Computer Vision:
K-means

Siheng Chen 陈思衡

Project team-up report template is online

everyone has to submit on your own

due on June 10th

Next Monday (June 13th),
the lecture will discuss everything about the project.
Please bring your question

K-means

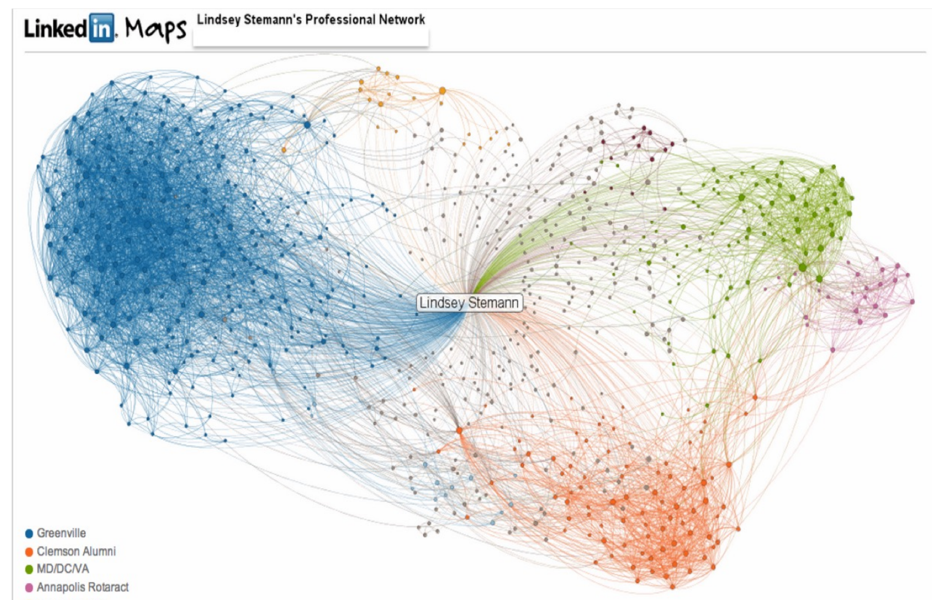
Unsupervised learning: Clustering

Application: Image segmentation & image retrieval

Method: K-means, K-means++, spectral clustering

Clustering

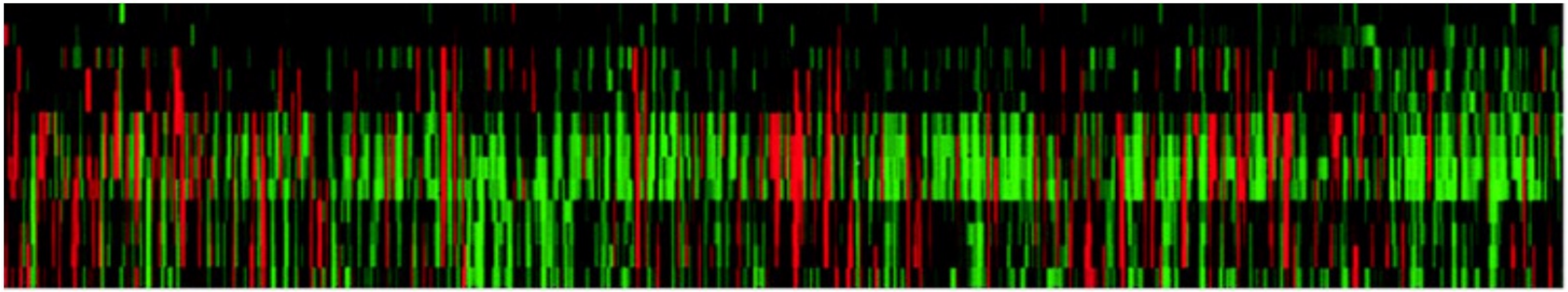
the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters)



Social networks

Clustering

Clustering gene expression data

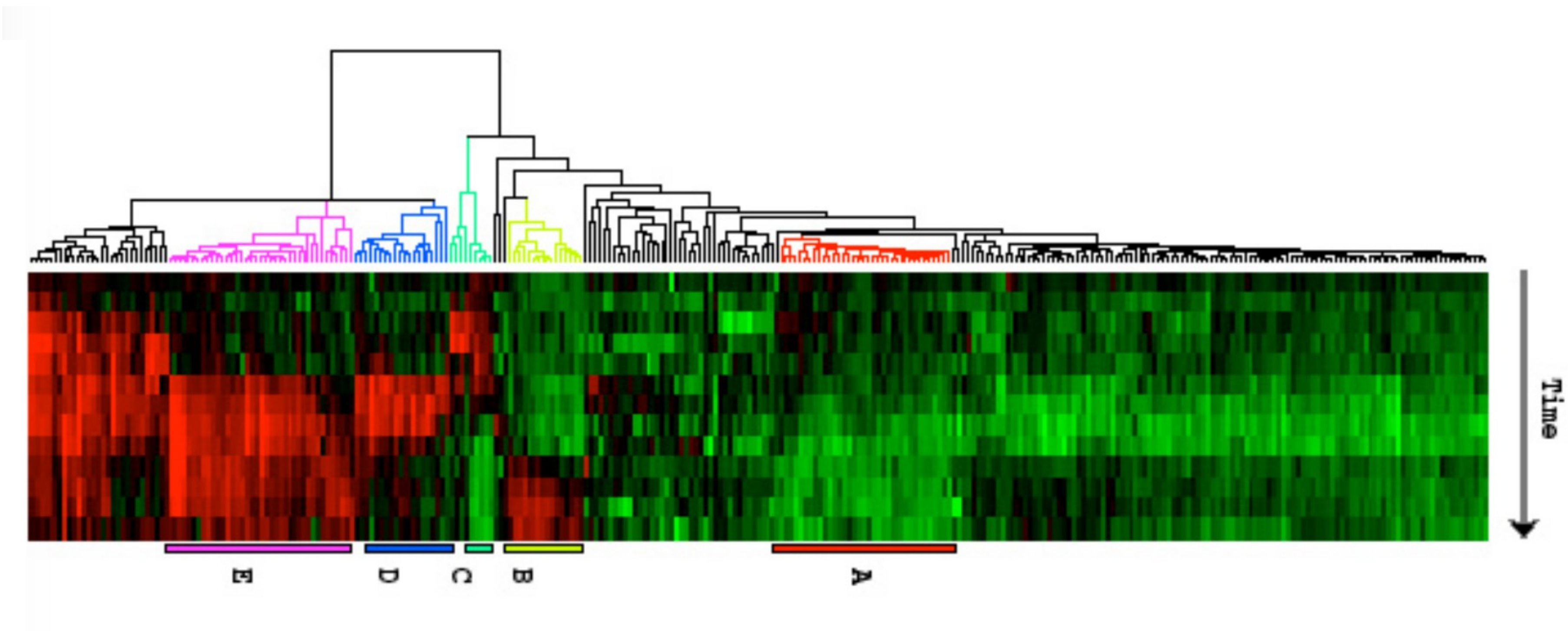


Eisen et al, PNAS 1998

https://hbctraining.github.io/scRNA-seq/lessons/07_SC_clustering_cells_SCT.html

Clustering

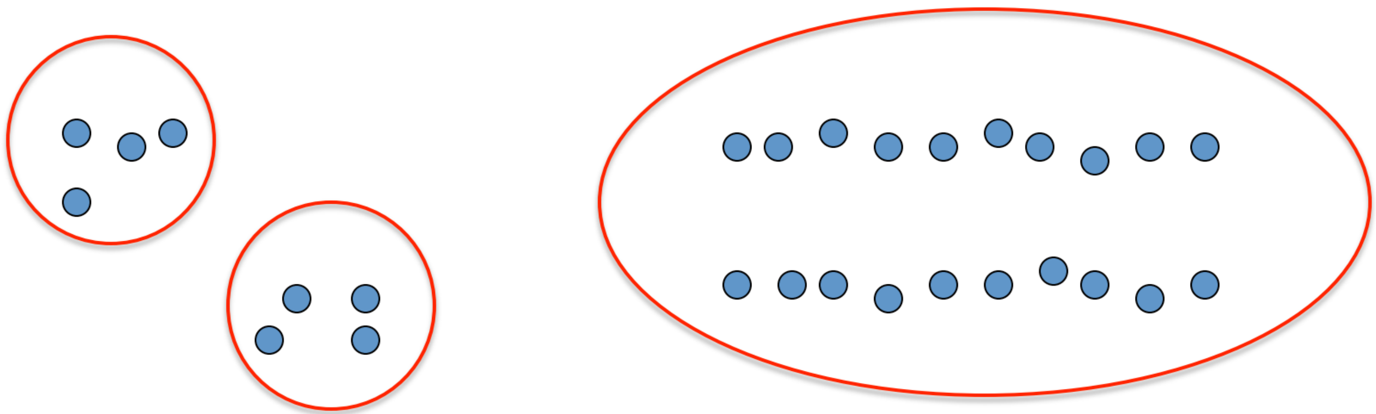
Clustering gene expression data



Eisen et al, PNAS 1998

Clustering

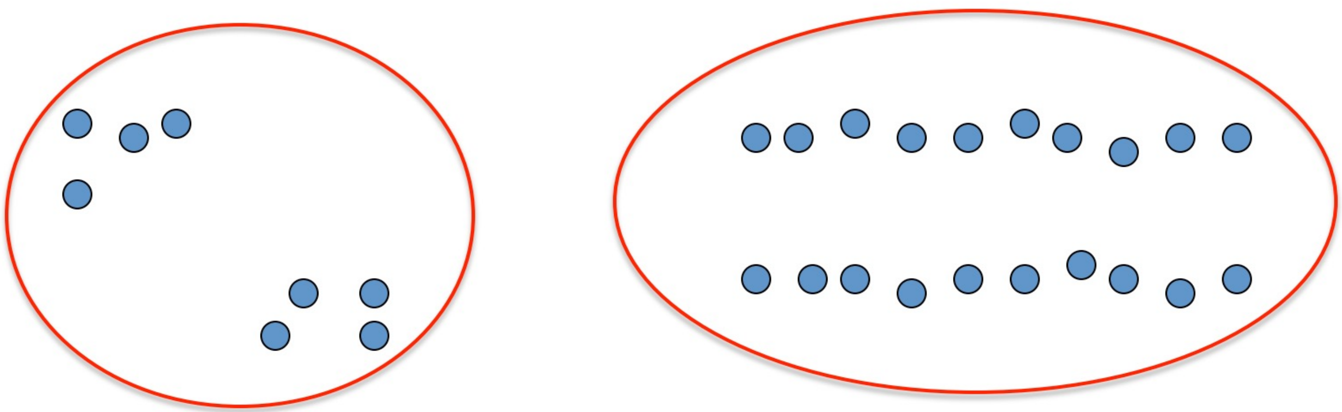
the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters)



http://data-science-sequencing.github.io/Win2018/lectures/lecture19/

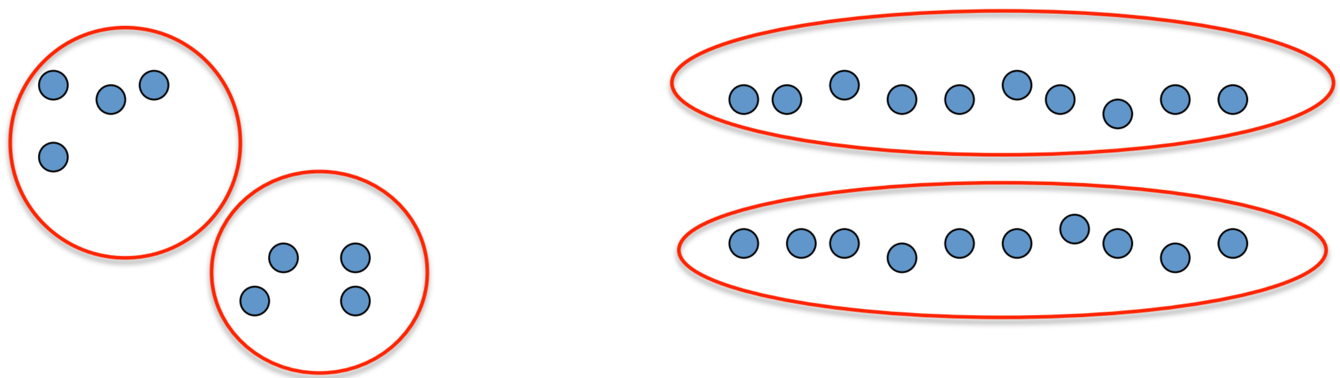
Clustering

the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters)



Clustering

the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters)



Clustering is challenging… because how would you describe similarity?

http://data-science-sequencing.github.io/Win2018/lectures/lecture19/

Image segmentation

Goal: Identify groups of pixels that go together

https://towardsdatascience.com/review-pspnet-winner-in-ilsvrc-2016-semantic-segmentation-scene-parsing-e089e5df177d

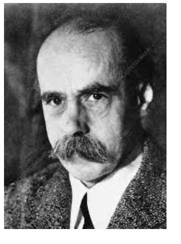
Image segmentation

Goal: Identify groups of pixels that go together

Psychologists identified series of factors that predispose set of elements to be grouped (by human visual system)

"I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have "327"? No. I have sky, house, and trees."

