Spectral Clustering

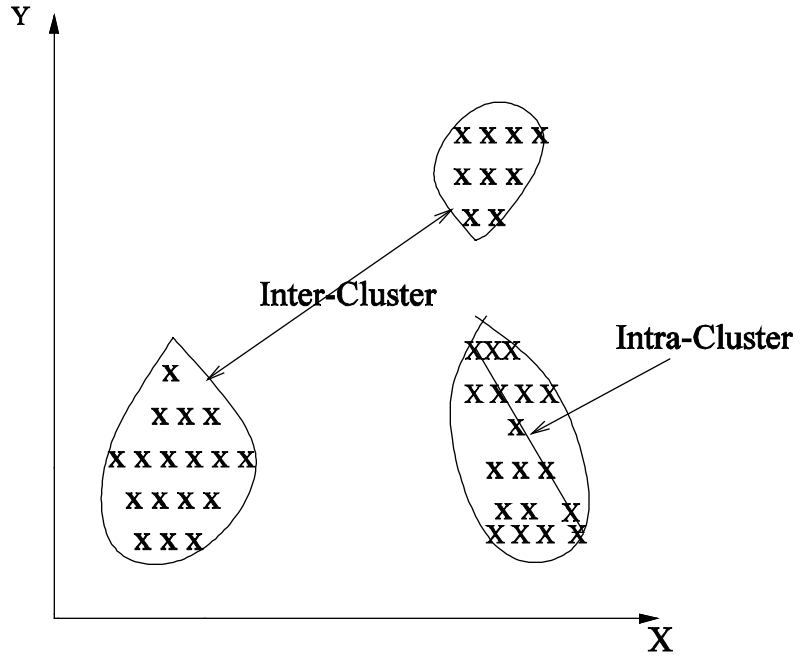
M Narasimha Murty
Professor, Dept. of CSA
Indian Institute of Science, Bengaluru

mnm@csa.iisc.ernet.in
August-December 2017

Piazza Page: https://piazza.com/iisc.ernet.in/fall2017/e0219

What is Clustering?

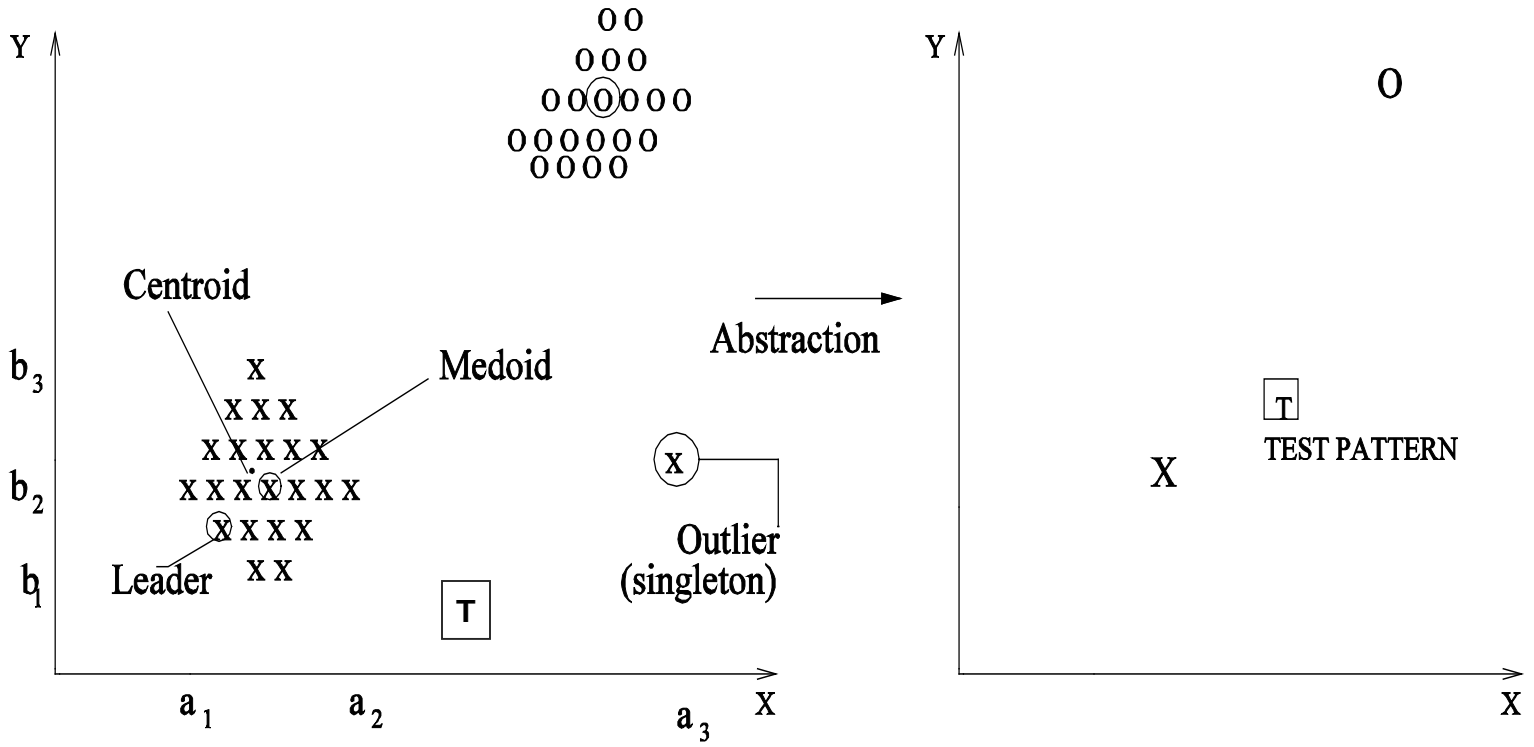
Clustering is partitioning of the dataset using inter-pattern proximity values



- Find clusters so that the objects of each cluster are similar to each other where as objects of different clusters are dissimilar.

- Such a partitioning helps in exploring the categorical structure in the data.

