EECS 445

Introduction to Machine Learning

Spectral Clustering

Prof. Kutty

Google x MHacks

AI Hackathon

- April 12-14
- Google Ann Arbor
- $5k in prizes
- Applications due Fri, Mar 29

mhacks.org

review: k-means clustering

k-means Clustering

$\in \{1, \dots, k\}$

algorithm: more formally

$c_1, \dots, c_n$ $\in \mathbb{R}^d$

Datapoints $\bar{x}^{(1)}, \dots, \bar{x}^{(n)}$ and fixed k

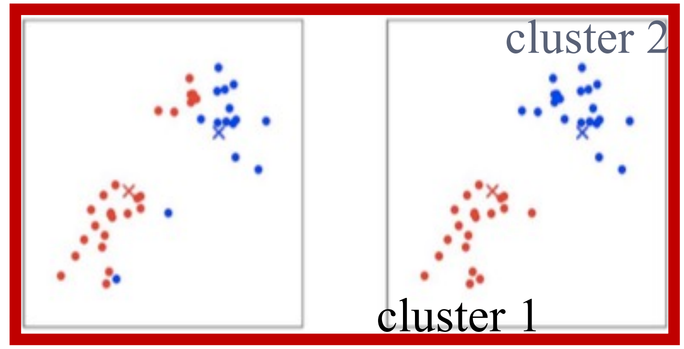
initialize means $\bar{\mu}^{(1)}, \dots, \bar{\mu}^{(k)}$ $\in \mathbb{R}^d$

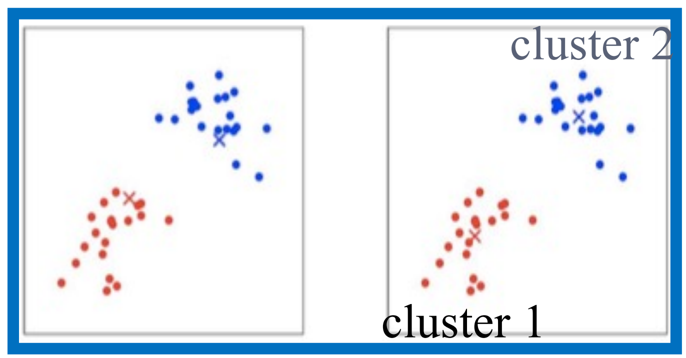
Iteratively

- for each point $\bar{x}^{(i)}$, reassign $\bar{x}^{(i)}$ to

$$c_i = \arg\min_j \left\| \bar{x}^{(i)} - \bar{\mu}^{(j)} \right\|^2$$

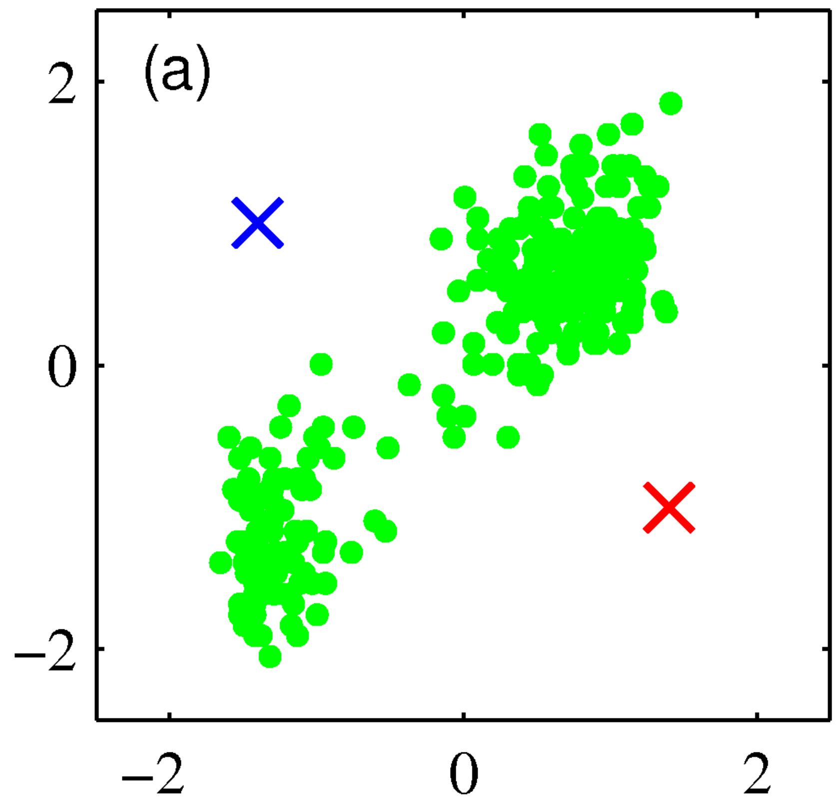
- recompute $\bar{\mu}^{(j)} = \frac{\sum_i [\![c_i = j]\!] \bar{x}^{(i)}}{\sum_i [\![c_i = j]\!]}$