Max Wertheimer, Psychologist (1880-1943)



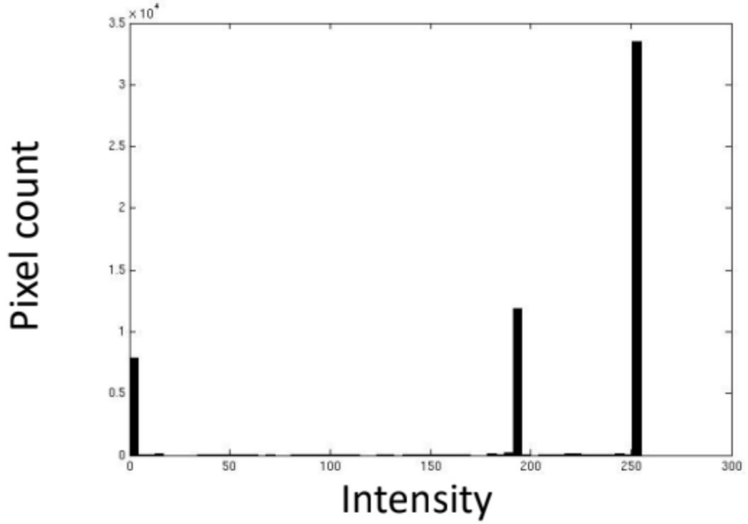
https://towardsdatascience.com/review-pspnet-winner-in-ilsvrc-2016-semantic-segmentation-scene-parsing-e089e5df177d

Image segmentation

Input image

Image segmentation

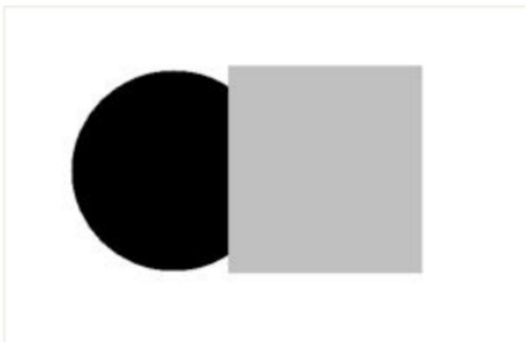
Input image



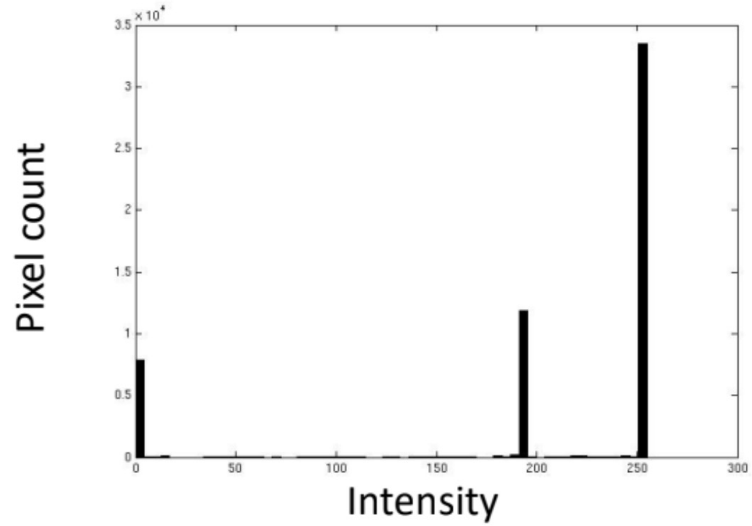
Intensities define the three groups.

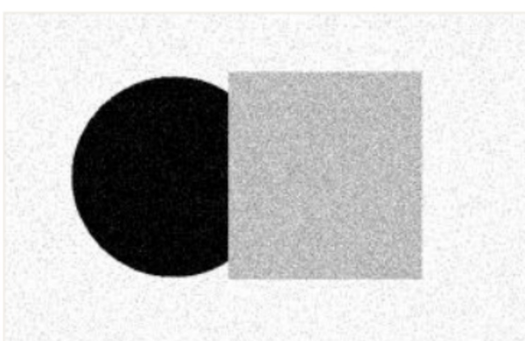
- Segment the image based on the intensity feature.
- What if the image isn't quite so simple?

Image segmentation



Input image





Input image

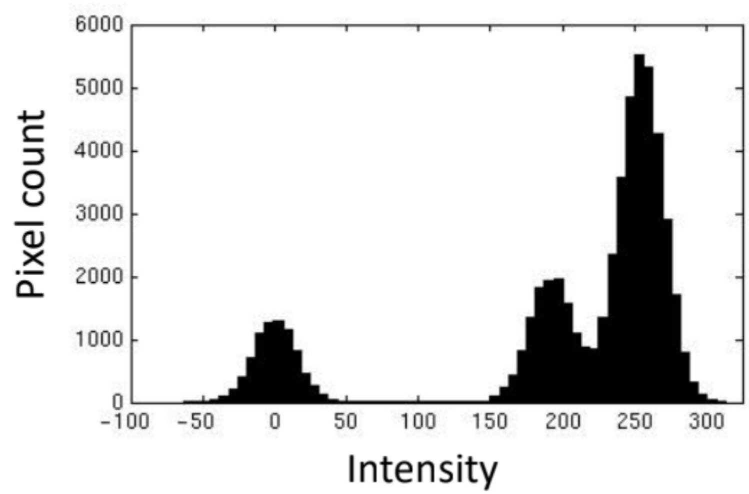
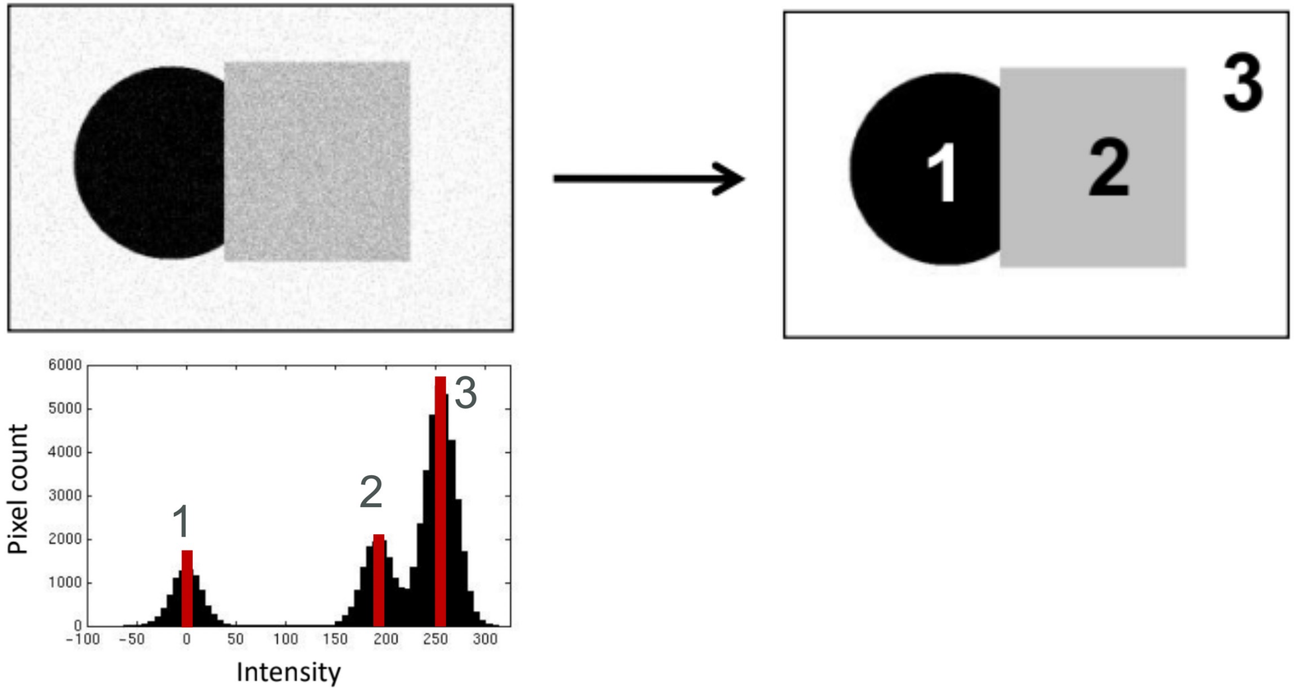


Image segmentation

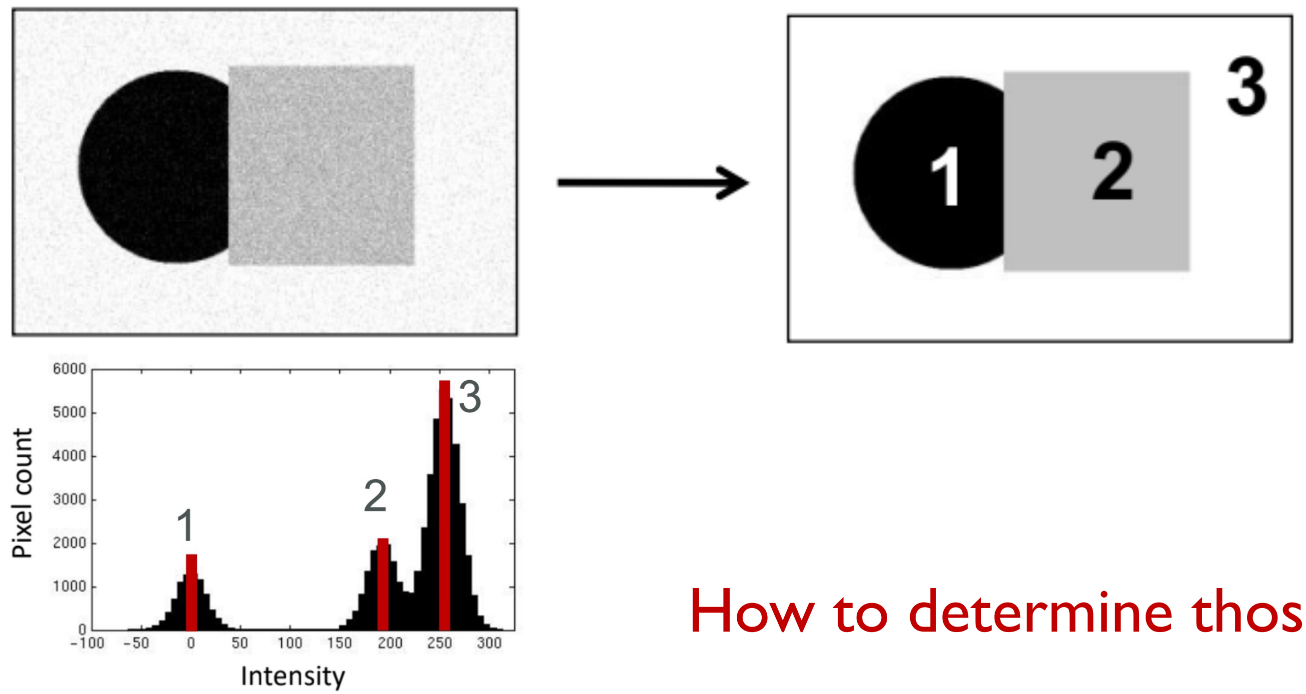


Step 1: Choose three "centers" as the representative intensities;