Clustering is Data Compression

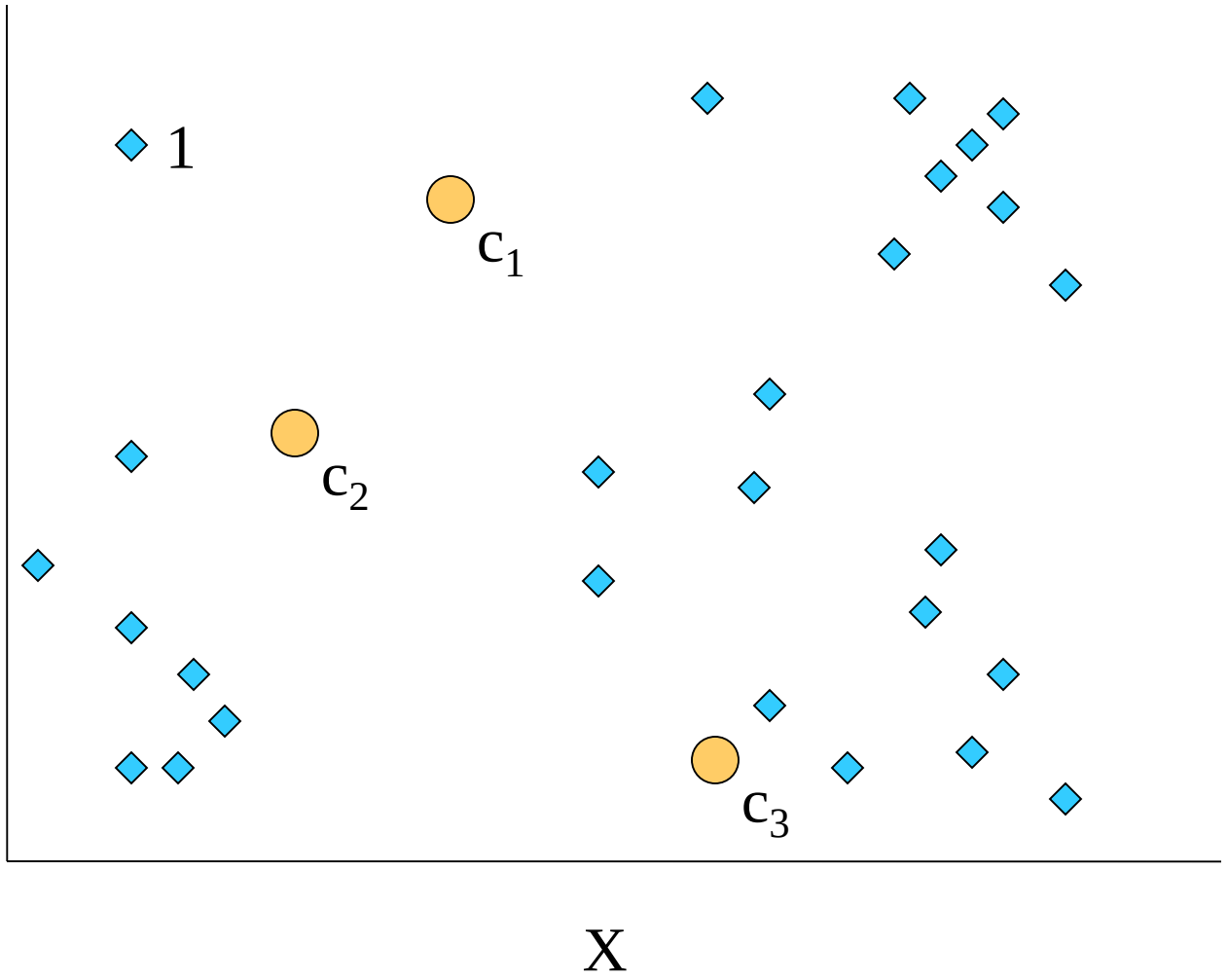


- It helps in efficient classification and in other decision making tasks.
- It is used in designing efficient classifiers (Support Vector Machine, Nearest Neighbor, Neural Net, and other classifiers).

K-means Clustering Algorithm

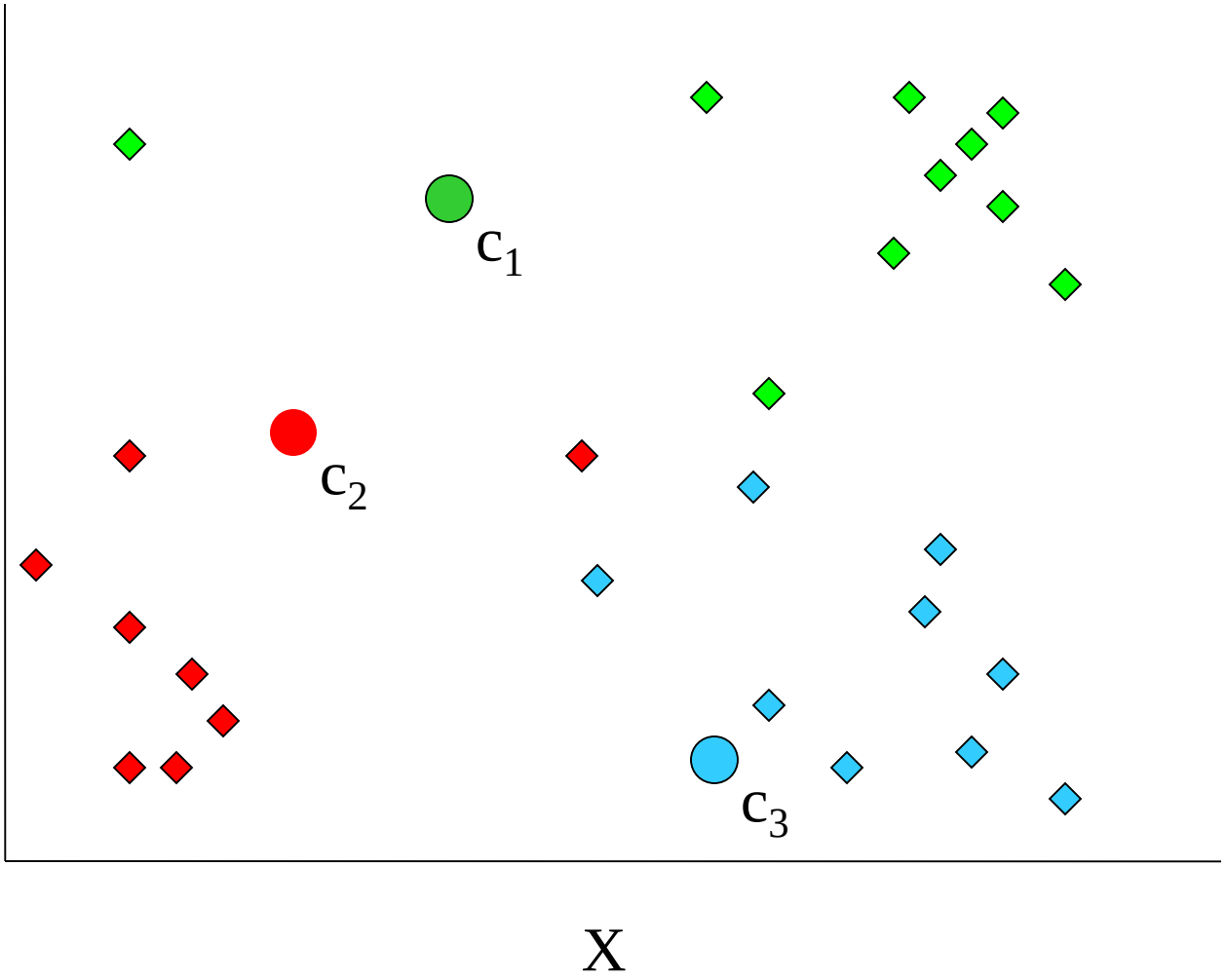
(Anderberg, *Cluster Analysis for Applications, Academic Press, 1973)*