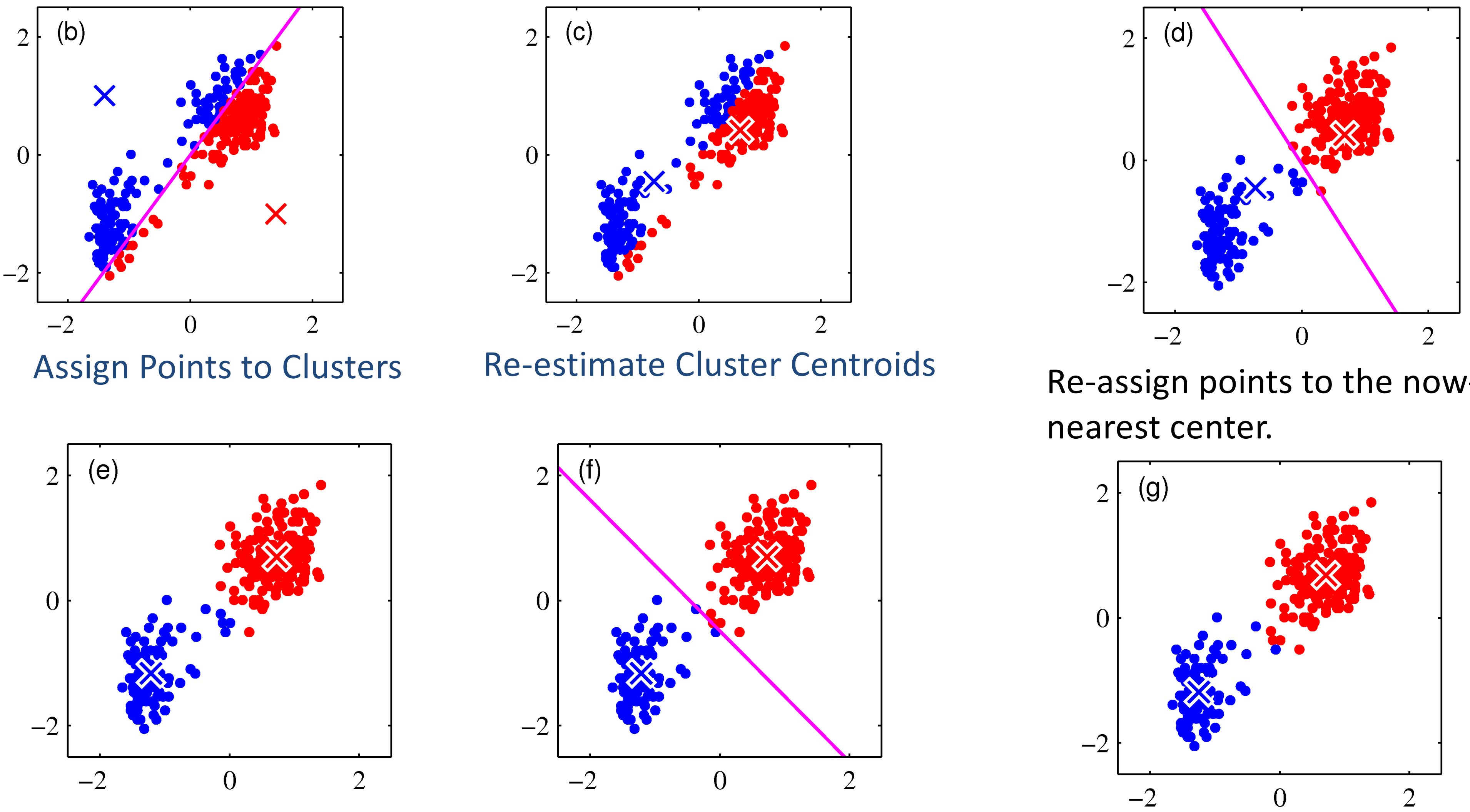


K-Means Clustering (Initialization)

- Select k. Pick random means.
  - Example with k = 2.



K-Means Clustering (Iterations)



Assign Points to Clusters

Re-estimate Cluster Centroids

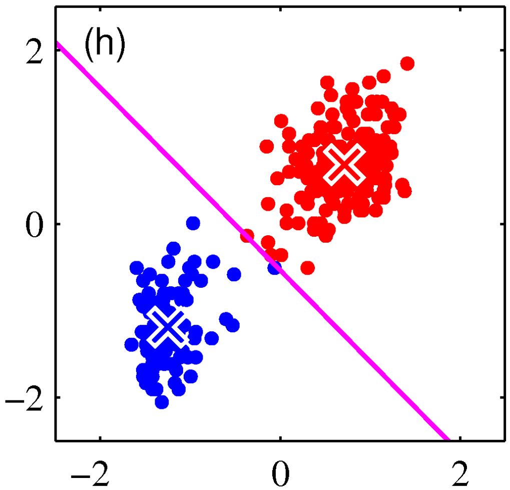
Re-assign points to the now-nearest center.

Compute centers for the new clusters assignments.

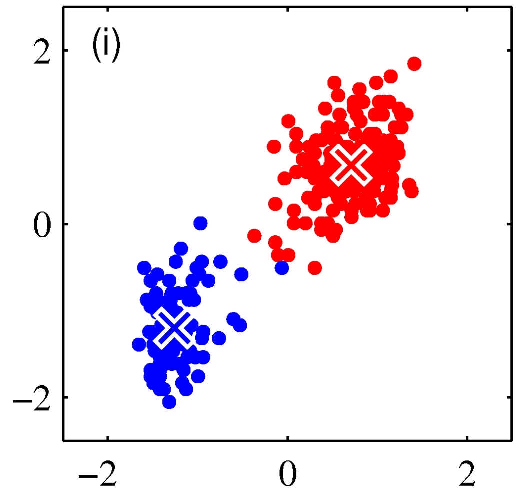
Re-assign Points to Clusters

Re-estimate Cluster Centroids

K-Means Clustering (Convergence)