Step 2: Label every pixel according to which of these centers it is nearest to

Slide credit: Kristen Grauman

Image segmentation

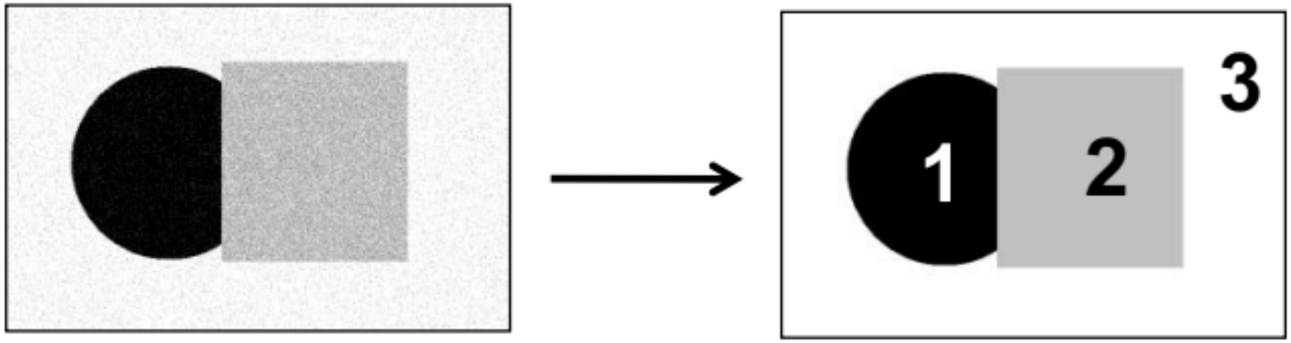


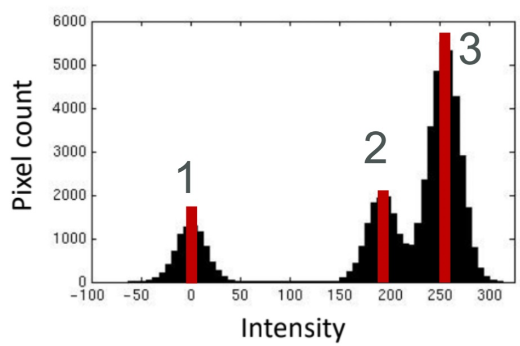
How to determine those centers?

Step 1: Choose three "centers" as the representative intensities;

Step 2: Label every pixel according to which of these centers it is nearest to

Image segmentation





How to determine those centers?

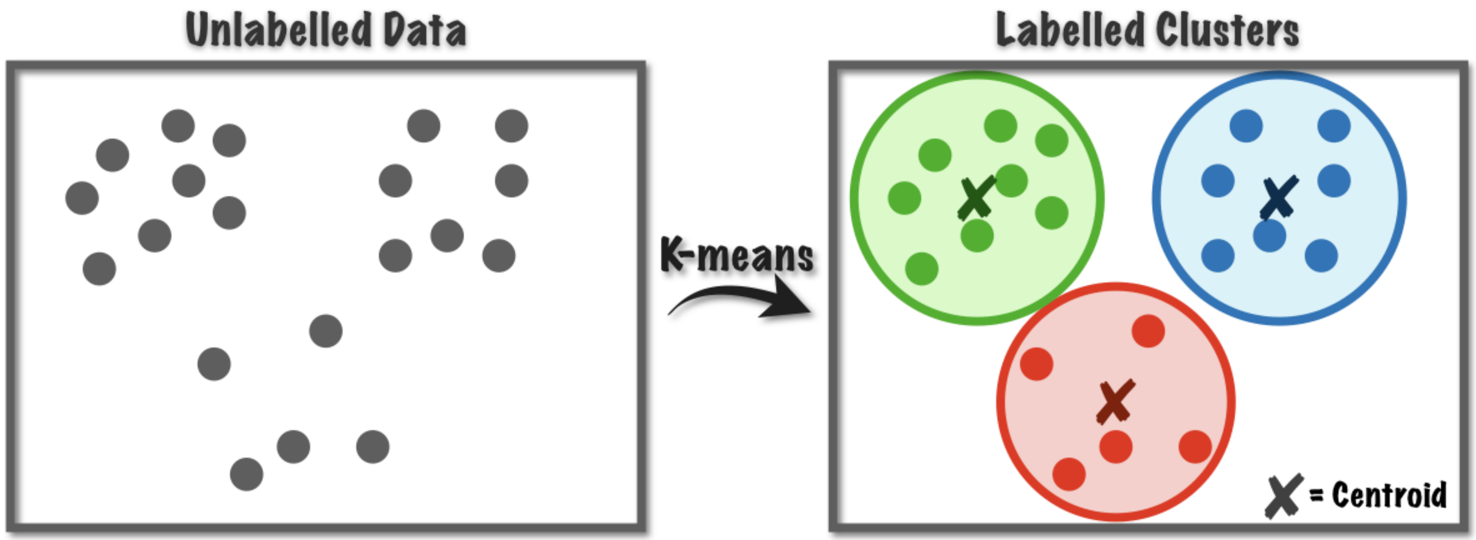
Best cluster centers are those that minimize Sum of Square Distance (SSD) between all points and their nearest cluster center

$$SSD(c_1, c_2, c_3) = \sum_{x_i} \min_j (x_i - c_j)^2$$

Slide credit: Kristen Grauman

What will we learn today?

K-means clustering



https://medium.com/@luigi.fiori.lf0303/k-means-clustering-using-python-db57415d26e6

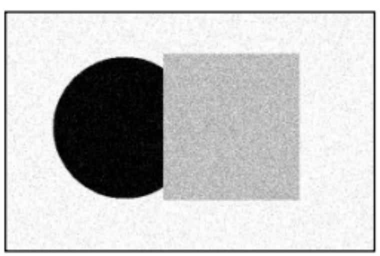
K-means clustering

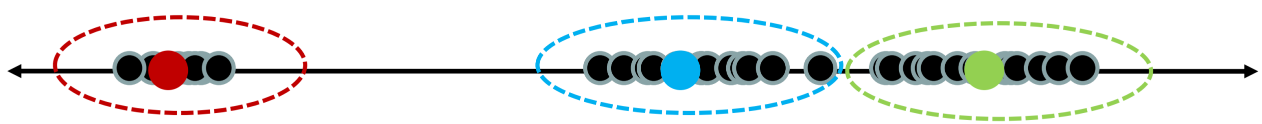
Goal: cluster to minimize variance in data given clusters

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \sum_{x_i} \min_j (x_i - c_j)^2$$

data center

find cluster center membership assignment

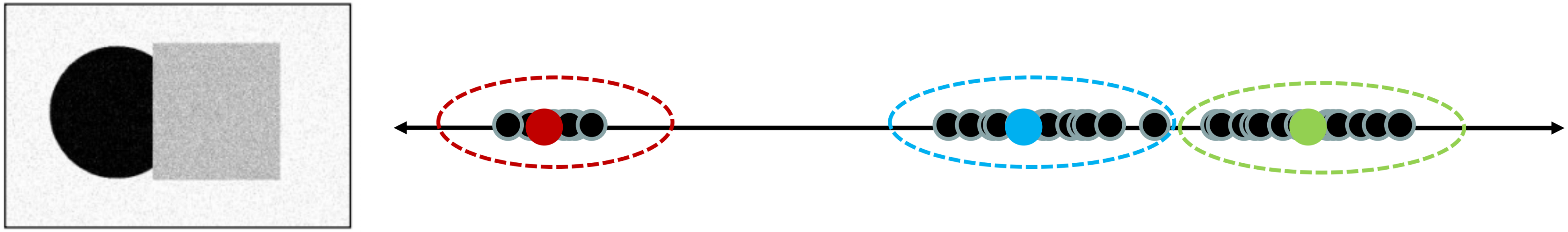




K-means clustering

With this objective, it is a "chicken and egg" problem:

- If we knew the cluster centers, we could allocate points to groups by assigning each to its closest center
- If we knew the group memberships, we could get the centers by computing the mean per group



K-means clustering: Optimization

Goal: cluster to minimize variance in data given clusters

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \sum_{x_i} \min_j (x_i - c_j)^2$$

$$\Updownarrow$$

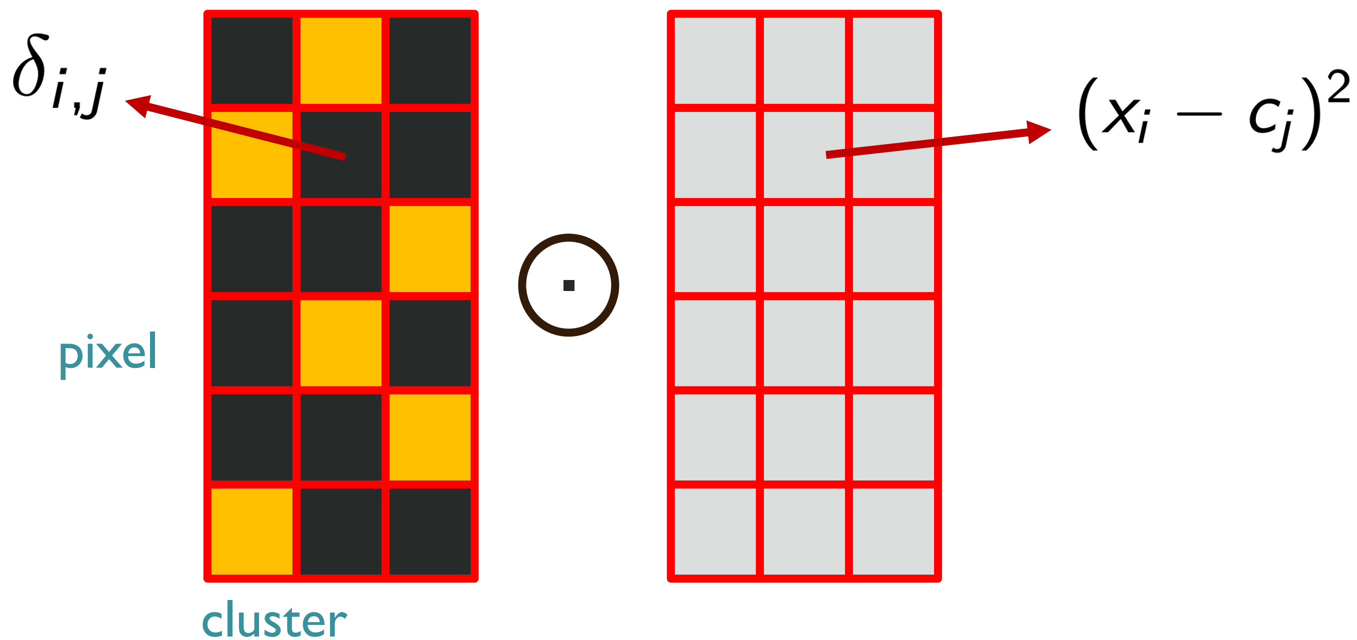
$$\boxed{\begin{array}{c} \mathbf{c}^*, \delta^* = \arg\min_{\mathbf{c},\delta} \sum_{x_i} \sum_{c_j} \delta_{i,j} (x_i - c_j)^2 \\ \delta \in \{0,1\}^{N \times K}, \delta 1 = 1 \end{array}}$$

membership matrix

K-means clustering: Optimization

Goal: cluster to minimize variance in data given clusters

$$\mathbf{c}^*, \delta^* = \arg\min_{\mathbf{c},\delta} \boxed{\sum_{x_i} \sum_{c_j} \delta_{i,j}(x_i - c_j)^2} \qquad \delta \in \{0,1\}^{N \times K}, \delta 1 = 1$$