Pick 3 initial cluster centers (randomly)



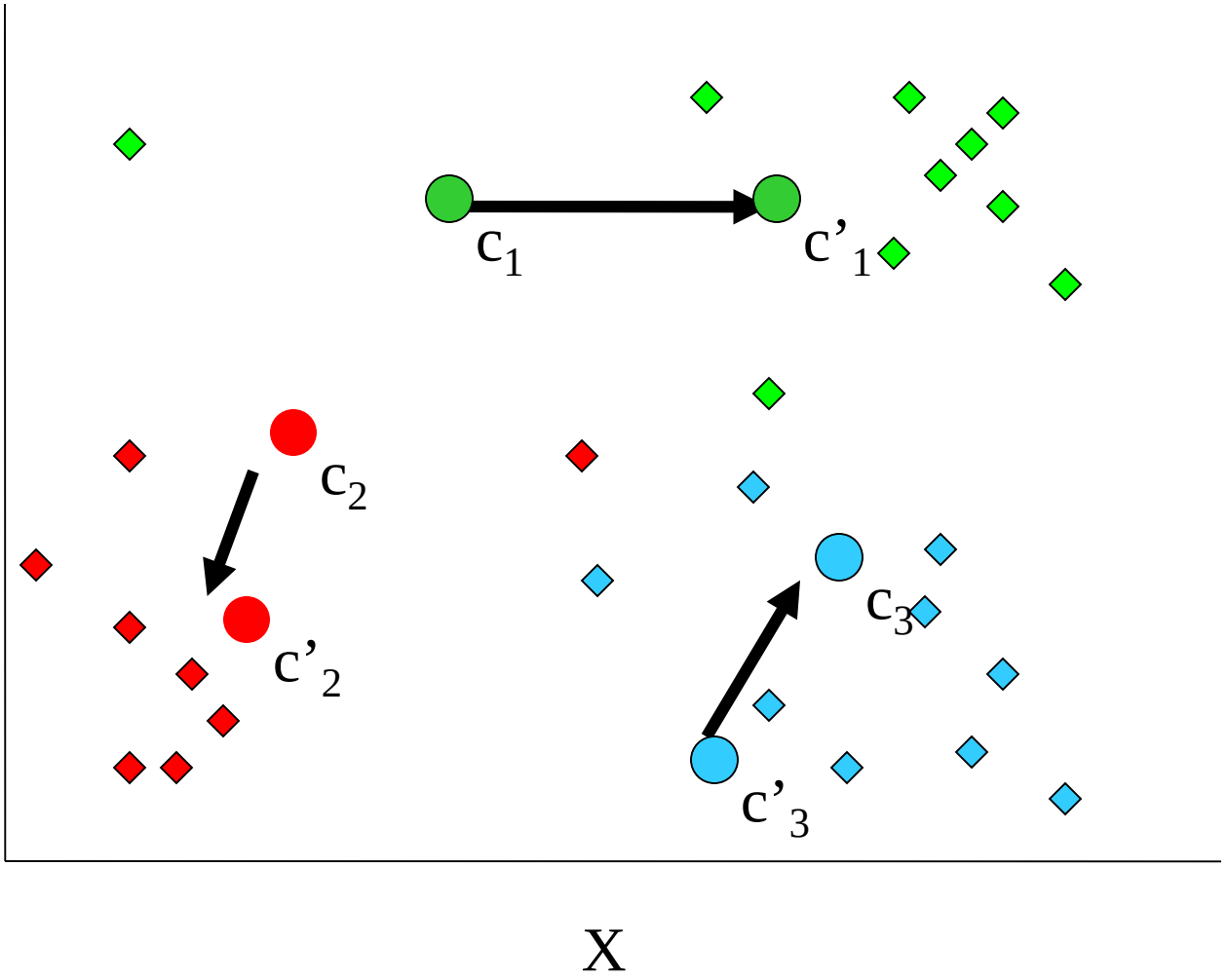
K-means example, step 2

Assign each point to the closest cluster center



K-means example, step 3

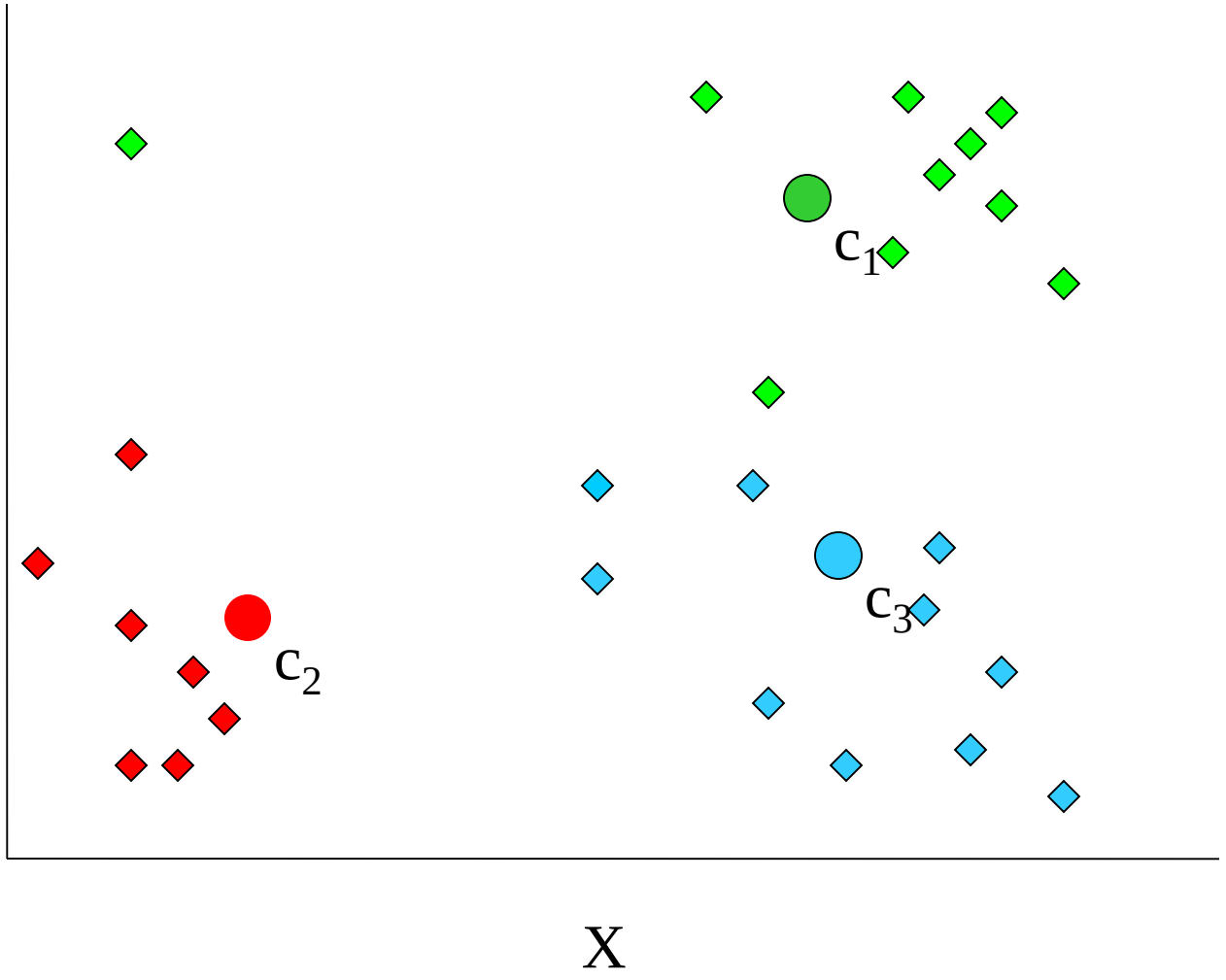
