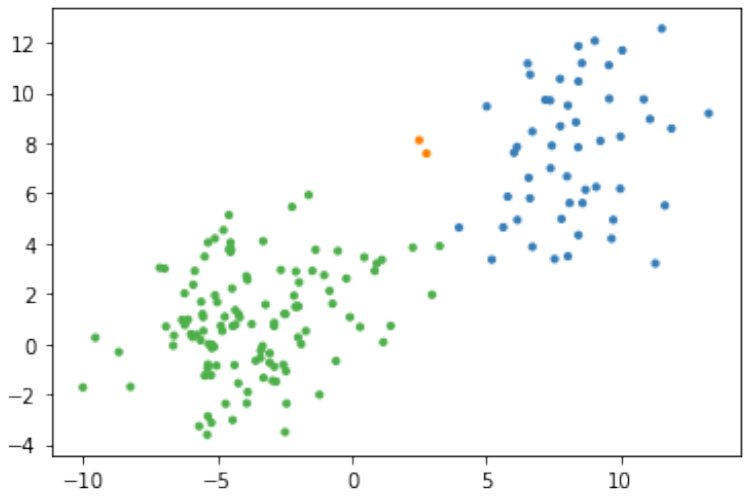
- Transform input features into "simpler" or "interpretable" representations.
- Used in feature extraction, dimension reduction, clustering etc

Unsupervised learning algorithms and their assumptions

- **K-Means**: assumes data are isotropic Gaussian, different clusters have the same prior probability
- **Spectral Methods**: manifold assumption, cluster labels of a node depends on its neighbors



ground truth cluster



spectral clustering (nCut)

Connection to Other Methods

Non-negative Matrix Factorization

- "k-Means Clustering via the Frank-Wolfe Algorithm" [Bauckhage 2016]
- "On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering" [Ding et. al. 2005]

Matrix factorization can be relaxed to a continuous problem, allowing us to use GD /deep neural networks to learn representation and cluster simultaneously.

e.g. Wu et al, "Deep k-Means: Re-Training and Parameter Sharing with Harder Cluster Assignments for Compressing Deep Convolutions", 2018