K-means clustering: Iterative algorithm

$$\boxed{\mathbf{c}^*, \delta^* = \arg\min_{\mathbf{c},\delta} \sum_{x_i} \sum_{c_j} \delta_{i,j}(x_i - c_j)^2 \quad \delta \in \{0,1\}^{N\times K}, \delta 1 = 1}$$

1. Initialize: cluster centers

2. Assign each point to the closest center

$$\delta^{(t)} = \arg\min_{\delta} \sum_{x_i} \sum_{c_j^{(t-1)}} \delta_{i,j}(x_i - c_j^{(t-1)})^2$$

3. Update cluster centers as the mean of the points

$$\mathbf{c}^{(t)} = \arg\min_{\mathbf{c}} \sum_{x_i} \sum_{c_j} \delta_{i,j}^{(t)}(x_i - c_j)^2$$

4. Repeat Step 2-3 till stopped

K-means clustering: Iterative algorithm

$$\boxed{\mathbf{c}^*, \delta^* = \arg\min_{\mathbf{c},\delta} \sum_{x_i} \sum_{c_j} \delta_{i,j}(x_i - c_j)^2 \quad \delta \in \{0,1\}^{N\times K}, \delta 1 = 1}$$

1. Initialize: cluster centers

2. Assign each point to the closest center

$$\delta^{(t)} = \arg\min_{\delta} \sum_{x_i} \sum_{c_j^{(t-1)}} \delta_{i,j}(x_i - c_j^{(t-1)})^2$$

$$\Updownarrow$$

Solution: $\delta_{i,j^*} = 1, \textit{when } j^* = \arg\min_j (x_i - c_j)^2$

3. Update cluster centers as the mean of the points

$$\mathbf{c}^{(t)} = \arg\min_{\mathbf{c}} \sum_{x_i} \sum_{c_j} \delta_{i,j}^{(t)}(x_i - c_j)^2 \quad \Leftrightarrow$$

Solution: $c_j^{(t)} = \frac{1}{\sum_i \delta_{i,j}^{(t)}} \sum_i x_i \delta_{i,j}^{(t)}$

4. Repeat Step 2-3 till stopped

K-means clustering: Iterative algorithm

$$\mathbf{c}^*, \delta^* = \arg\min_{\mathbf{c},\delta} \sum_{x_i} \sum_{c_j} \delta_{i,j}(x_i - c_j)^2 \qquad \delta \in \{0,1\}^{N\times K}, \delta 1 = 1$$

1. Initialize: cluster centers Random/ K-means++

2. Assign each point to the closest center

$$\delta^{(t)} = \arg\min_{\delta} \sum_{x_i} \sum_{c_j^{(t-1)}} \delta_{i,j}(x_i - c_j^{(t-1)})^2$$

$\Updownarrow$

Solution: $\delta_{i,j^*} = 1, \mathit{when}\ j^* = \arg\min_j (x_i - c_j)^2$

3. Update cluster centers as the mean of the points

$$\mathbf{c}^{(t)} = \arg\min_{\mathbf{c}} \sum_{x_i} \sum_{c_j} \delta_{i,j}^{(t)}(x_i - c_j)^2 \Leftrightarrow$$

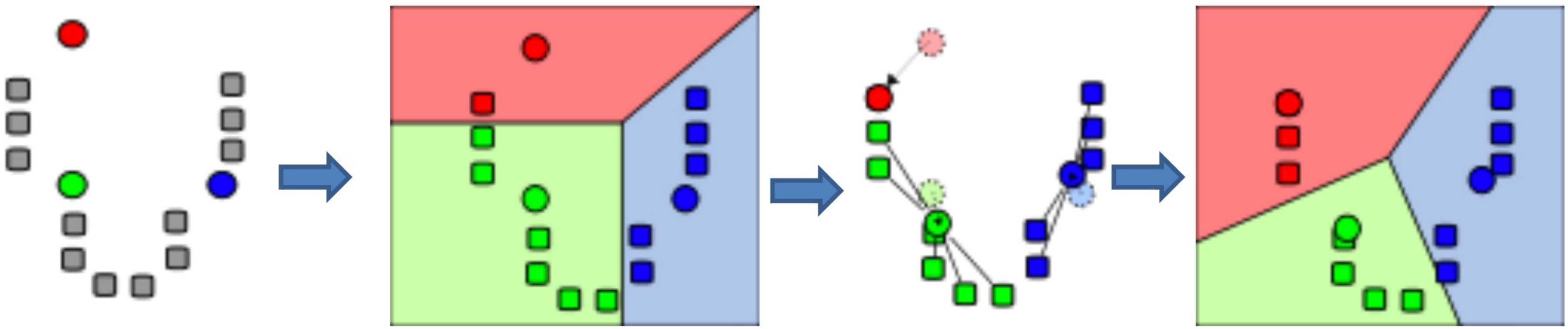
Solution: $c_j^{(t)} = \frac{1}{\sum_i \delta_{i,j}^{(t)}} \sum_i x_i \delta_{i,j}^{(t)}$

4. Repeat Step 2-3 till stopped

Convergence analysis

K-means clustering: Iterative algorithm

$$\mathbf{c}^*, \delta^* = \arg\min_{\mathbf{c},\delta} \sum_{x_i} \sum_{c_j} \delta_{i,j}(x_i - c_j)^2 \quad \delta \in \{0,1\}^{N\times K}, \delta 1 = 1$$



1. Initialize Cluster Centers

2. Assign Points to Clusters

3. Re-compute Means

Repeat (2) and (3)

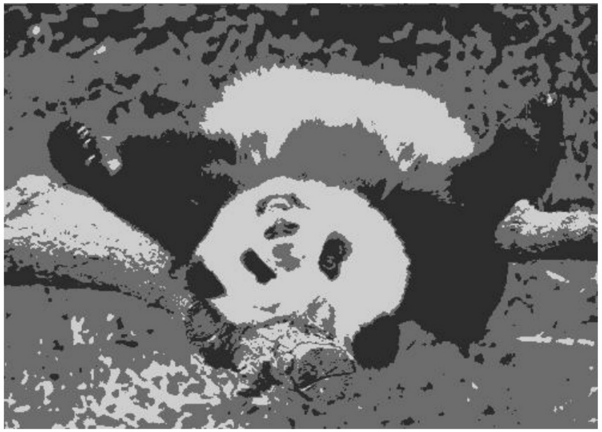
K-means clustering for segmentation

https://github.com/Yuvrajchopra25/Project-8-Image-Segmentation-using-Sklearn-and-K-Means

K-means clustering for segmentation

Original image

2 clusters

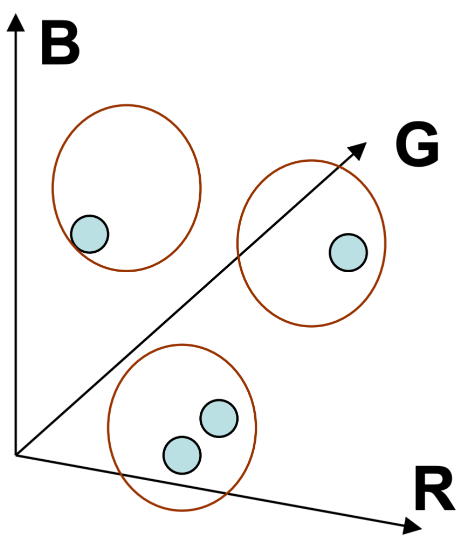


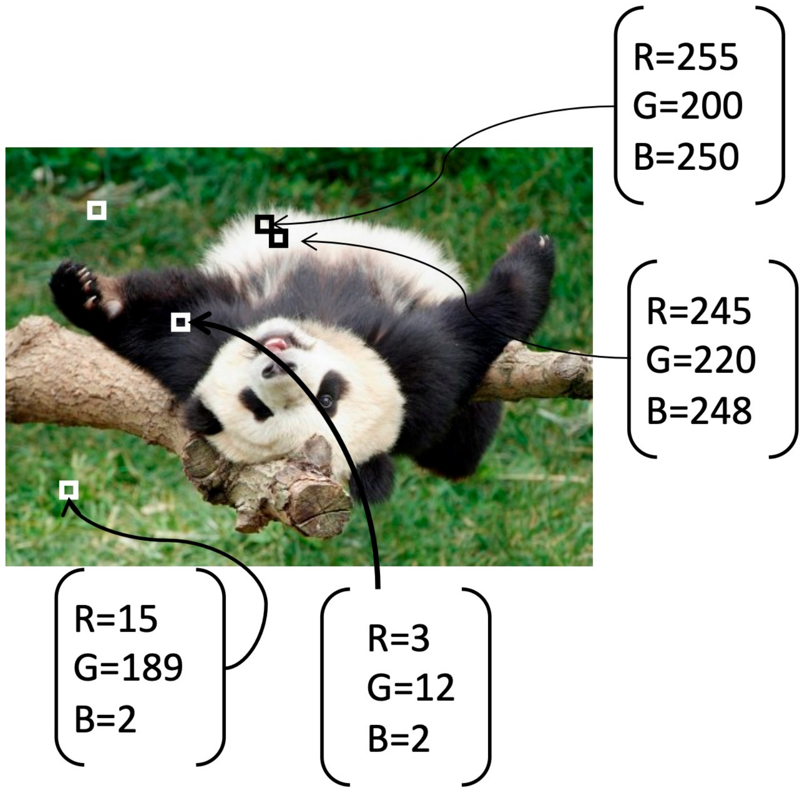
3 clusters

K-means clustering for segmentation

Depending on what we choose as the feature space, we can group pixels in different ways