Move
each cluster
center
to the mean
of each cluster



K-means example, step 4a

Reassign points closest to a different new cluster center

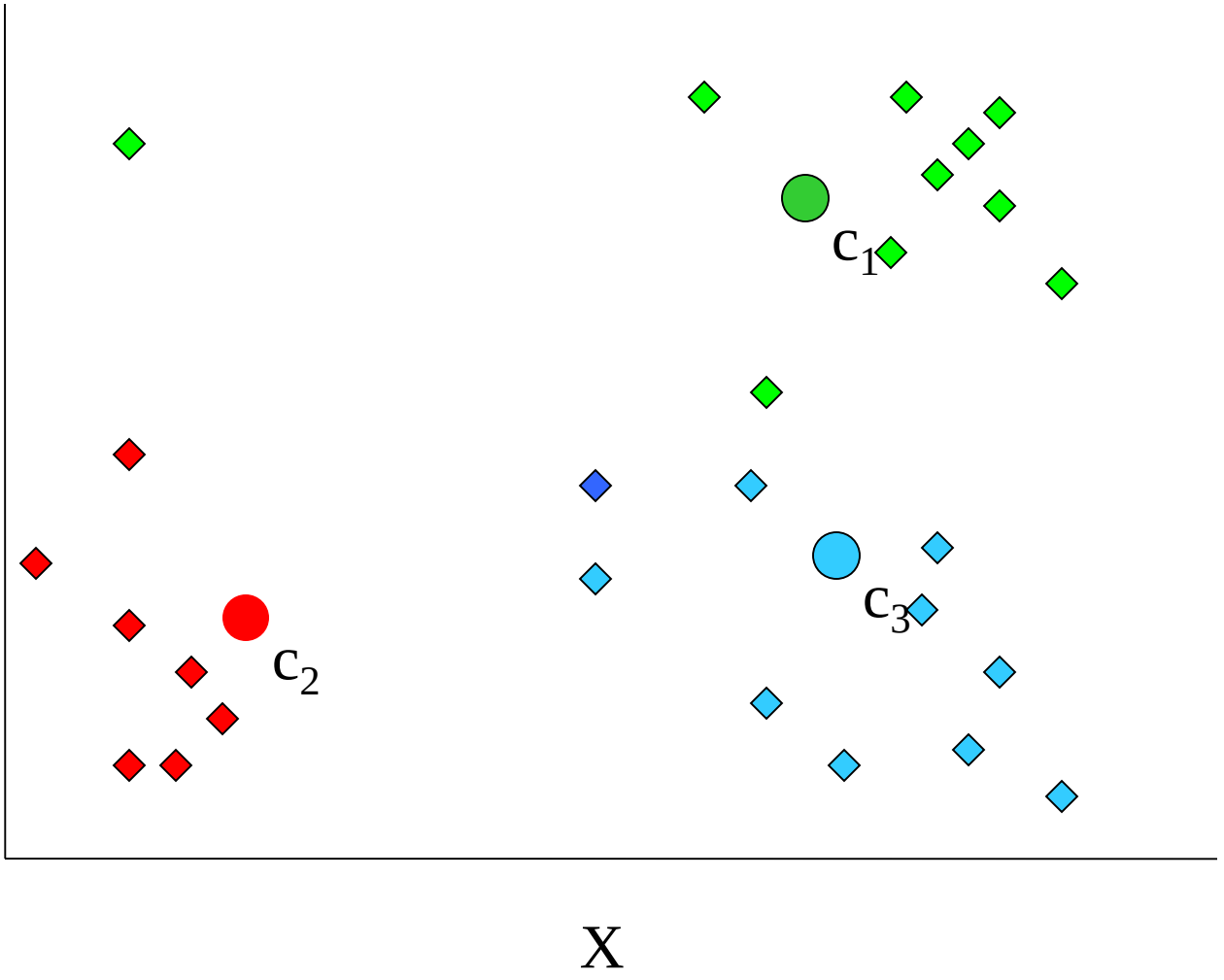
Q: Which points are reassigned?