Re-assign Points to Clusters

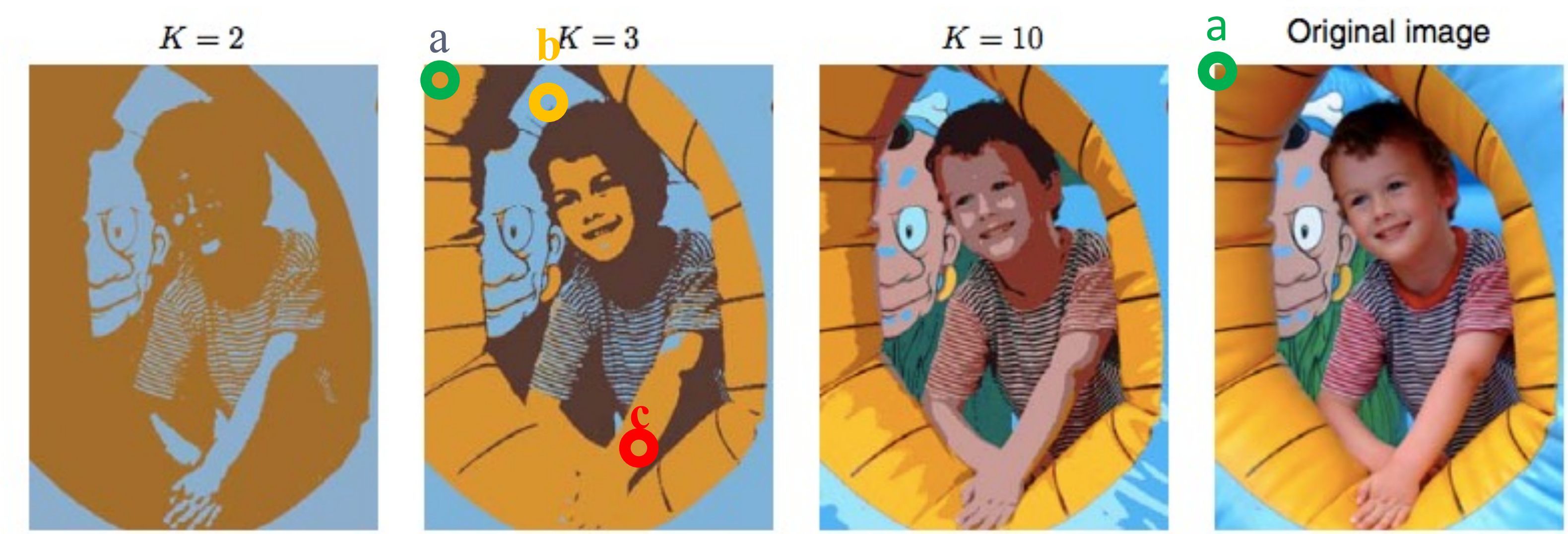


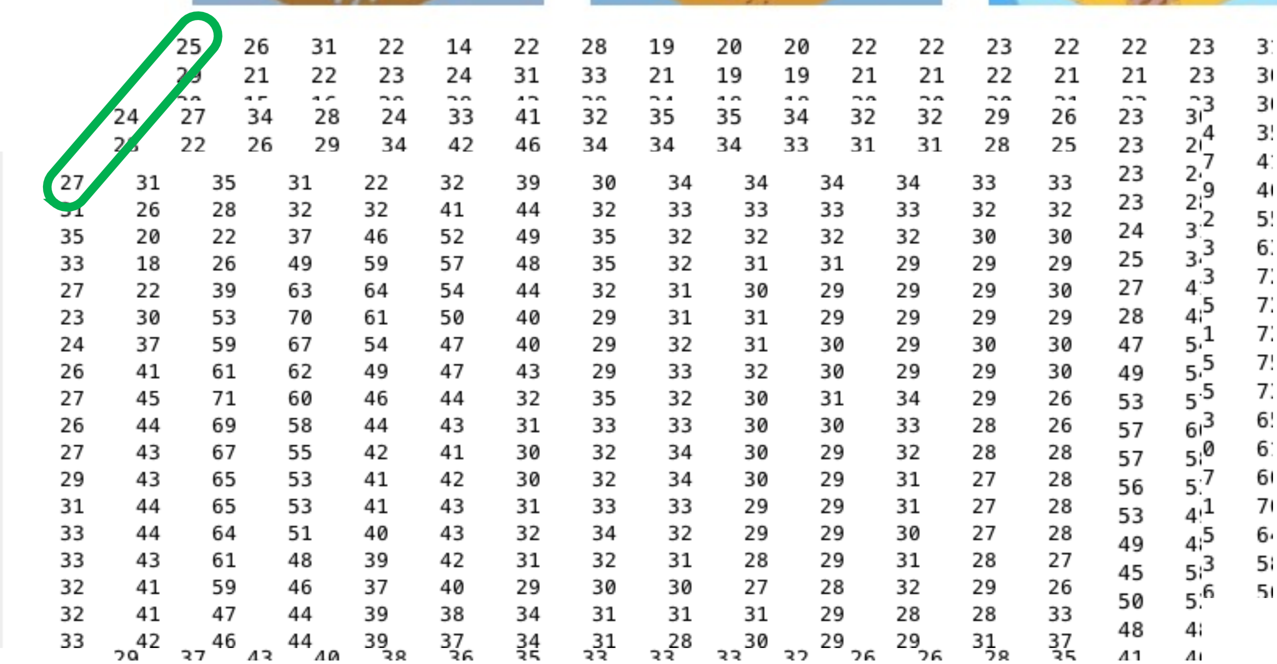
The cluster centers have stopped changing.

image courtesy Bishop 2006

Example: Image segmentation

using k-means clustering





Pixels a, c
are in the *same* cluster
Pixels a, b
are in *different* clusters

Datapoints would the pixels
$\vec{x}^{(i)} \in \mathbb{R}^3$

k-means Clustering

Iteratively

- reassign $\bar{x}^{(i)}$ to $c_i = \arg\min_j \left\|\bar{x}^{(i)} - \bar{\mu}^{(j)}\right\|^2$
- recompute $\bar{\mu}^{(j)} = \frac{\sum_i [\![c_i = j]\!] \bar{x}^{(i)}}{\sum_i [\![c_i = j]\!]}$

fix $\bar{\mu}$, choose $\bar{c}$ to minimize

$$J(\bar{c}, M) = \sum_{i=1}^{n} \left\|\bar{x}^{(i)} - \bar{\mu}^{(c_i)}\right\|^2$$

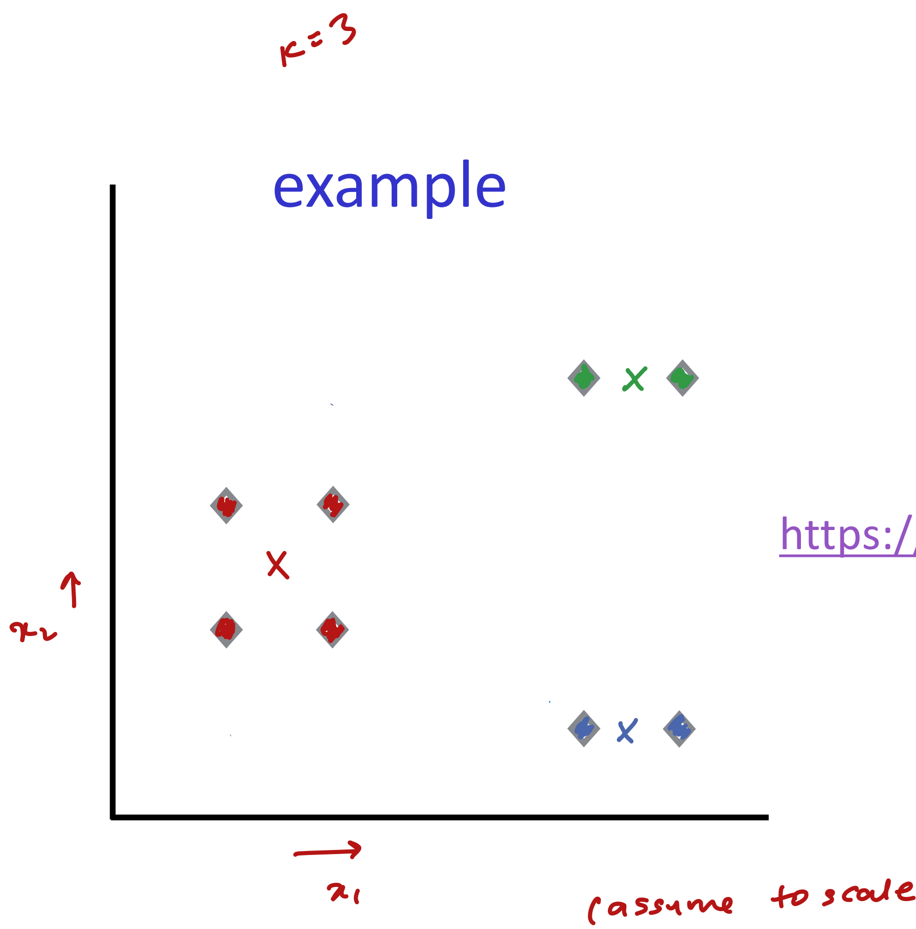
fix $\bar{c}$, choose $\bar{\mu}$ to minimize

$$J(\bar{c}, M) = \sum_{i=1}^{n} \left\|\bar{x}^{(i)} - \bar{\mu}^{(c_i)}\right\|^2$$

k-means is guaranteed to converge but… *not guaranteed to converge to global minimum*