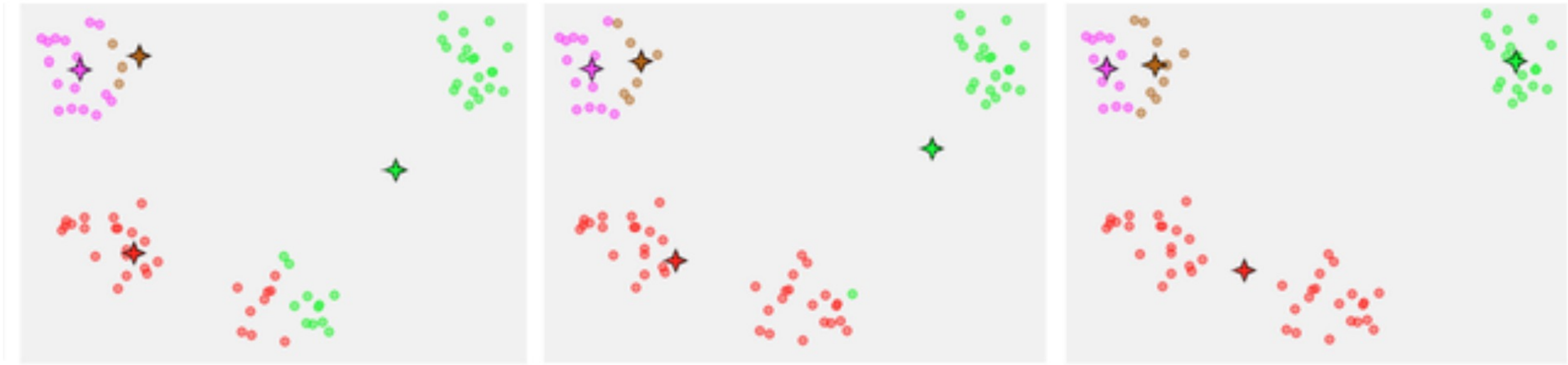
feature space





K-means clustering: Issues

- Converge to local minimum



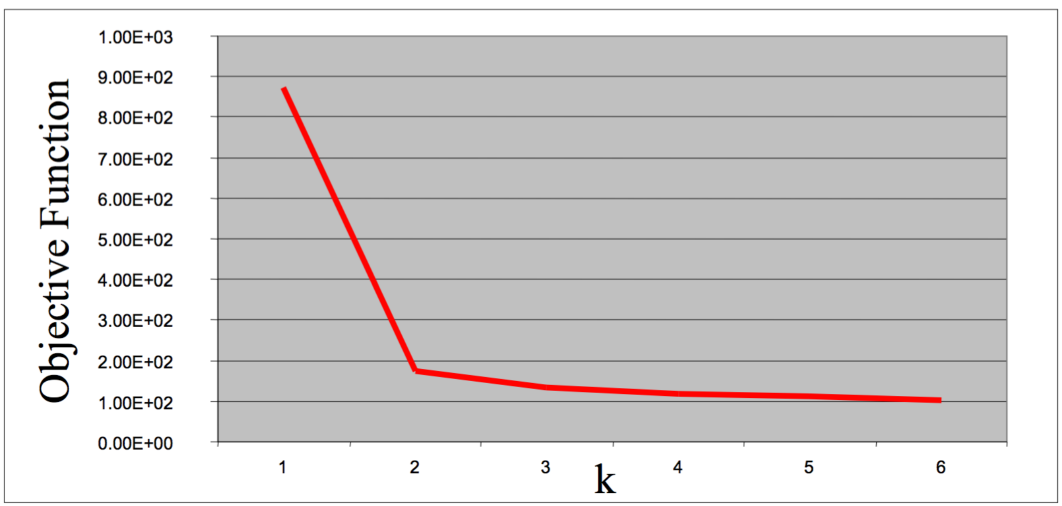
Solution: Initialize multiple times!

K-means clustering: Issues

- Need to pick the number of centers K

We can plot the objective function values for k equals 1 to 6…

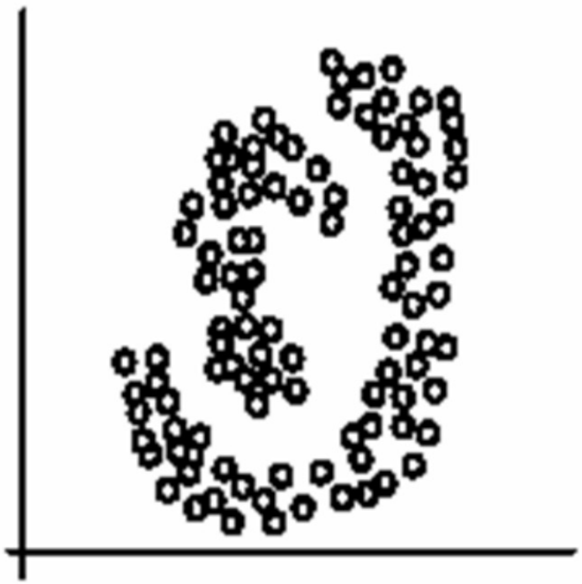
The abrupt change at k = 2, is highly suggestive of two clusters in the data. This technique for determining the number of clusters is known as “knee finding” or “elbow finding”.



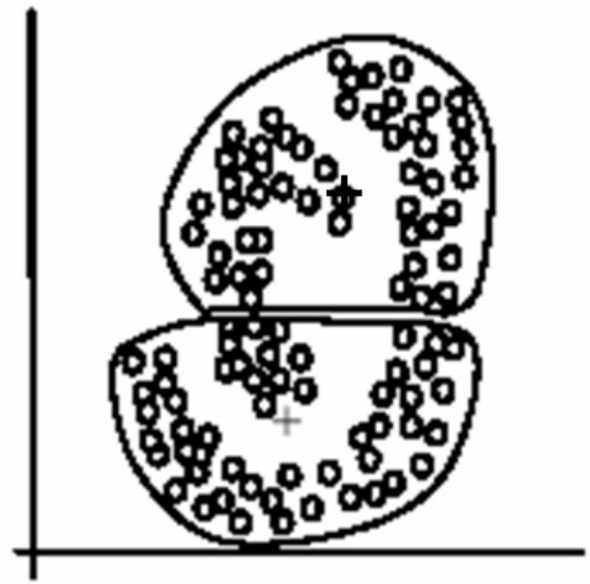
Solutions: Try different numbers of clusters and look at performance!

K-means clustering: Issues

- Wrong similarity metric



(A): Two natural clusters



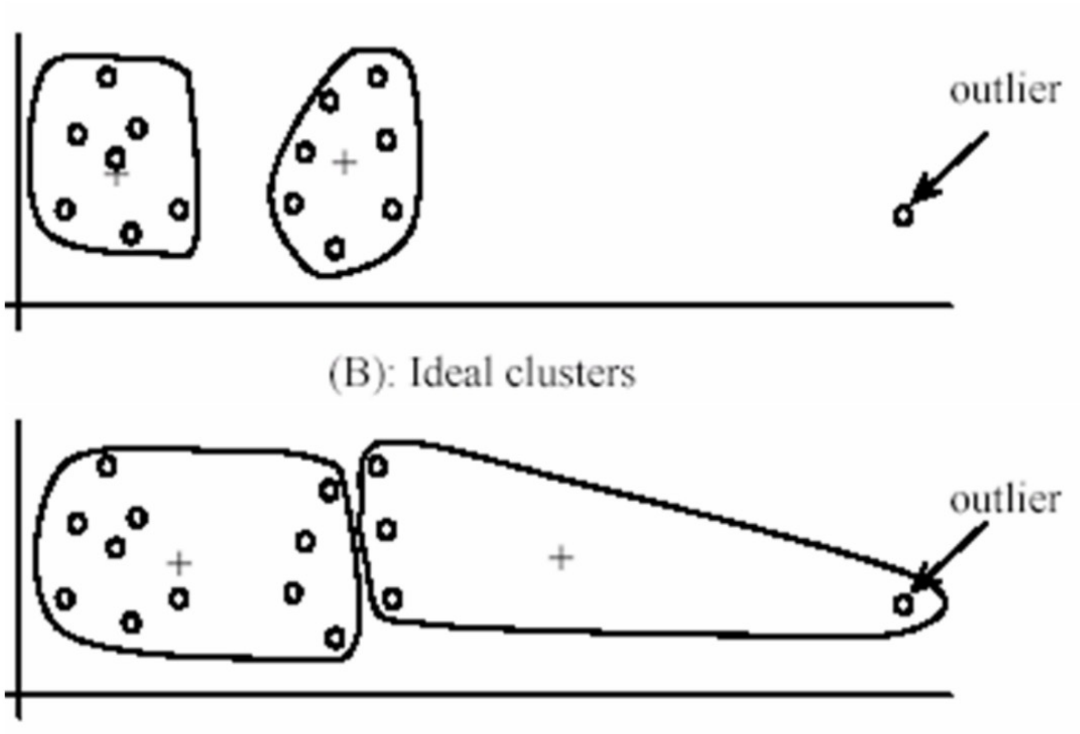
(B): *k*-means clusters

Solutions: Adaptive similarity metric!

Slide: Derek Hoiem

K-means clustering: Issues

- Sensitive to outliers



Solutions: Consider density!

Slide: Derek Hoiem

K-means clustering

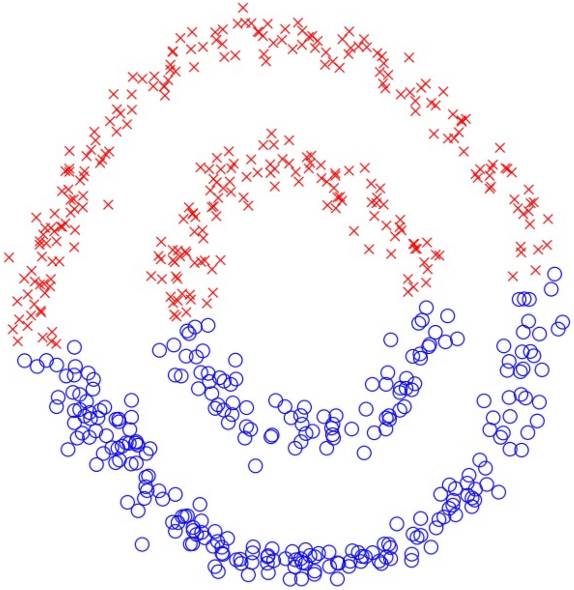
Pros

- Finds cluster centers that minimize conditional variance
- Simple and fast, Easy to implement

Cons

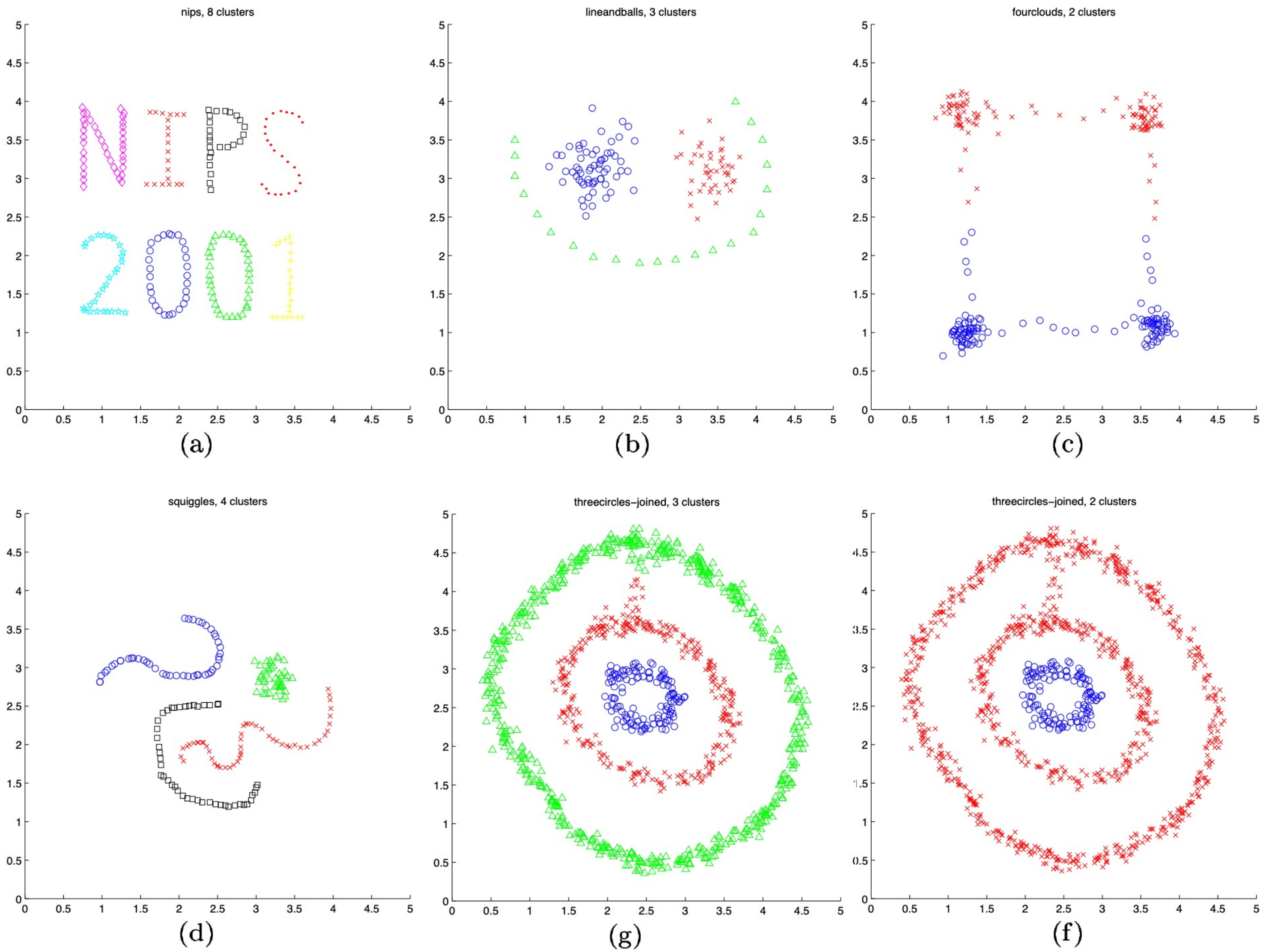
- Prone to local minima
- Need to choose K
- Sensitive to outliers
- All clusters have the same parameters (e.g., distance measure is non -adaptive)