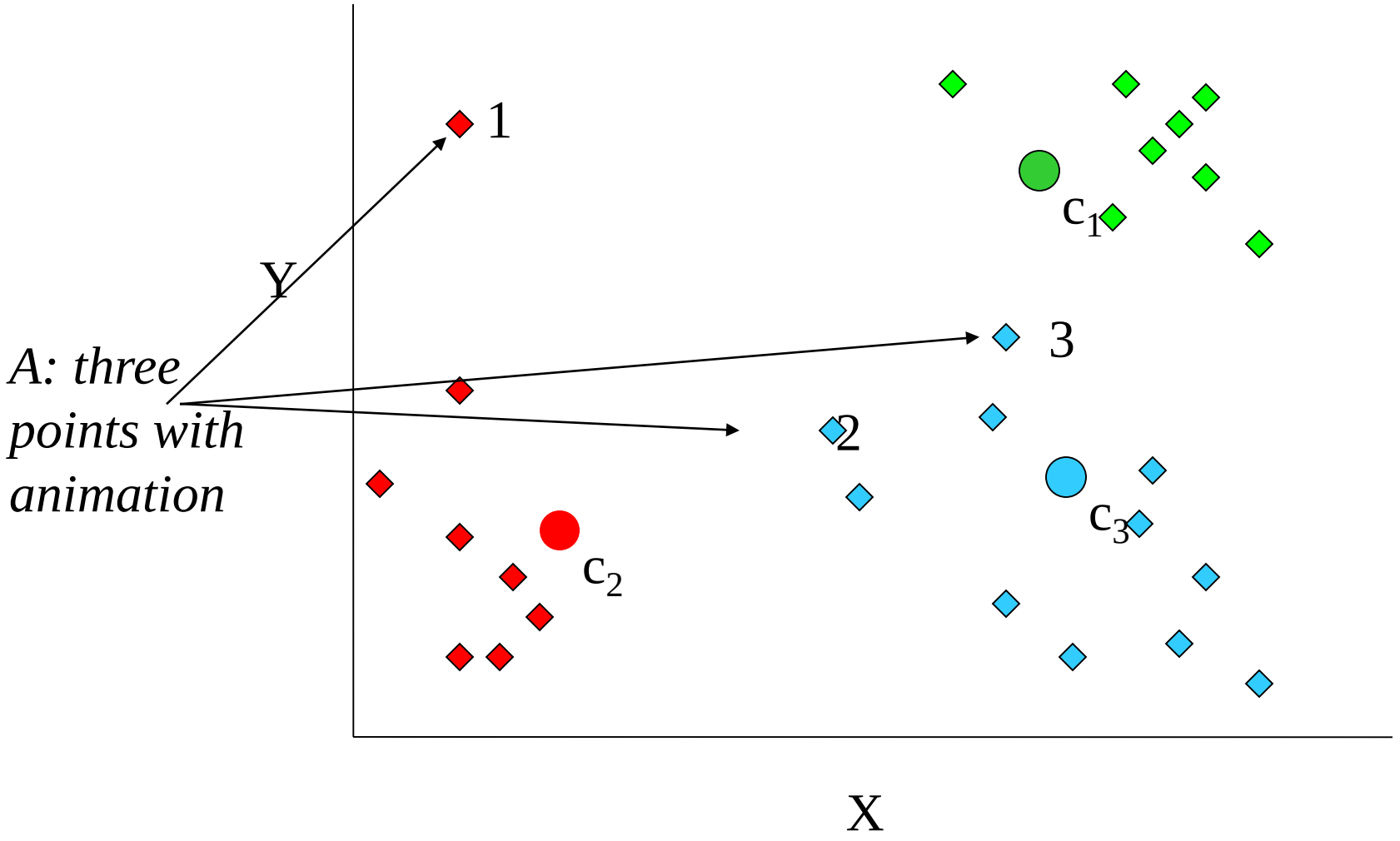
K-means example, step 4b

Reassign points closest to a different new cluster center

Q: Which points are reassigned?

