ISyE 6416: Computational Statistics
Spring 2017

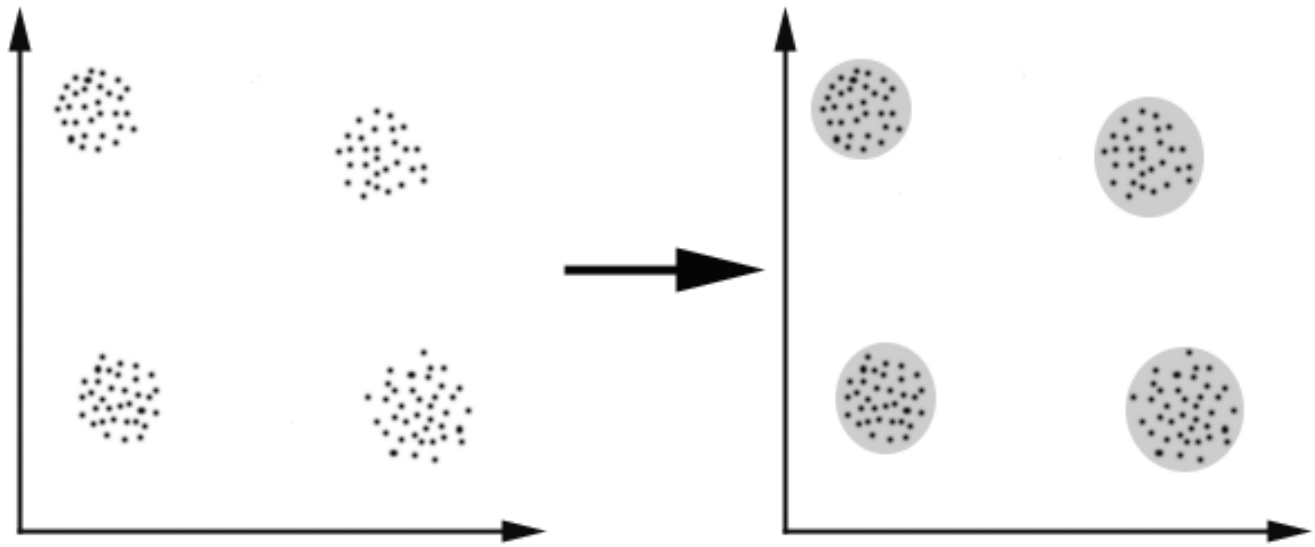
Lecture 6: Clustering

Prof. Yao Xie

H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Clustering

- A basic tool in data mining/pattern recognition
- Divide a set of data into groups
- Samples in one cluster are closer
- Samples in different clusters are far apart
- Usually done without knowing "label" information: unsupervised learning



Approach

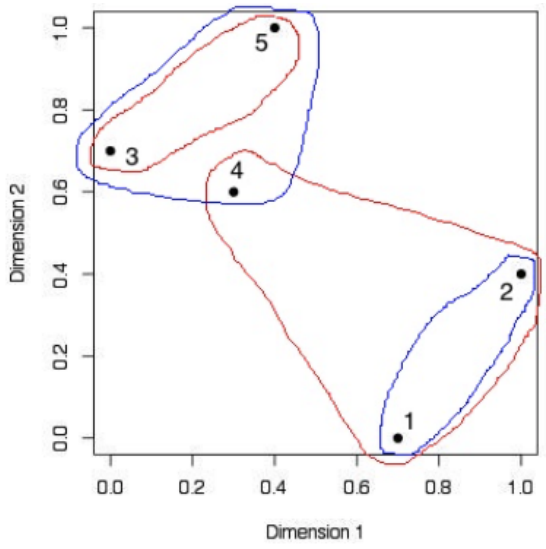
- Represent samples by feature vectors: $X_1, \ldots, X_n$
- Define a distance measure for the closeness between samples $d(X_i, X_j)$ (e.g. Euclidean distance)
- Choose K number of clusters (fixed)
- A clustering of points is an assignment function $C : \mathbb{R}^p \to \{1, \ldots, K\}$ that maps X_i to a group label (short-handed as $C(i)$ for the ith sample)
- $C(i) = k$ means X_i is assigned to group k
- n_k is the number of points in the group k
- Goal: minimize within-cluster scatter

$$W = \frac{1}{2} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{C(i)=k, C(j)=k} d_{ij}$$