Getting stuck in local minima

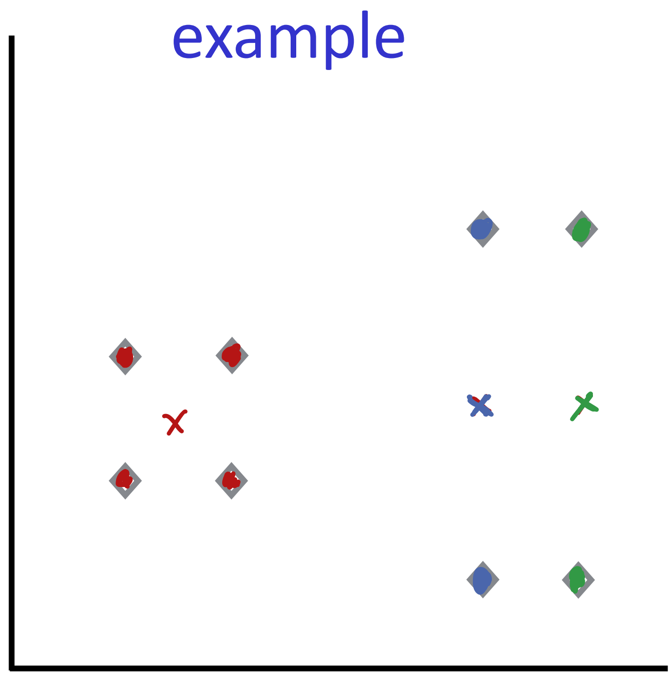
- What we want



https://forms.gle/ffiBvNbPjHF8ghi77

Getting stuck in local minima

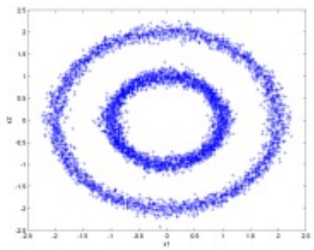
- What we get

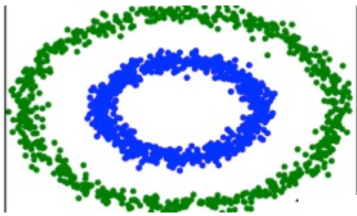


k-means global optimum

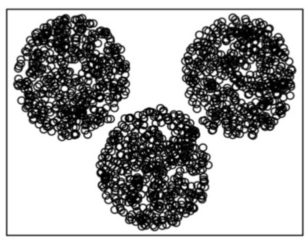
- What we want

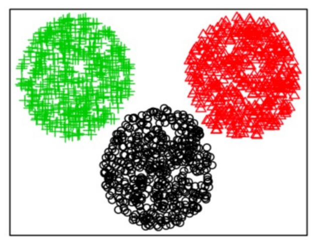
k=2



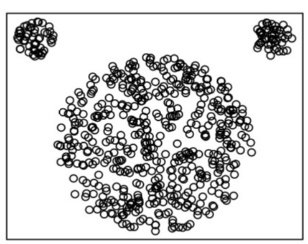


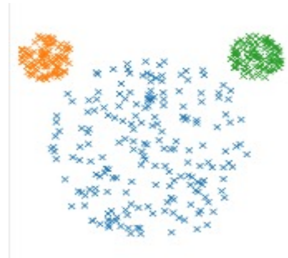
k=3





k = 3

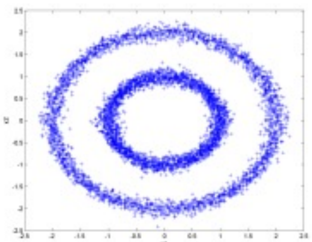


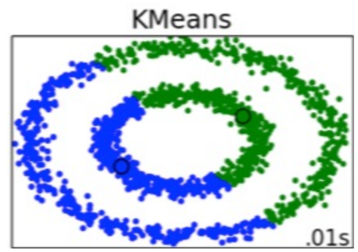


k-means global optimum

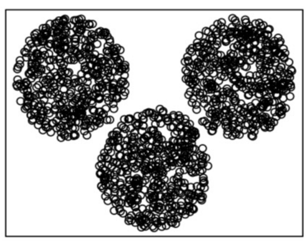
- What we get

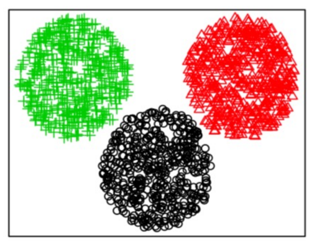
A.





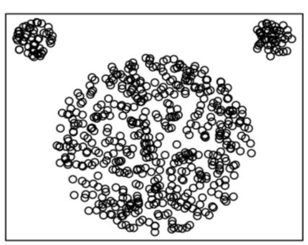
B.

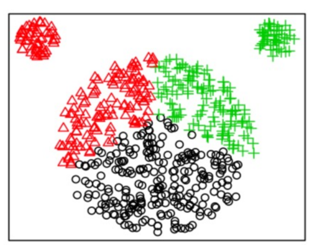




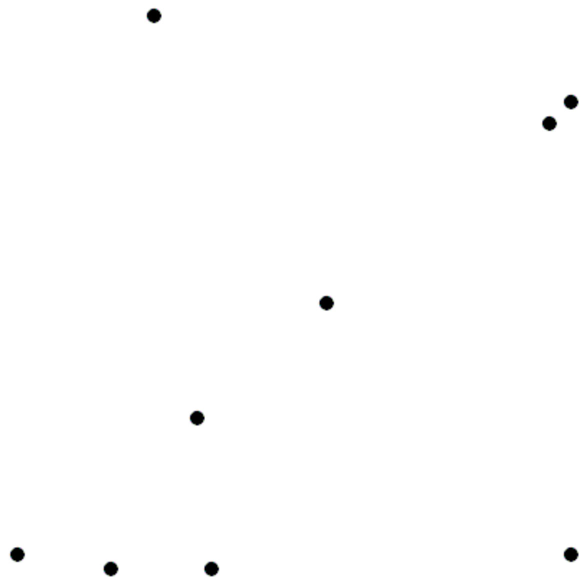
https://forms.gle/ffiBvNbPjHF8ghi77

C.





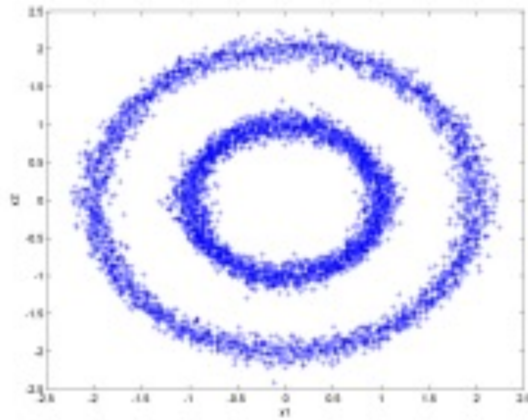
How do means determine cluster assignments?