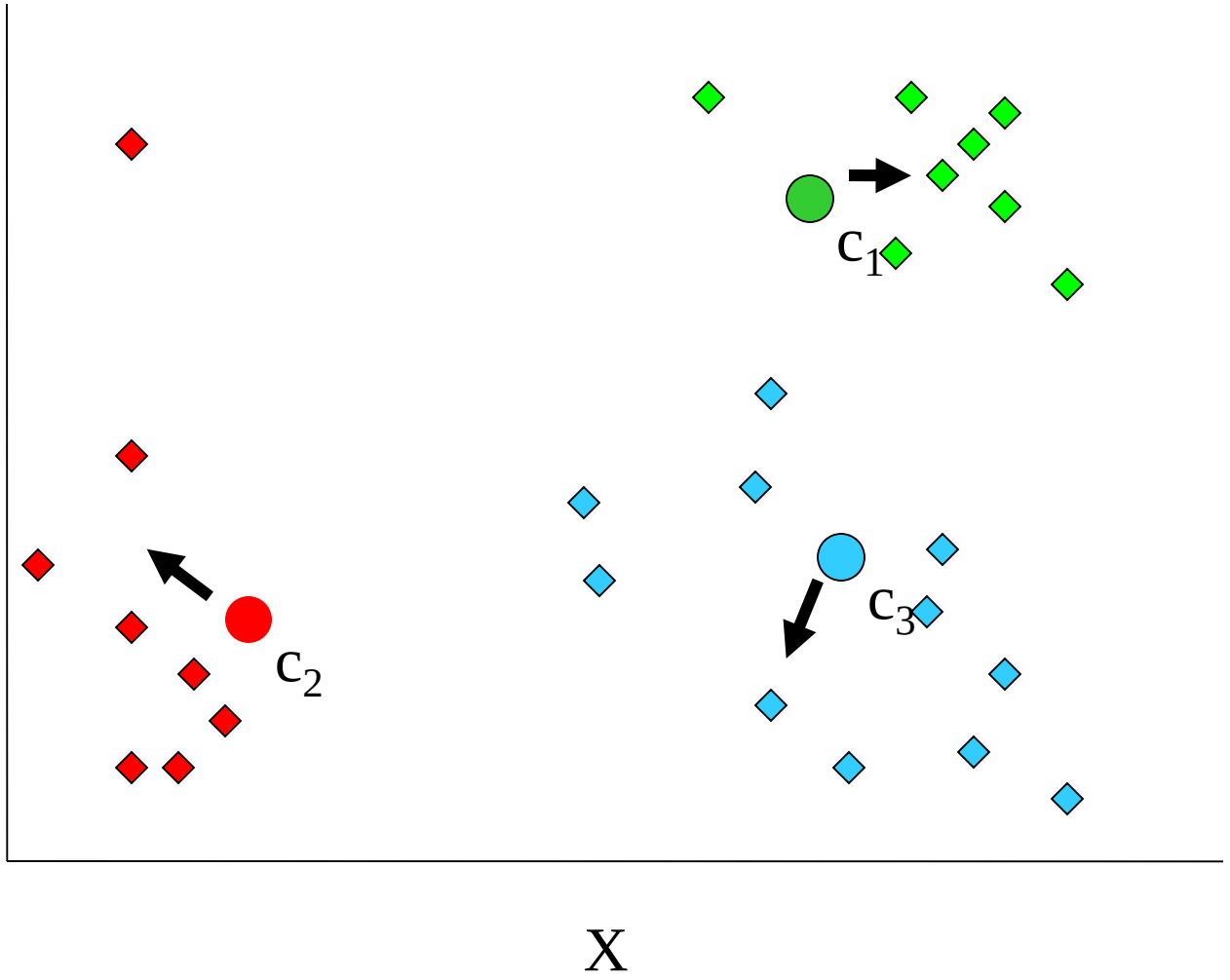


K-means example, step 4c



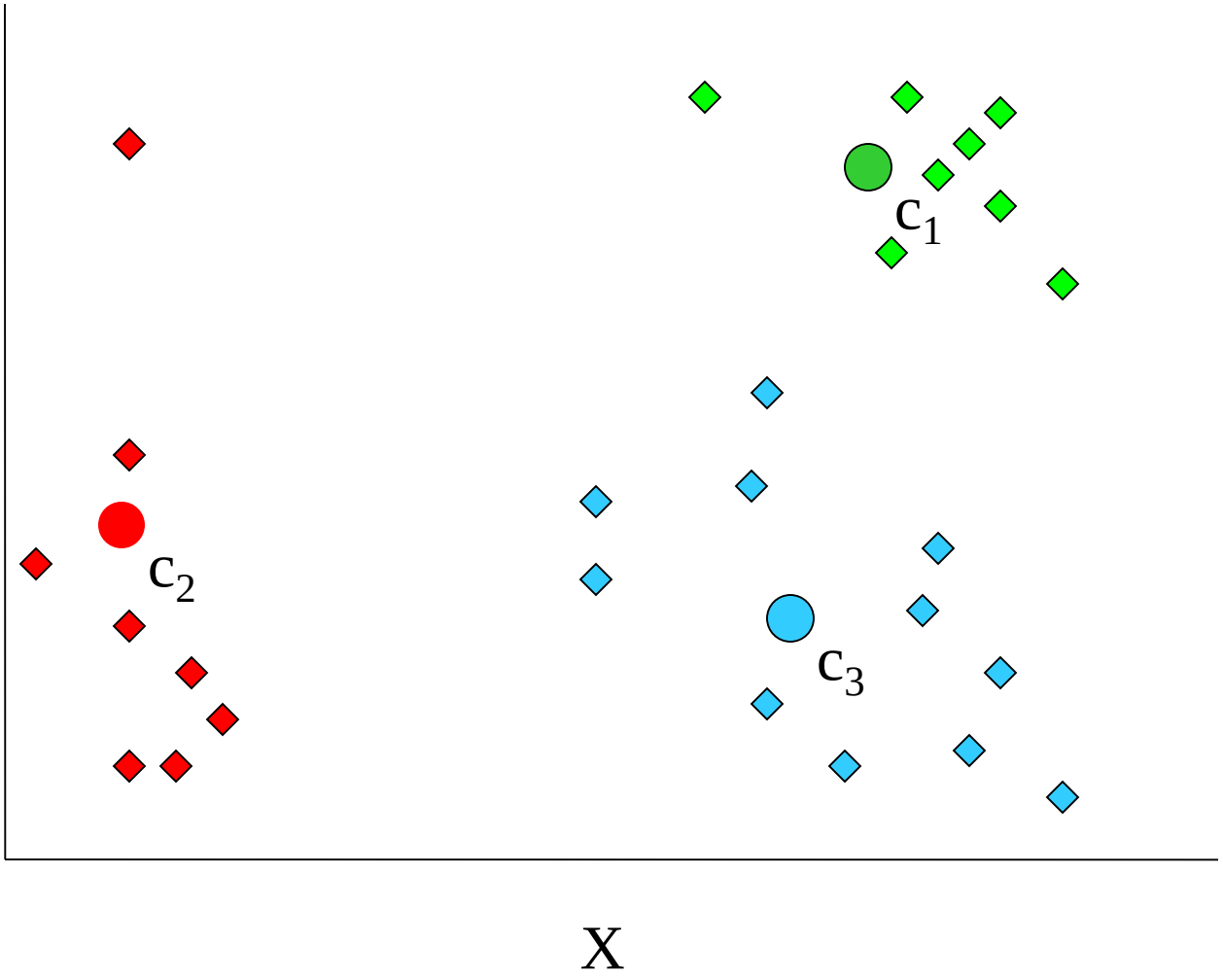
K-means example, step 4d

re-compute cluster means

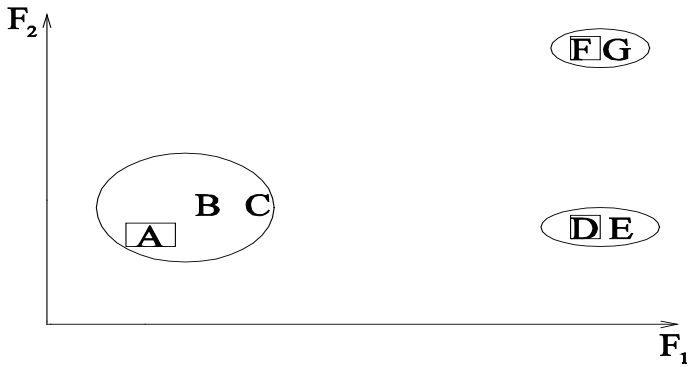


K-means example, step 5

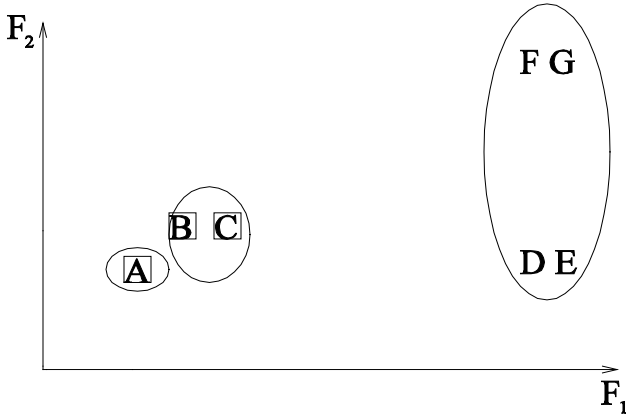
move cluster centers to cluster means



Effect of the Initial Partition



Optimal 3-partition



Nonoptimal 3-partition

A good heuristic is to select the initial K centers so that they are as far away from each other as possible.

Clustering: Example

- We consider some variants of an important algorithm and its behavior using the dataset shown in Table 1.

Pattern Number	feature1	feature2	feature3
1	10	3.5	2.0
2	63	5.4	1.3
3	10.4	3.5	2.1
4	10.3	3.3	2.0
5	73.5	5.8	1.2
6	81	6.1	1.3
7	10.4	3.3	2.3
8	71	6.4	1.0
9	10.4	3.5	2.3
10	10.5	3.3	2.1

Table: A Dataset of 10 patterns

K-Means Algorithm

- K-means algorithm is the most popular partitional clustering algorithm.
- It generates a K-partition of the dataset and the clusters are represented by their respective centroids.
- The algorithm is given below:

K-Means Algorithm
Input: Dataset, $\mathcal{X}$; Number of Clusters, K
Output: A K-partition of $\mathcal{X}$, Π_K^n
1. Select K initial centroids corresponding to the K clusters.
2. Assign each of the n points in $\mathcal{X}$ to the cluster whose centroid is closest to the data point. Update the centroids of the clusters based on the current assignment of points to the clusters.
3. Stop if there is no change in the cluster assignments during two successive iterations. Otherwise goto 2.

Features of K-Means Algorithm

- **Optimization of Squared Error:**
- The basic idea behind K-Means algorithm is to minimize this criterion function.
- Formally, the function may be specified as

$$\sum_{i=1}^{K} \sum_{X \in C_i} || \, X - centroid_i \, ||^2$$

- Note that the squared error will be maximum when $K = 1$ and is minimum (zero) when $K = n$. So, we consider the minimization of the criterion function for a given K.
- The K-means algorithm does not guarantee global minimum value of the squared error criterion shown.