Simple example

Here $n = 5$ and $K = 2$,
$X_i \in \mathbb{R}^2$ and $d_{ij} = \|X_i - X_j\|_2^2$

	1	2	3	4	5
1	0	0.25	0.98	0.52	1.09
2	0.25	0	1.09	0.53	0.72
3	0.98	1.09	0	0.10	0.25
4	0.52	0.53	0.10	0	0.17
5	1.09	0.72	0.25	0.17	0



- Red clustering: $W = (0.25 + 0.53 + 0.52)/3 + 0.25/2 = 0.56$
- Blue clustering: $W = 0.25/2 + (0.1 + 0.17 + 0.25)/3 = 0.30$

Blue cluster is better.
courtesy: Ryan Tibshirani.

Solving optimization problem

- To minimize W by choosing C is exponentially complex: a combinatoric problem and one has to enumerate all possible assignment of n points into K groups
- Instead, we will solve this approximately, using K-means

K-means for clustering

- Assume there are K clusters with centroids $\{Z_1, \ldots Z_k\}$
- Each training sample is assigned to one cluster
- Cost function: total mean squared error between the training samples and their representative cluster centroids

$$\arg\min_{Z,C} \sum_{i=1}^{n} \|X_i - Z_{C(i)}\|$$

Two steps in K-means

- Fix centroids, update then assignment using the nearest neighbor rule

$$C(i) = \arg \min_{j \in [1,\ldots,K]} \|X_i - Z_j\|$$

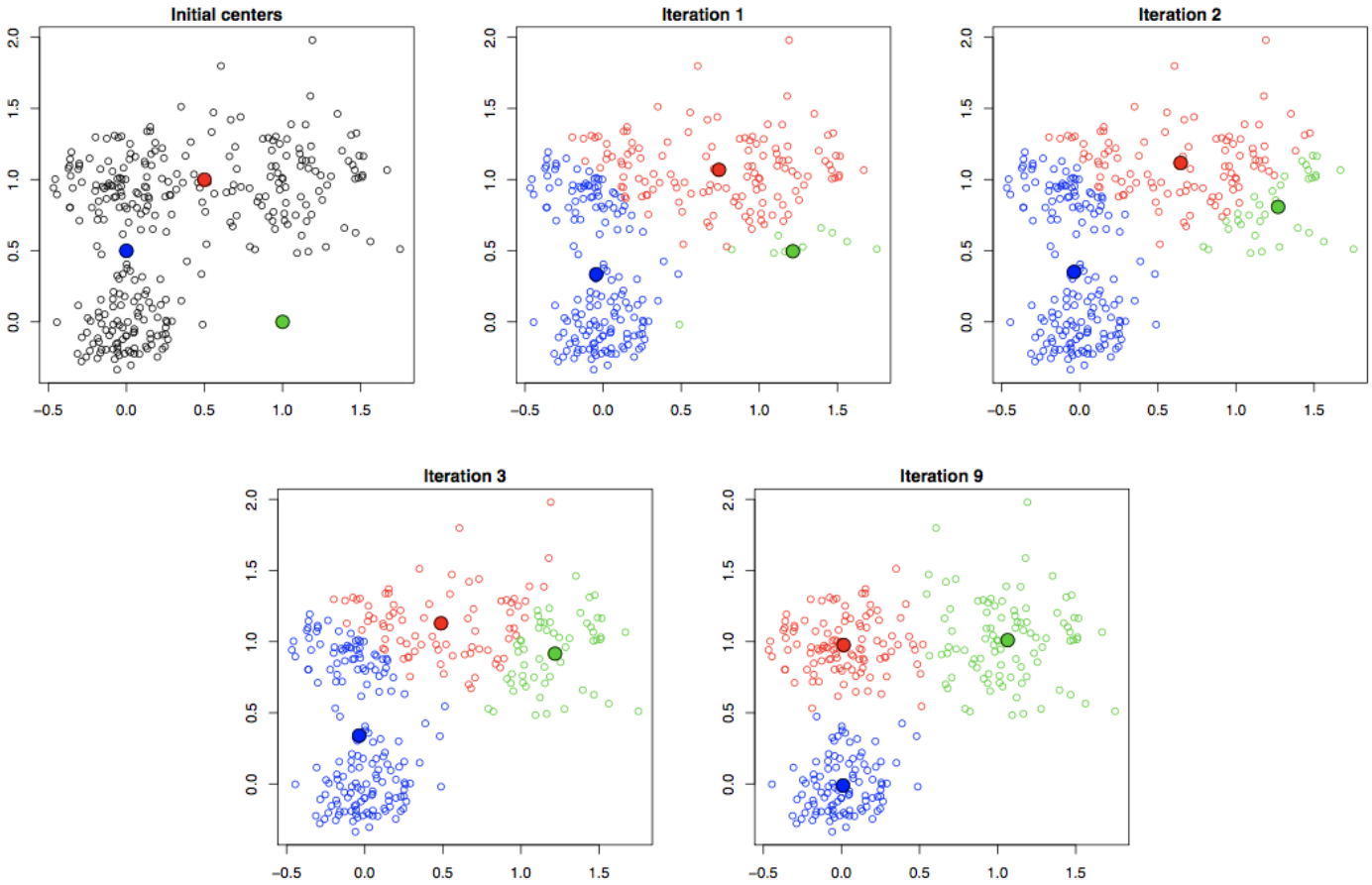
- Fix assignments for samples, update the centroids by a simple averaging