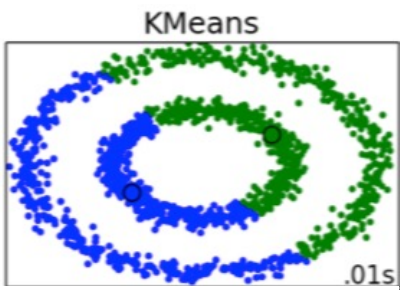


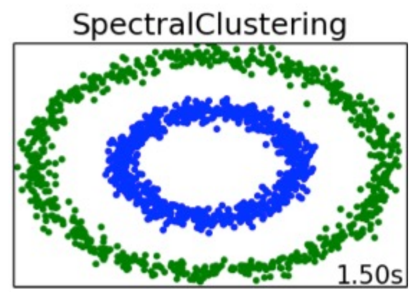
By Jahobr - Own work, CC0, https://commons.wikimedia.org/w/index.php?curid=61356414

Use with caution and understanding!
https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

Clustering Algorithms







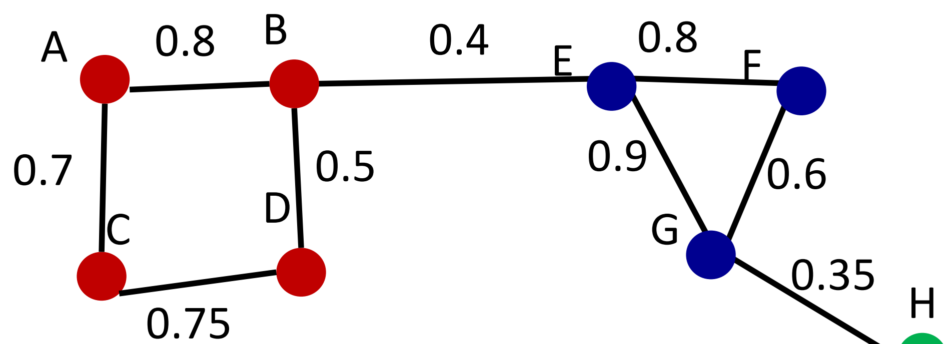
http://scikit-learn.org/stable/modules/clustering.html

spectral clustering

Spectral Clustering

- doesn't assume globular-shaped clusters
- Reformulates data clustering problem as graph partitioning problem
- Broadly
  - first, convert data into a weighted graph
  - next, partition graph so that each component has a weaker across-partition connection and stronger within-partition connection; *ensure similar sized partitions*

Given datapoints
{A, B, C, D, E, F, G, H}

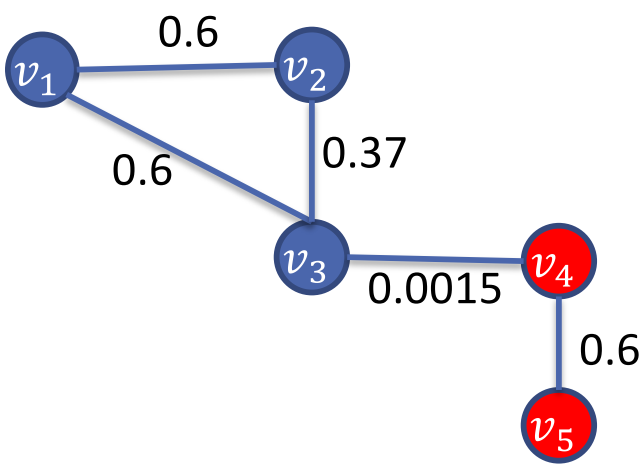


- How?

Definition: Cost of a cut

- Complement of A: $\boxed{\bar{A} = V \setminus A}$ where V is the vertex set
- Cost of a cut between A and $\bar{A}$

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



if $A = \{v_1, v_2, v_3\}$
then $\bar{A} = \{v_4, v_5\}$

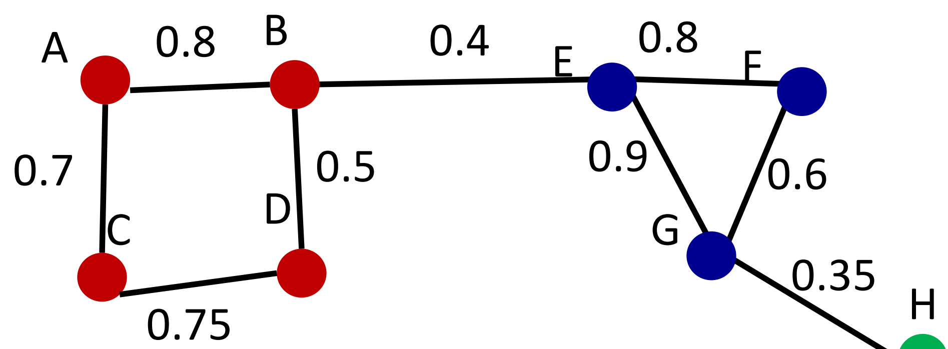
$cut(A, \bar{A})=$ 0.0015

Spectral Clustering: try 1

Goal: Given a graph representing the data, find a minimum cost cut?

Issue: May not give a reasonable solution

$k=2$

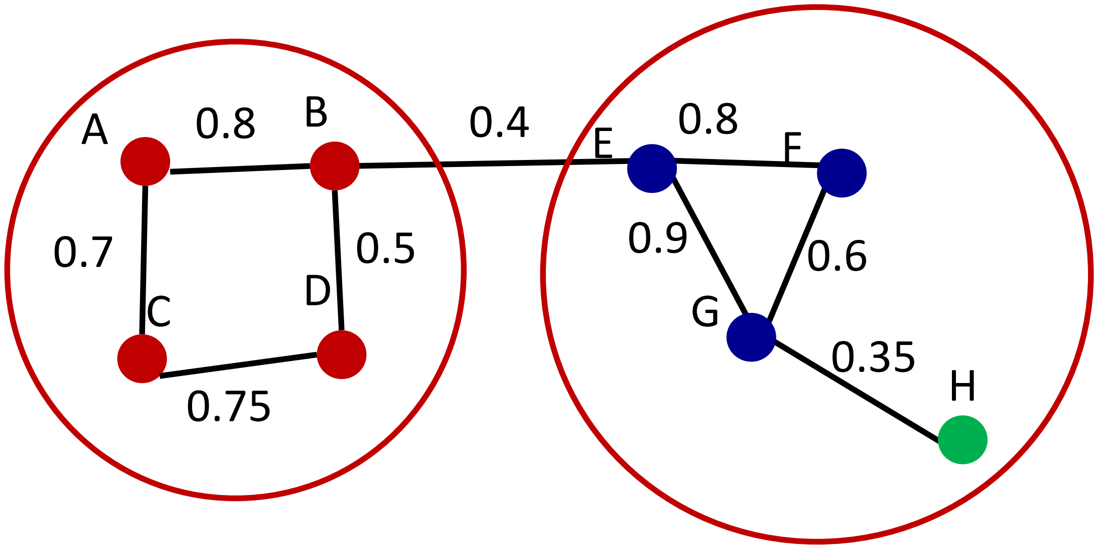


$$\min_{A_1,\ldots,A_k} Cut(A_1, \ldots, A_k)$$

https://forms.gle/ffiBvNbPjHF8ghi77

Spectral Clustering

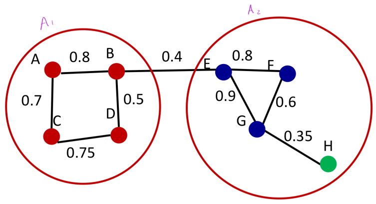
Goal: Given a graph representing the data,
find a minimum cost RatioCut → k clustering