- Further, the squared error minimization corresponds to minimizing the variance of points in each cluster. So, naturally this algorithm has a tendency to generate spherical clusters.

Features of K-Means Algorithm

- **Selection of initial centroids**:
- Select *K* out of the *n* data points as the initial centroids. Various options are:

  Select the first *K* out of *n* data points as the initial centroids. Considering the first 3 ($K = 3$) patterns $(10, 3.5, 2.0)$, $(63, 5.4, 1.3)$, $(10.4, 3.5, 2.1)$ in Table as the centroids of 3 clusters respectively, the algorithm stops after two iterations. The 3 clusters obtained and their centroids respectively are:

  - *Cluster*$1 : \{(10, 3.5, 2.0)\}$
  - *Cluster*$2 : \{(63, 5.4, 1.3), (73.5, 5.8, 1.2), (81, 6.1, 1.3), (71, 6.4, 1.0)\}$
  - *Cluster*3 :

$\{(10.4, 3.5, 2.1), (10.3, 3.3, 2.0), (10.4, 3.3, 2.3), (10.4, 3.5, 2.3), (10.5, 3.3, 2.1)\}$

  - *ClusterCentroids* : $(10, 3.5, 2.0),\ (72.1, 5.9, 1.2),\ (10.4, 3.4, 2.2)$

Features of K-Means Algorithm

- Select K out of n points as initial centroids such that the K points selected are as far away from each other as possible.
  1. Select the most dissimilar points in $\mathcal{X}$ as two centroids. Let them be X^1 and X^2. Set $q = 2$.
  2. If $q = K$ stop. Otherwise select X^{q+1}, the $q + 1^{th}$ centroid from the remaining $n - q$ points, where
     $$X^{q+1} = \underset{X}{argmax} \; (d(X^1, X) + \cdots + d(X^q, X)) \; X \in \mathcal{X} - \{X^1, X^2, \cdots, X^q\}.$$
  3. In the dataset shown in the Table the two extreme points are $(10, 3.5, 2.0)$ and $(81, 6.1, 1.3)$; these are selected as the first two centroids.
  4. The third centroid is $(63, 5.4, 1.3)$ as it is away from the already selected centroids significantly. Using these three initial centroids, we get the 3 clusters and their respective centroids, in two iterations, as:
     - *Cluster*1 : $\{(10, 3.5, 2.0), (10.4, 3.5, 2.1), (10.3, 3.3, 2.0), (10.4, 3.3, 2.3), (10.4, 3.5, 2.3), (10.5, 3.3, 2.1)\}$
     - *Cluster*2 : $\{(73.5, 5.8, 1.2), (81, 6.1, 1.3)\}$
     - *Cluster*3 : $\{(63, 5.4, 1.3), (71, 6.4, 1.0)\}$
     - *ClusterCentroids* : $(10.3, 3.4, 2.1), \; (77.3, 6, 1.2), \; (67, 5.9, 1.2)$