$$Z_j = \frac{\sum_{C(i)=j} X_i}{n_j}$$

  where n_j is the number of samples assigned to cluster j
- Alternate these two steps until converge to a local minimal (the algorithm will converge since the objective function is non-increasing)
- Solution depends on initialization

K-means example

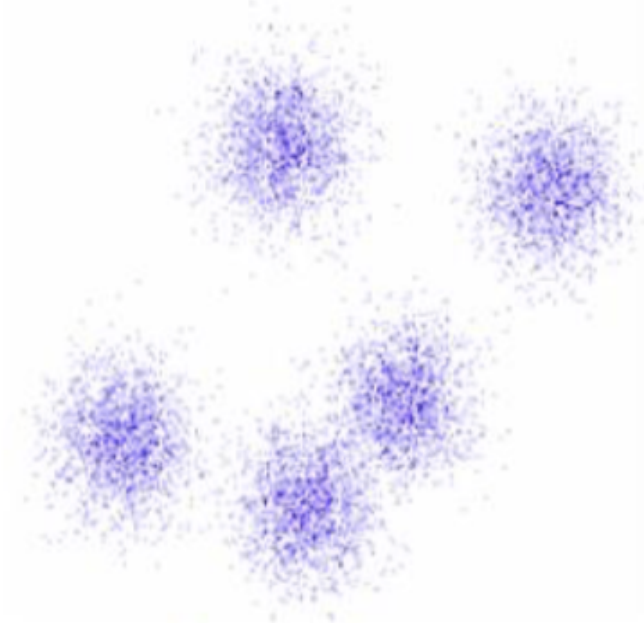
Here $X_i \in \mathbb{R}^2$, $n = 300$, and $K = 3$

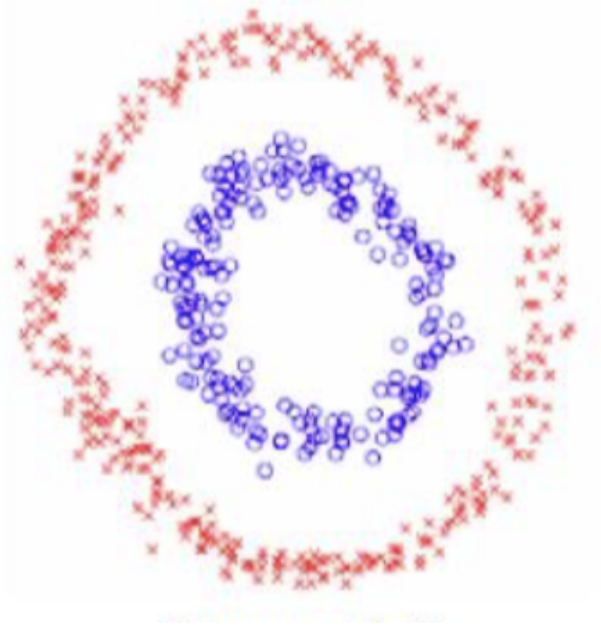


courtesy: Ryan Tibshirani.