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

called "ratio cut" ensures clusters are reasonably large groups

$$\boxed{\min_{A_1,\ldots,A_k} RatioCut(A_1, \ldots, A_k)}$$



$A_1 = \{A, B, C, D\}$

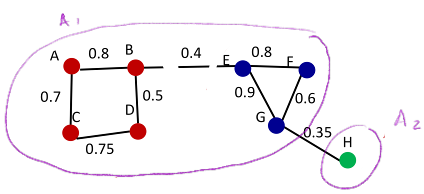
$A_2 = \{E, F, G, H\}$

$|A_1| = |A_2| = 4$

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

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

$$= \frac{1}{2}\left[\frac{0.4}{4} + \frac{0.4}{4}\right] = 0.1$$



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

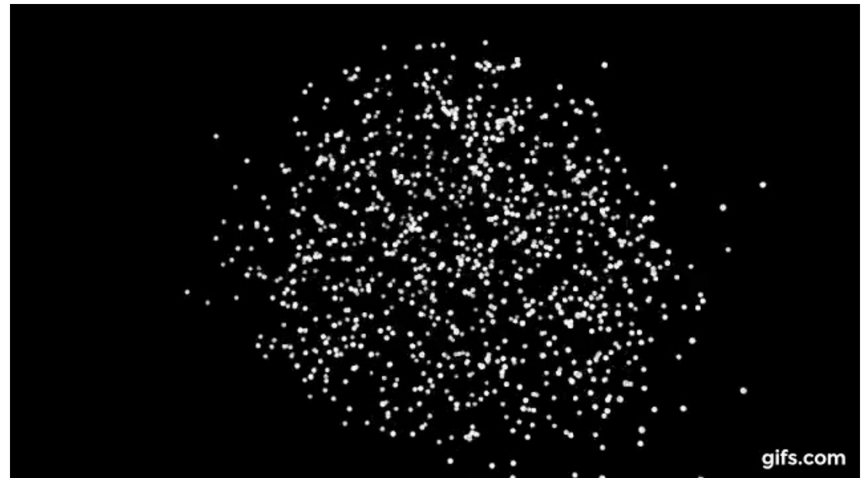
$$RatioCut(A_1, A_2) = \frac{1}{2}\left[\frac{cut(A_1, \bar{A}_1)}{|A_1|} + \frac{cut(A_2, \bar{A}_2)}{|A_2|}\right]$$

$$= \frac{1}{2}\left[\frac{0.35}{7} + \frac{0.35}{1}\right] = 0.2$$

spectral clustering: Big idea

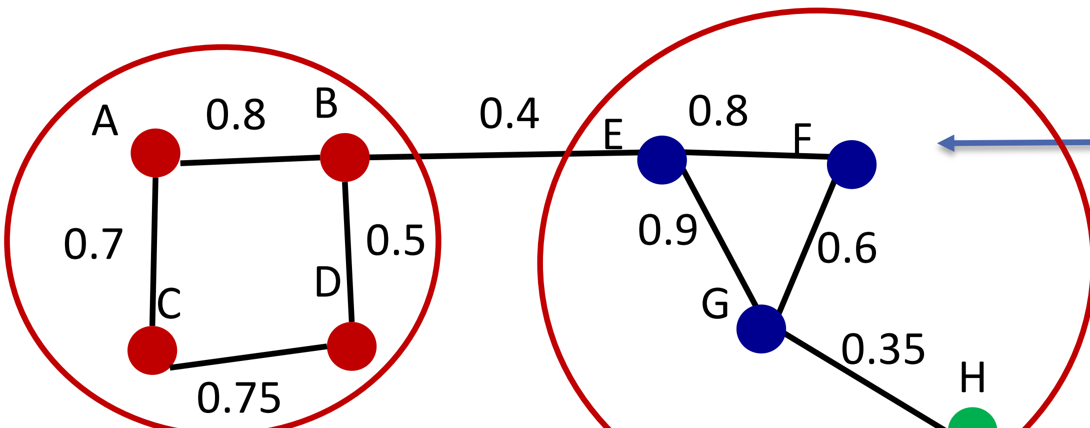
To get k clusters

$$\min_{A_1,\dots,A_k} RatioCut(A_1,\dots,A_k)$$



for each pair of datapoints

compute edge weights based on similarity



e.g., k=2

Spectral Clustering for k partitions

Input: valid similarity metric, number of clusters k

1. Build adjacency matrix W

2. Compute graph Laplacian $L = D - W$

3. Compute (eigenvector, eigenvalue) pairs of L

4. Build matrix with the first k eigenvectors (corresponding to the k smallest eigenvalues) as columns interpret rows as new data points
Low dimensional embedding ($\in \mathbb{R}^k$) of the original dataset ($\in \mathbb{R}^d$)

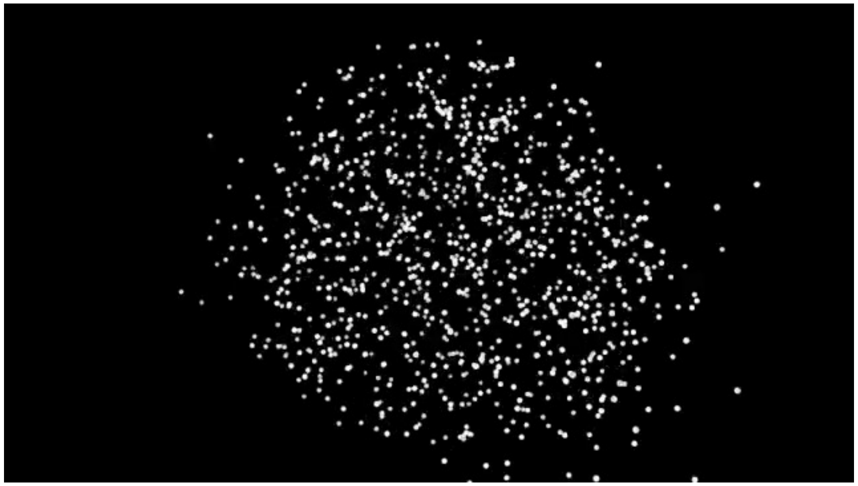
5. Apply k-means to new data representation