K-means clustering



two circles, 2 clusters

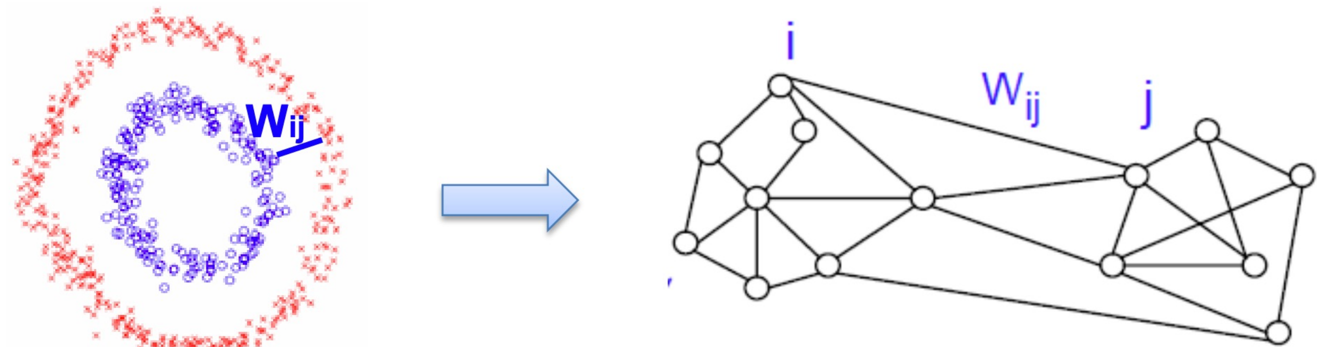
Spectral clustering



Andrew Ng, Michael Jordan, Yair Weiss, On Spectral Clustering: Analysis and an algorithm, NeurIPS 2001

Spectral clustering

Group points based on links in a graph



Data clustering

$G = \{V,E\}$

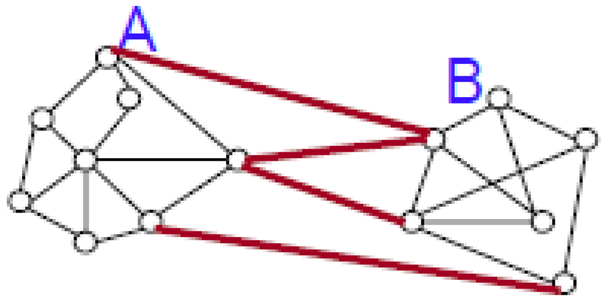
$$W_{ij} = e^{\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

Andrew Ng, Michael Jordan, Yair Weiss, On Spectral Clustering: Analysis and an algorithm, NeurIPS 2001

Spectral clustering

Min-cut: Partition graph into two sets A and B such that weight of edges connecting vertices in A to vertices in B is minimum.

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



Notations

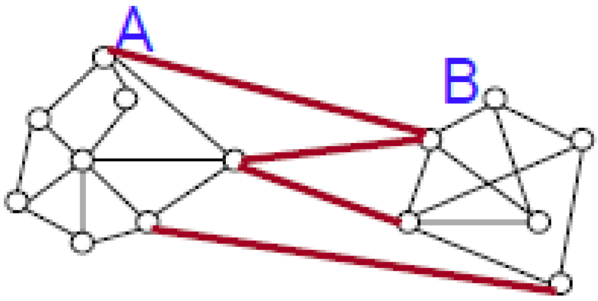
- $W = (w_{ij})$ adjacency matrix of the graph
- $d_i = \sum_j w_{ij}$ degree of a vertex
- $D = diag(d_1, \ldots, d_n)$ degree matrix
- $|A| =$ number of vertices in A
- $\text{vol}(A) = \sum_{i \in A} d_i$

https://www.ml.uni-saarland.de/code/GraphDemo/DemoSpectralClustering.htm

Spectral clustering

Min-cut: Partition graph into **two nonoverlapping** sets A and B such that weight of edges connecting vertices in A to vertices in B is minimum.

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



$$\text{Ncut}(A, B) := \text{cut}(A, B)\left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)}\right)$$

$$\text{vol}(A) = \sum_{i \in A} d_i$$

Spectral clustering

Min-cut: Partition graph into **two nonoverlapping** sets A and B such that weight of edges connecting vertices in A to vertices in B is minimum.

$$\mathrm{Ncut}(A, B) := \mathrm{cut}(A, B)\left(\frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)}\right)$$

$$= \frac{\mathbf{f}^T \mathbf{L} \mathbf{f}}{\mathbf{f}^T \mathbf{D} \mathbf{f}}$$

$f = [f_1\ f_2\ \dots\ f_n]^T$ with $f_i = \begin{cases} \frac{1}{\mathrm{vol}(A)} & \text{if } i \in A \\ -\frac{1}{\mathrm{vol}(B)} & \text{if } i \in B \end{cases}$

- $W = (w_{ij})$ adjacency matrix of the graph
- $d_i = \sum_j w_{ij}$ degree of a vertex
- $D = diag(d_1, \dots, d_n)$ degree matrix
- $L = D - W$

Spectral clustering

Min-cut: Partition graph into **two nonoverlapping** sets A and B such that weight of edges connecting vertices in A to vertices in B is minimum.

$$f'Lf = f'Df - f'Wf \qquad L = D - W$$

Spectral clustering

Min-cut: Partition graph into **two nonoverlapping** sets A and B such that weight of edges connecting vertices in A to vertices in B is minimum.

$$L = D - W$$

$$
\begin{aligned}
f'Lf &= f'Df - f'Wf \\
&= \sum_i d_i f_i^2 - \sum_{i,j} f_i f_j w_{ij} \\
&= \frac{1}{2}\left(\sum_i \left(\sum_j w_{ij}\right) f_i^2 - 2\sum_{ij} f_i f_j w_{ij} + \sum_j \left(\sum_i w_{ij}\right) f_j^2\right) \\
&= \frac{1}{2}\sum_{ij} w_{ij}(f_i - f_j)^2 \quad = \sum_{i\in A, j\in B} w_{ij}\left(\frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)}\right)^2
\end{aligned}
$$

$$f = [f_1\ f_2\ \dots\ f_n]^T \quad \text{with} \quad f_i = \begin{cases} \frac{1}{\mathrm{vol}(A)} & \text{if } i \in A \\ -\frac{1}{\mathrm{vol}(B)} & \text{if } i \in B \end{cases}$$

Spectral clustering

Min-cut: Partition graph into **two nonoverlapping** sets A and B such that weight of edges connecting vertices in A to vertices in B is minimum.

$$\mathbf{f}^T\mathbf{D}\mathbf{f} = \sum_j d_i \mathbf{f}_i^2 = \sum_{i \in A} \frac{d_i}{\mathrm{vol}(A)^2} + \sum_{j \in B} \frac{d_i}{\mathrm{vol}(B)^2} = \frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)}$$

$$\mathrm{f} = [\mathrm{f}_1\ \mathrm{f}_2\ \dots\ \mathrm{f}_n]^T \text{ with } \mathrm{f}_i = \begin{cases} \frac{1}{\mathrm{vol}(A)} & \text{if } i \in A \\ -\frac{1}{\mathrm{vol}(B)} & \text{if } i \in B \end{cases}$$

Spectral clustering

Min-cut: Partition graph into **two nonoverlapping** sets A and B such that weight of edges connecting vertices in A to vertices in B is minimum.

$$\text{Ncut}(A,B) := \text{cut}(A,B)\left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)}\right)$$

$$= \frac{\mathbf{f}^T \mathbf{L} \mathbf{f}}{\mathbf{f}^T \mathbf{D} \mathbf{f}}$$

$$= \frac{\sum_{i \in A, j \in B} w_{ij} \left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)}\right)^2}{\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)}}$$

Spectral clustering

minimize $\text{Ncut}(A, B) = \frac{\mathbf{f}^T\mathbf{L}\mathbf{f}}{\mathbf{f}^T\mathbf{D}\mathbf{f}}$

subject to $f = [f_1\ f_2\ \dots\ f_n]^T$ with $f_i = \begin{cases} \frac{1}{\text{vol}(A)} & \text{if } i \in A \\ -\frac{1}{\text{vol}(B)} & \text{if } i \in B \end{cases}$

Relax the constraint

minimize $\text{Ncut}(A, B) = \frac{\mathbf{f}^T\mathbf{L}\mathbf{f}}{\mathbf{f}^T\mathbf{D}\mathbf{f}}$

subject to $f^T D 1 = 0$

Spectral clustering

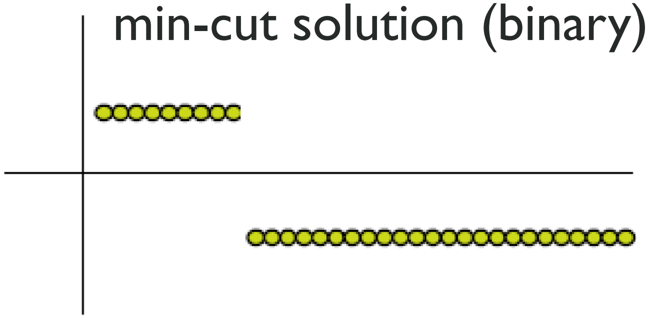
minimize $\quad \text{Ncut}(A, B) = \frac{\mathbf{f}^T \mathbf{L} \mathbf{f}}{\mathbf{f}^T \mathbf{D} \mathbf{f}}$

subject to $\quad f^T D 1 = 0$

second eigenvector of generalized eigendecompostion problem

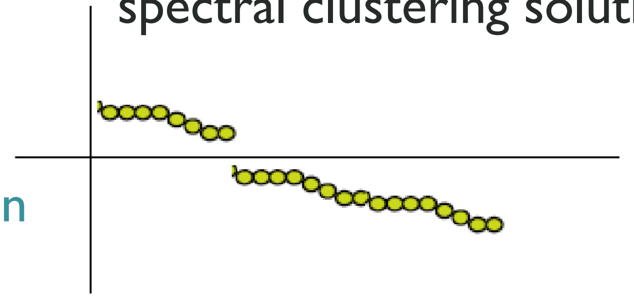
$$\boxed{\mathbf{L}\mathbf{f} = \lambda \mathbf{D}\mathbf{f}}$$

min-cut solution (binary)