Graph clustering

- Two different criteria
  - Compactness, e.g., K-means
  - Connectivity: spectral clustering





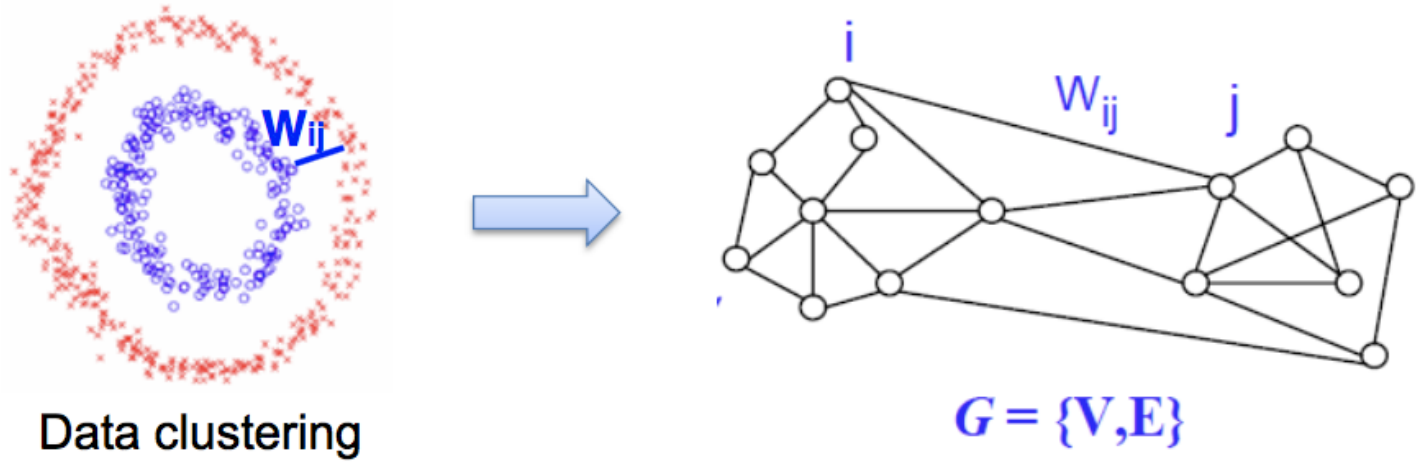
courtesy: Aarti Singh.

Graph clustering

- Given data $X_1, \ldots, X_n$ and similarity $w(X_i, X_i)$, partition the data into groups so that points in a group are similar and points in different groups are dissimilar
- For example, Gaussian kernel to define similarity

$$W_{ij} = e^{\frac{\|X_i - X_j\|^2}{2\sigma^2}}$$

σ^2 controls the size of the neighborhood



Graph partition

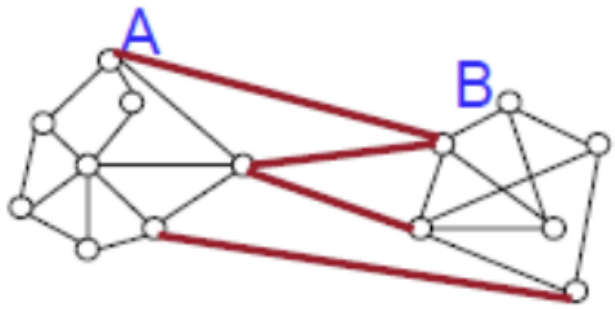
- Partition graph into sets (e.g., two sets)
- minimizing the total weights of the edges cut by the partitioning

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

- Consider the normalized cut

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

$d_i = \sum_{j=1}^{n} W_{ij}$
$\text{vol}(A) = \sum_{i \in A} d_i$: measures the size of A by the weights of its edges



Graph Laplacian

- Define diagonal matrix $D = \text{diag}\{d_1, \ldots, d_n\}$
- Similarity matrix W
- The unnormalized graph Laplacian matrix is defined by

$$L = D - W$$

- Properties
  - $\forall f \in \mathbb{R}^n$

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} W_{ij}(f_i - f_j)^2$$

  - L is symmetric and positive semi-definite
  - the smallest eigenvalue of L is o and the eigenvector is the all-one vector
  - L has n non-negative, real-valued eigenvalues

Normalized cut and graph Laplacian

- Finding a partition to minimize the normalized cut

$$\min \mathsf{Ncut}(A, B) = \min \frac{f^T L f}{f^T D f}$$

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

- Relaxation

$$\min \frac{f^T L f}{f^T D f} \quad \text{s.t.} \quad f^T D 1 = 0$$

- Solution: f: second eigenvector of generalized eigenvalue problem

$$Lf = \lambda D f$$

- Obtain cluster assignment by thresholding f at 0