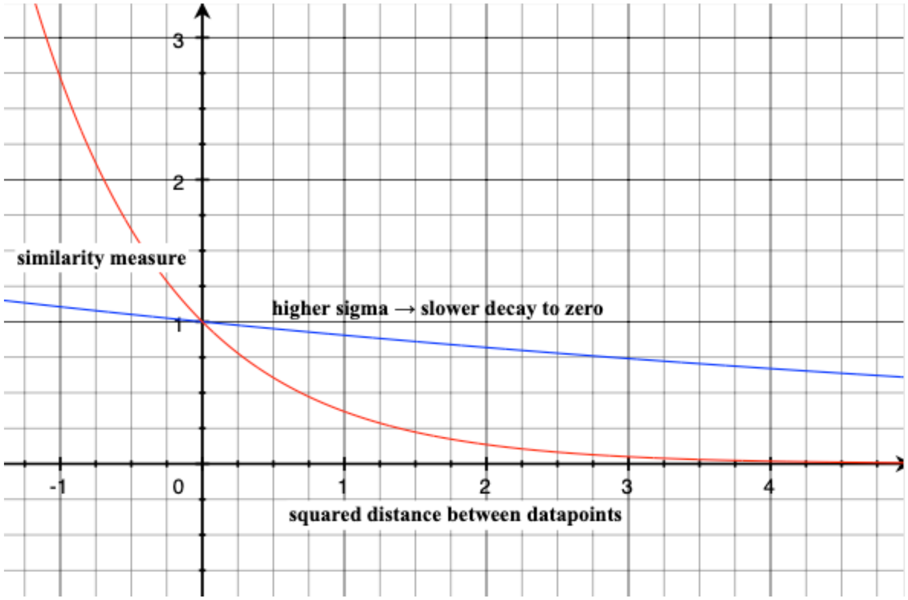
Output: clusters assignments

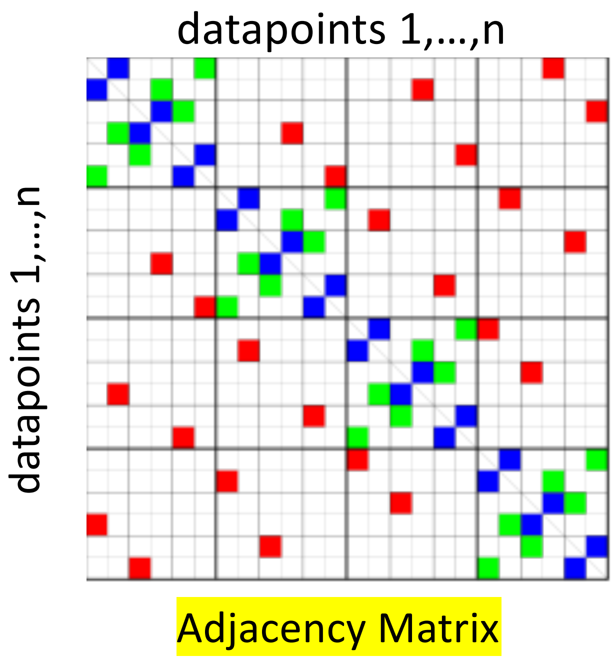
Data ⟶ Weighted Graph

Gaussian kernel similarity metric

$$w_{ij} = exp\left\{-\frac{\left\|\bar{x}^{(i)} - \bar{x}^{(j)}\right\|^2}{2\sigma^2}\right\}$$







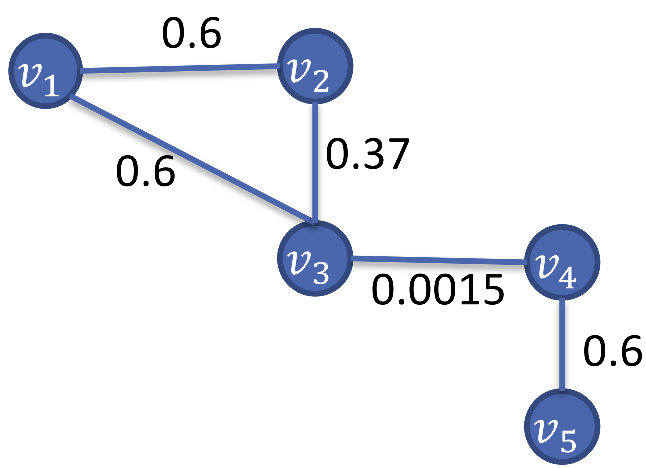
Data → Graph: Adjacency matrix, W

Given $S_n = \{\bar{x}^{(i)}\}_{i=1}^{n}$

compute similarity between each pair of datapoints e.g., Gaussian kernel similarity metric $w_{ij} = exp\left\{-\frac{\left\|\bar{x}^{(i)}-\bar{x}^{(j)}\right\|^2}{2\sigma^2}\right\}$

How might you get a weight of 0? e.g., any value <0.0001 is interpreted as 0

example:



W =

1	**0.6**	**0.6**	**0**	**0**
0.6	**1**	**0.37**	**0**	**0**
0.6	**0.37**	**1**	**0.0015**	**0**
0	**0**	**0.0015**	**1**	**0.6**
0	**0**	**0**	**0.6**	**1**

$W = \{w_{ij}\}$ for $i = 1, \ldots, n; j = 1, \ldots, n$

W is symmetric and each $w_{ij} \geq 0$

Spectral Clustering for k partitions

Input: valid similarity metric, number of clusters k

1. Build adjacency matrix W

2. Compute graph Laplacian $L = D - W$

3. Compute (eigenvector, eigenvalue) pairs of L

4. Build matrix with the first k eigenvectors (corresponding to the k smallest eigenvalues) as columns interpret rows as new data points
Low dimensional embedding ($\in \mathbb{R}^k$) of the original dataset ($\in \mathbb{R}^d$)

5. Apply k-means to new data representation

Output: clusters assignments

Adjacency Matrix W, Degree Matrix D and Graph Laplacian L

- Adjacency Matrix W

  $W = \{w_{ij}\}$ for $i = 1, \ldots, n; j = 1, \ldots, n$

  where $w_{ij} = \mathrm{sim}(\bar{x}^{(i)}, \bar{x}^{(j)})$

- Degree matrix D with

  $d_{ii} = \sum_{j=1}^{n} w_{ij}$ and $d_{ij} = 0$ for $i \neq j$

$$D = \begin{bmatrix} d_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_{nn} \end{bmatrix}$$

1	0.6	0.6	0	0
0.6	1	0.37	0	0
0.6	0.37	1	0.0015	0
0	0	0.0015	1	0.6
0	0	0	0.6	1

$\Sigma \rightarrow d_{33}$

- The graph Laplacian is the matrix $L = D - W$
- We are interested in the **eigenvectors** and **eigenvalues** of L

Recall: Eigenvalues and eigenvectors

- A scalar λ is called an eigenvalue of a matrix A if there is a non-trivial solution v of

$$Av = \lambda v$$

- We say that v is the eigenvector corresponding to the eigenvalue λ

Note: an eigenvector *cannot* be $\bar{0}$, but an eigenvalue *can* be 0.

- Example:

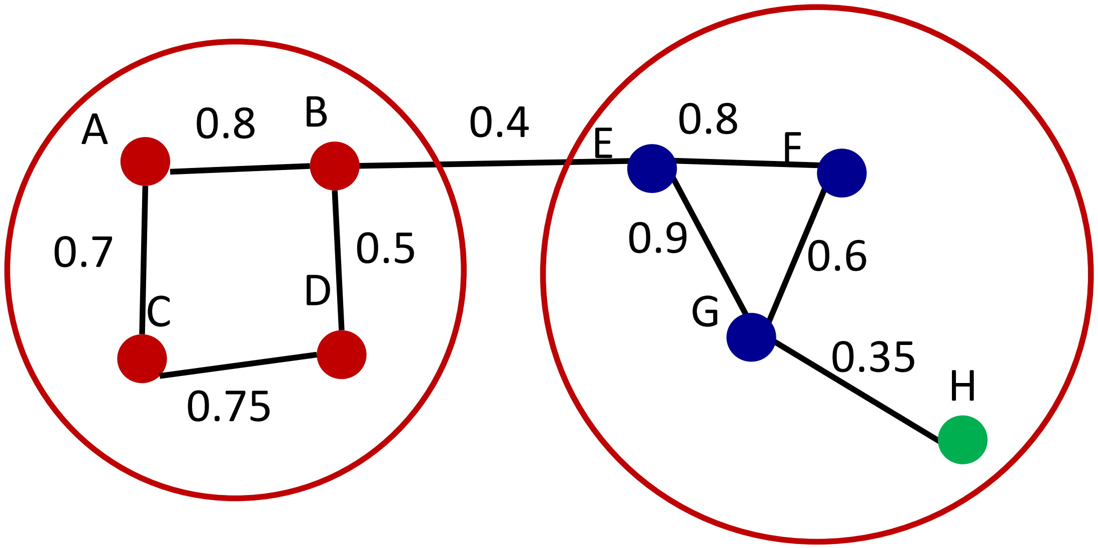
$$A = \begin{bmatrix} 3 & 6 & -8 \\ 0 & 0 & 6 \\ 0 & 0 & 2 \end{bmatrix} \qquad \lambda = 2 \qquad v = \begin{bmatrix} -10 \\ 3 \\ 1 \end{bmatrix}$$