Approximation

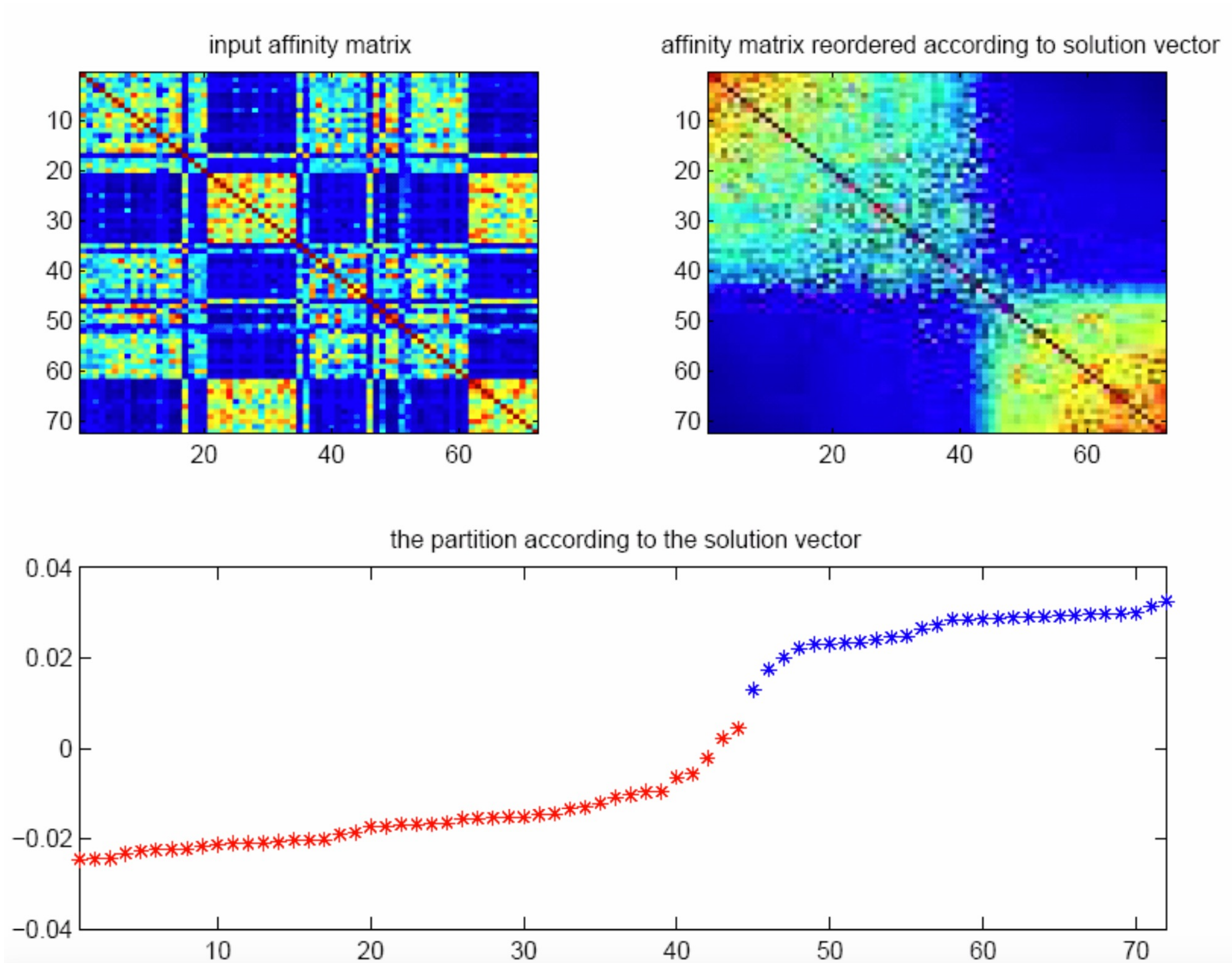
spectral clustering solution



Spectral clustering

Example:

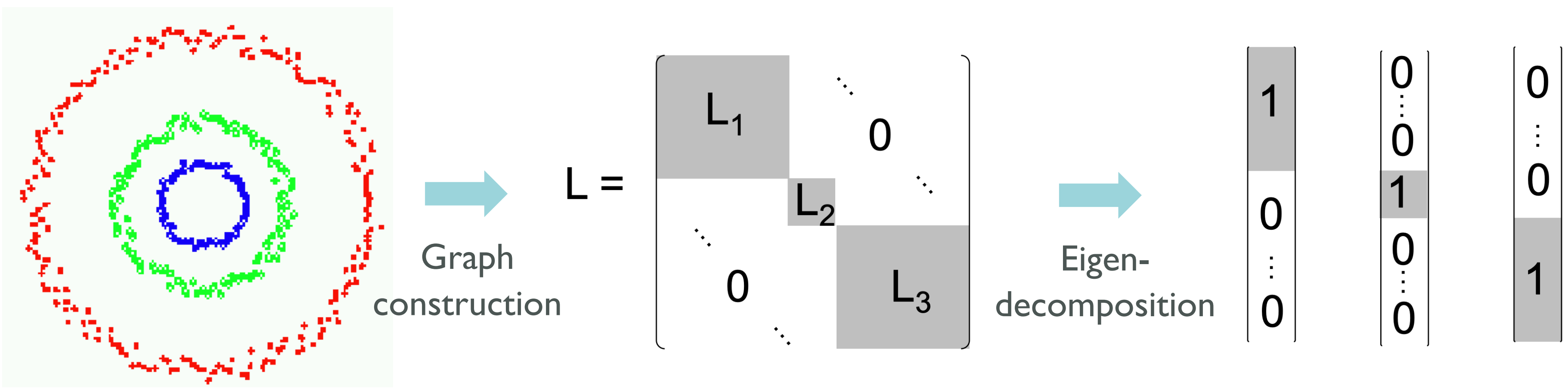
block-wise matrix -> eigenvector with positive / negative elements



Spectral clustering

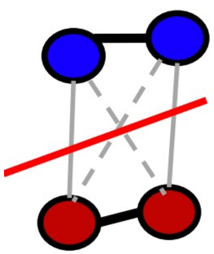
How to partition a graph into k clusters?

graph modeling + k-means



block-wise matrix -> eigenvectors with nonoverlapping nonzero indices

Spectral clustering



ideally

1	1	0	0
1	1	0	0
0	0	1	1
0	0	1	1

Eigen-decomposition

.71
.71
0
0

0
0
.71
.71

Graph construction

1	1	.2	0
1	1	0	.1
.2	0	1	1
0	.1	1	1

Eigen-decomposition

.50
.50
.50
.50

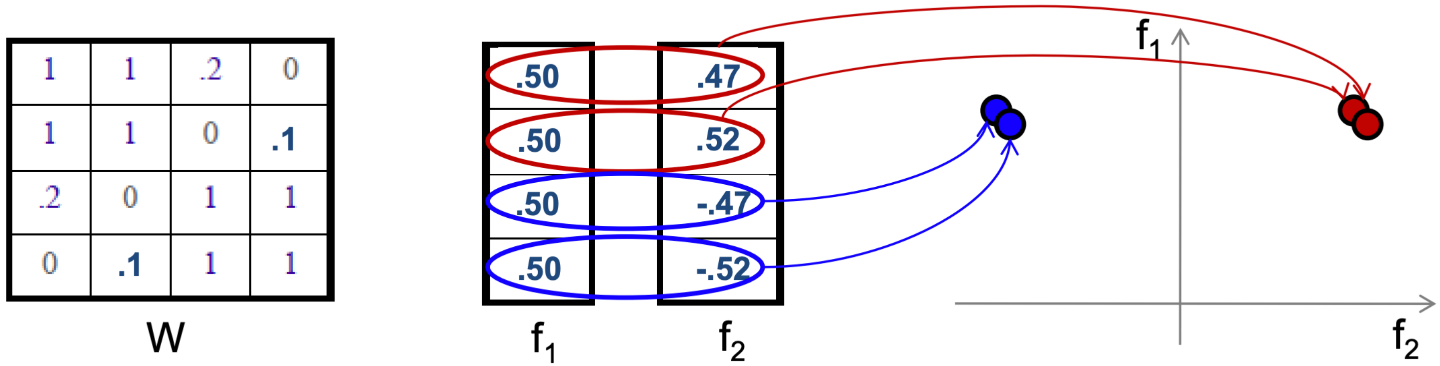
.47
.52
-.47
-.52

1st eigenvector is constant since graph is connected

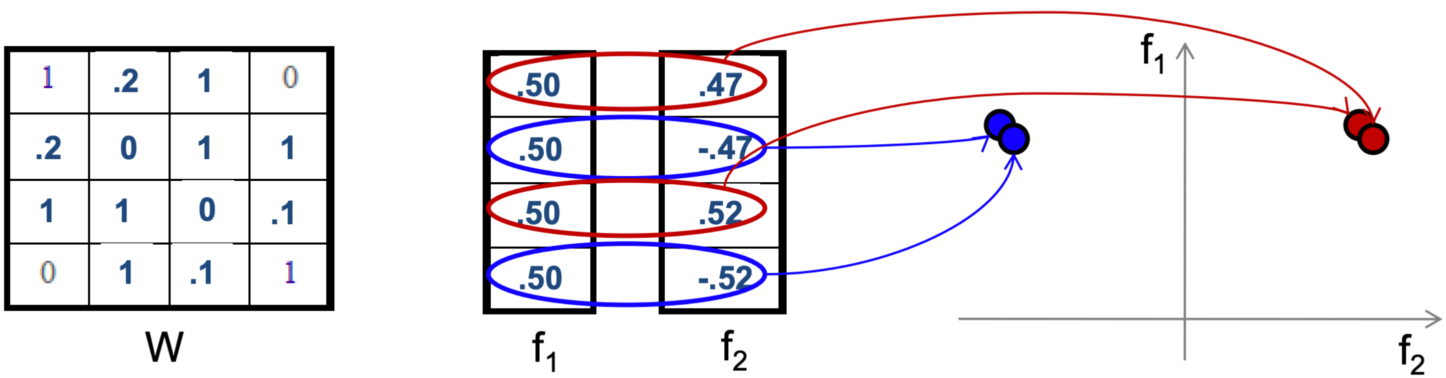
sign of 2nd eigenvector indicates blocks

Spectral clustering

Can put data points into blocks using eigenvectors:

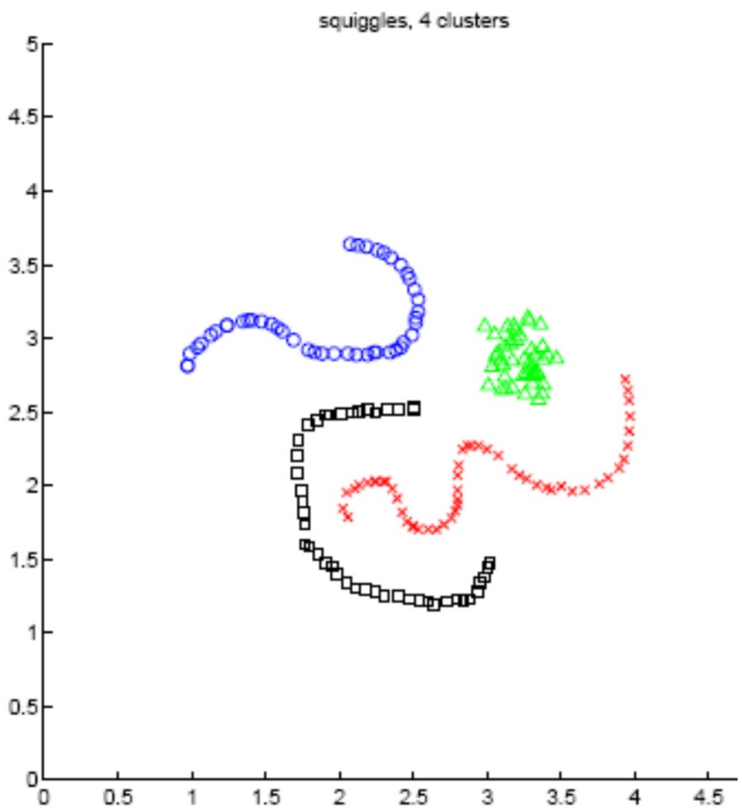


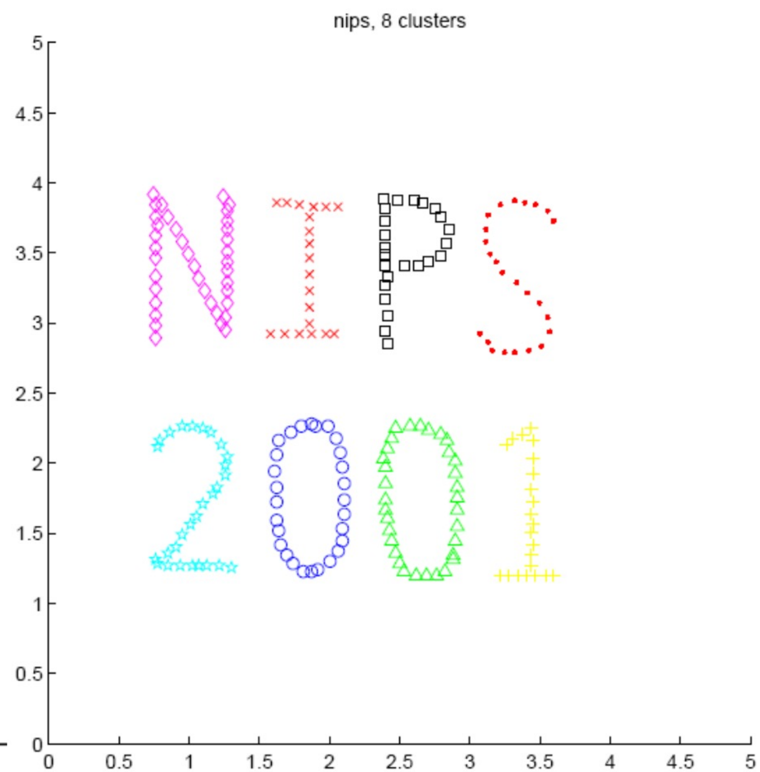
Embedding is same regardless of data ordering:



https://www.ml.uni-saarland.de/code/GraphDemo/DemoSpectralClustering.htm

Spectral clustering

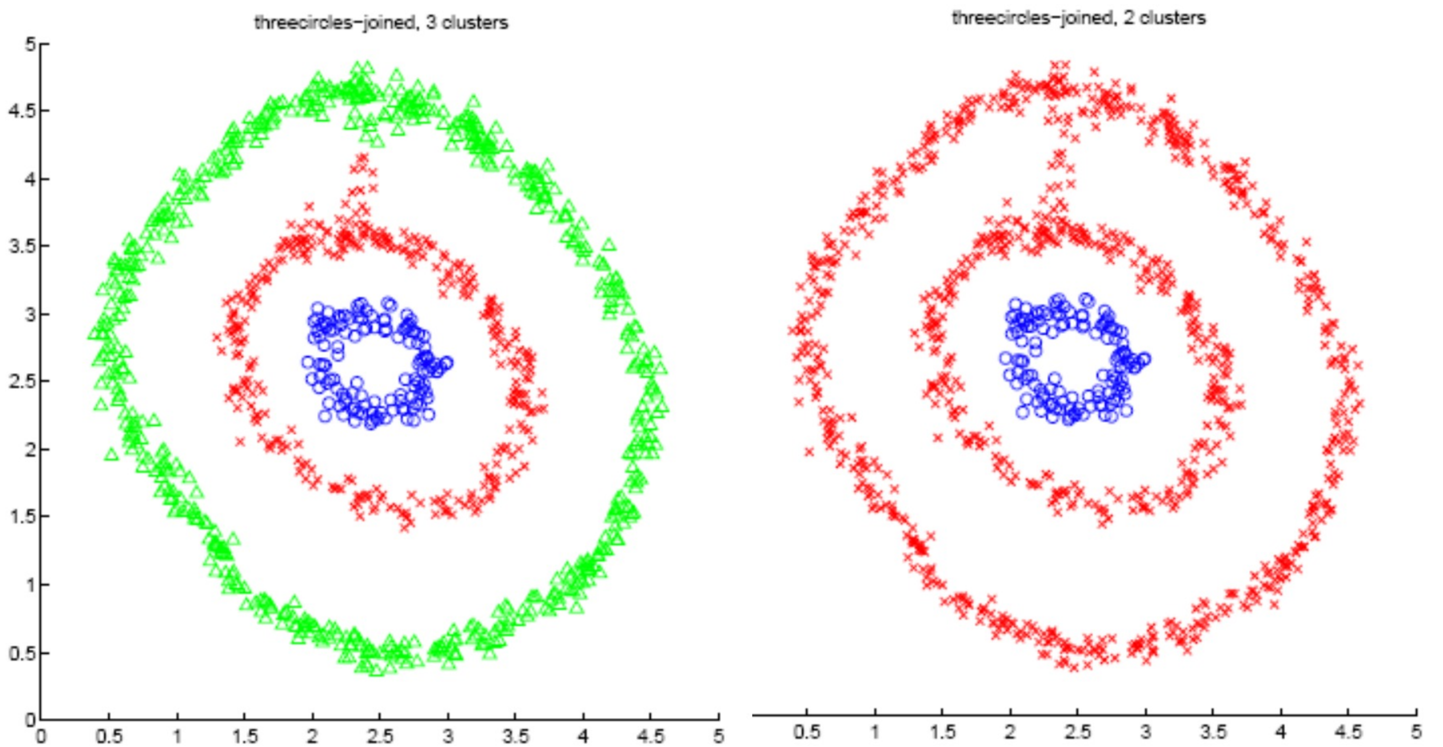




https://www.cs.cmu.edu/~aarti/Class/10701/slides/Lecture21_2.pdf

Spectral clustering

How to determine the number of clusters K?



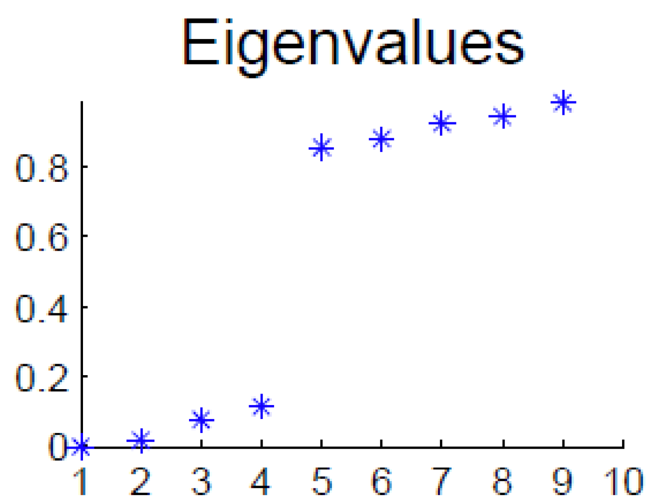
https://www.cs.cmu.edu/~aarti/Class/10701/slides/Lecture21_2.pdf

Spectral clustering

How to determine the number of clusters K?

Most stable clustering is usually given by the value of k that maximizes the eigengap (difference between consecutive eigenvalues)

$$\Delta_k = \left| \lambda_k - \lambda_{k-1} \right|$$

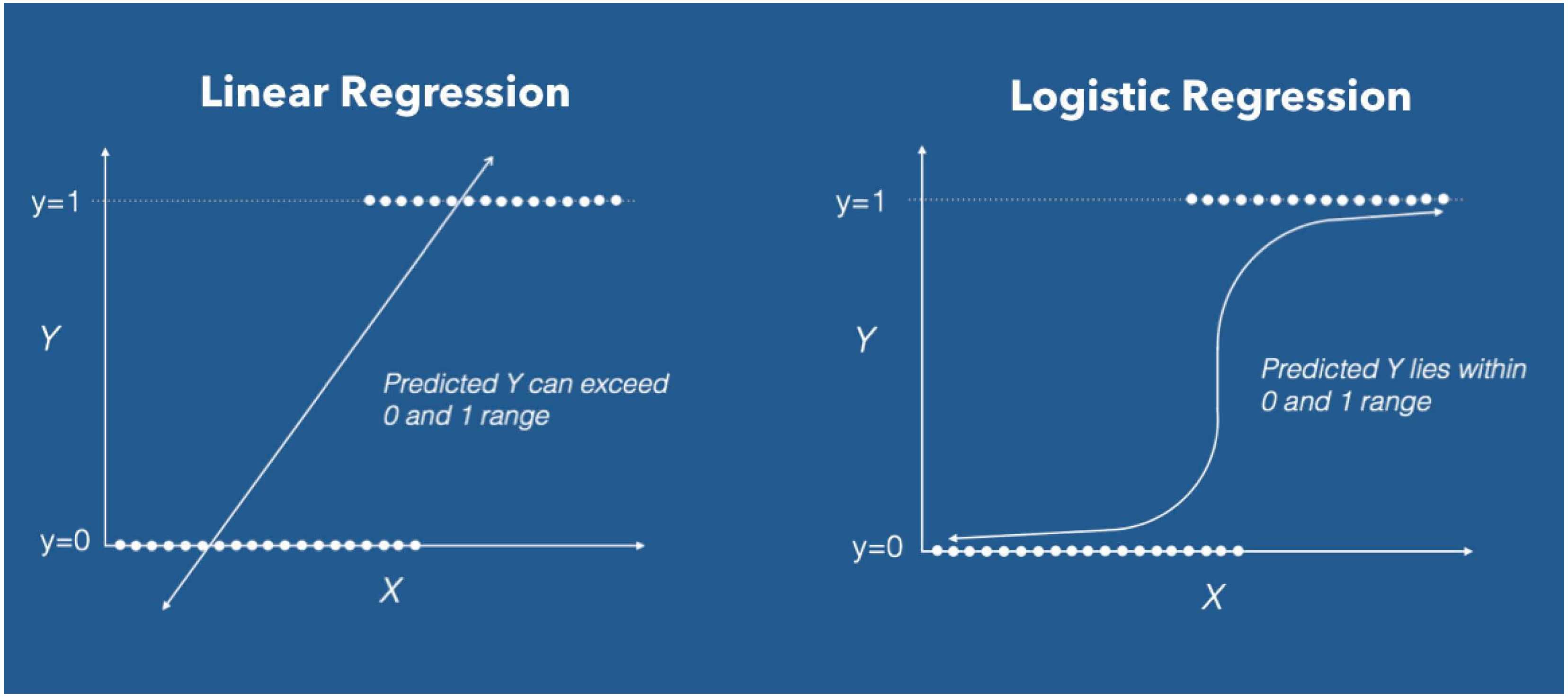


https://www.cs.cmu.edu/~aarti/Class/10701/slides/Lecture21_2.pdf

Spectral clustering

- Algorithms that cluster points using eigenvectors of matrices derived from the data
- Useful in hard non-convex clustering problems
- Obtain data representation in the low-dimensional space that can be easily clustered
- Variety of methods that use eigenvectors of unnormalized or normalized Laplacian, differ in how to derive clusters from eigenvectors, k-way vs repeated 2-way
- Empirically very successful

Next lecture: Logistic regression



Thank you very much!

sihengc@sjtu.edu.cn