$$Av = \begin{bmatrix} 3 & 6 & -8 \\ 0 & 0 & 6 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} -10 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} -20 \\ 6 \\ 2 \end{bmatrix} = 2 \begin{bmatrix} -10 \\ 3 \\ 1 \end{bmatrix} = \lambda v$$

Spectral Clustering

Goal: Given a graph representing the data,

find a minimum cost RatioCut → k clustering



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

called "ratio cut" ensures clusters are reasonably large groups

Idea: Use the **eigenvectors** and **eigenvalues** of the graph Laplacian matrix $L = D - W$

Spectral Clustering for k partitions

Input: valid similarity metric, number of clusters k

1. Build adjacency matrix W

2. Compute graph Laplacian $L = D - W$

3. Compute (eigenvector, eigenvalue) pairs of L

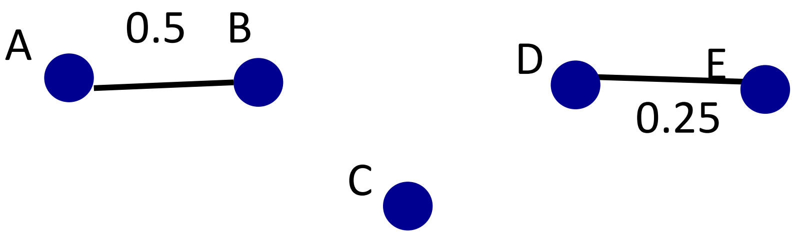
4. Build matrix with the first k eigenvectors (corresponding to the k smallest eigenvalues) as columns interpret rows as new data points

Low dimensional embedding ($\in \mathbb{R}^k$) of the original dataset ($\in \mathbb{R}^d$)

5. Apply k-means to new data representation

Output: clusters assignments

Spectral Clustering Example #1



$$L = \begin{bmatrix} 0.5 & -0.5 & 0 & 0 & 0 \\ -0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & -0.25 \\ 0 & 0 & 0 & -0.25 & 0.25 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 0.5 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0.25 \\ 0 & 0 & 0 & 0.25 & 1 \end{bmatrix}$$

$n \times n$ matrix

Input: similarity metric, number of clusters k

1. Build adjacency matrix W
2. Compute graph Laplacian $L = D - W$

$$D = \begin{bmatrix} 1.5 & & & & \\ & 1.5 & & 0 & \\ & & 1 & & \\ 0 & & & 1.25 & \\ & & & & 1.25 \end{bmatrix}$$

n: #datapoints

Spectral Clustering Example #1

$$L = \begin{bmatrix} 0.5 & -0.5 & 0 & 0 & 0 \\ -0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & -0.25 \\ 0 & 0 & 0 & -0.25 & 0.25 \end{bmatrix}$$

Input: similarity metric, number of clusters k

1. Build adjacency matrix W
2. Compute graph Laplacian $L = D - W$
3. Compute eigenvectors/eigenvalues of L

n (eigenvector, eigenvalue) pairs

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \\ 0 \\ 0 \\ 0 \end{pmatrix},0 \quad \begin{pmatrix} 0 \\ 0 \\ 1.00 \\ 0 \\ 0 \end{pmatrix},0 \quad \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \end{pmatrix},0 \quad \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix},0.5 \quad \begin{pmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \\ 0 \\ 0 \end{pmatrix},1$$

Spectral Clustering Example #1

Input: similarity metric, number of clusters k

1. Build adjacency matrix W
2. Compute graph Laplacian $L = D - W$
3. Compute eigenvectors of L
4. Build matrix with the first k eigenvectors (corresponding to the *k* smallest eigenvalues) as columns interpret rows as new data points
5. Apply k-means to new data representation

Output: clusters assignments

Eigenvectors

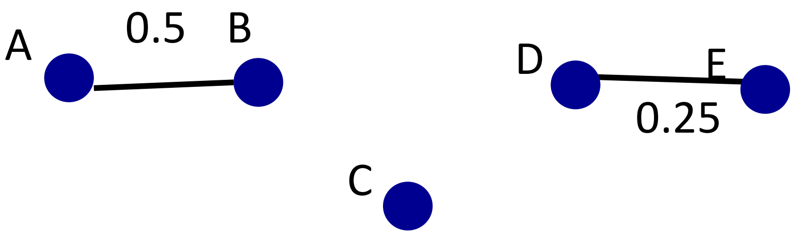
$$\begin{array}{c} A \\ B \\ C \\ D \\ E \end{array} \begin{bmatrix} -\frac{\sqrt{2}}{2} & 0 & 0 & 0 & -\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & 0 & 0 & 0 & \frac{\sqrt{2}}{2} \\ 0 & 1.00 & 0 & 0 & 0 \\ 0 & 0 & -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{bmatrix}$$

Eigenvalues

$$[\boxed{0}, \boxed{0}, \boxed{0}, \boxed{0.5}, \boxed{1.0}]$$

non decreasing order of eigenvalues

Spectral Clustering Example #1



$$L = \begin{bmatrix} 0.5 & -0.5 & 0 & 0 & 0 \\ -0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & -0.25 \\ 0 & 0 & 0 & -0.25 & 0.25 \end{bmatrix}$$

Eigenvectors

$$\begin{array}{c} \text{A} \\ \text{B} \\ \text{C} \\ \text{D} \\ \text{E} \end{array} \begin{bmatrix} -\frac{\sqrt{2}}{2} & 0 & 0 & 0 & -\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & 0 & 0 & 0 & \frac{\sqrt{2}}{2} \\ 0 & 1.00 & 0 & 0 & 0 \\ 0 & 0 & -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{bmatrix}$$

Eigenvalues

$$[0, 0, 0, 0.5, 1.0]$$

$k = 3$

k dimensional embedding

A = $\left[-\frac{\sqrt{2}}{2} \quad 0 \quad 0\right]^T$

B = $\left[-\frac{\sqrt{2}}{2} \quad 0 \quad 0\right]^T$

C = $\left[0 \quad 1.00 \quad 0\right]^T$

D = $\left[0 \quad 0 \quad -\frac{\sqrt{2}}{2}\right]^T$

E = $\left[0 \quad 0 \quad -\frac{\sqrt{2}}{2}\right]^T$

Run k-means clustering on this embedding

Return clusters