Spectral Clustering

- *K*-means produces good clusters when the data has isotropic or spherical clusters.
- *K*-means algorithm is not suited when the clusters are non-isotropic; specifically when the clusters are chain-like (elongated in a direction) or concentric (where the clusters have roughly the same centroid.
- Spectral clustering algorithms are well suited to deal with such data sets.
- Spectral clustering algorithms work on data set represented in the form of a graph.
- Here a graph is viewed as a triple $< V, E, S >$ where S is the matrix of similarity values between pairs of nodes in the graph. Here the sets V, E, and S are:
  - $V = \{X_1, X_2, \cdots, X_n\}$. That is each node/vertex in V corresponds to a data point in the collection.
  - $E = \{< X_i, X_j >: X_i \in V, \text{ and } X_j \in V\}$ for $i, j = 1, 2, \cdots, n$. So, each element of E characterizes an edge between a pair of vertices.

Spectral Clustering

- $S = \{s_{ij} : X_i, X_j \in V\}$. Each element of S characterizes similarity between a pair of nodes. $s_{ij} = 0$ if X_i and X_j are not similar (or not connected); and $s_{ij} = 1$ if X_i and X_j are similar (or connected).
- In our treatment the graph is undirected; so $s_{ij} = s_{ji}$. However, there could be applications where the graph is directed.
- Further, we are assuming that the similarity values are binary, either 0 or 1; in general these could be nonegative real numbers.
- *Weightmatrix*, W : is a diagonal matrix and $W_{ii} = \sum_{j \in V} s_{ij}$ *and* $W_{ij} = 0$ *if* $i \neq j$.
- That is i^{th} diagonal element in W is the sum of the elements in the i^{th} row of S.
- This could be called the *degree matrix* when s_{ij}'s are binary as the entry W_{ii} corresponds to the degree of node X_i.

Spectral Clustering: Example

- We illustrate these ideas using the graph shown in the Figure below.

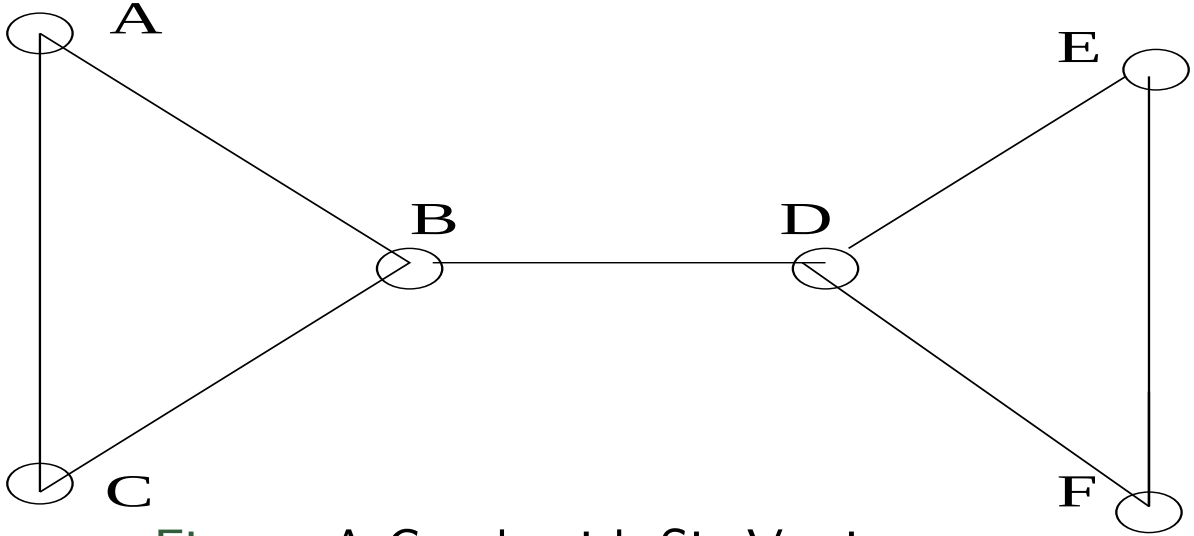


Figure: A Graph with Six Vertices

- The corresponding S matrix is given by

$$S = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Spectral Clustering: Example

- Here we are assuming that a node is similar to itself and so the diagonal entries are all 1.
- The weight matrix W (or degree matrix in this case) is

$$W = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

- Let C_1 be a subset of V and C_2, the complement of C_1 be $V - C_1$. Based on this notation we can generate a two-partition of V using the notion of *mincut*.
-

$$cut(C_1, C_2) = \sum_{x_i \in C_1 x_j \in C_2} s_{ij}$$

Spectral Clustering: Example

- Then *mincut* is defined as

$$mincut(C_1^*, C_2^*) \;=\; minimum_{C_1,C_2} \;\; cut(C_1, C_2)$$

- where C_1^* *and* $C_2^* (= V - C_1^*)$ are the members of the optimal partition corresponding to choices of C_1 and C_2.
- Such a C_1^* and its complement C_2^* correspond to the two required clusters of the partition.
- It is possible to abstract the mincut expression in a form suitable for optimization by considering the following.
  - Let C_1 *and* C_2 be the two possible clusters being considered. Let these two clusters be viewed as negative (C_1) and positive (C_2) clusters.
  - Based on this one can abstract the index vector I of size n where there are n vertices in the graph.
  - Let I_i be -1 if $X_i \in C_1$ and $+1$ if $X_i \in C_2$ for $i = 1, 2, \cdots, n$ (X_i is the i^{th} node).
  - Note that $(I_i - I_j)$ is 0 if both X_i *and* X_j are either in C_1 or in C_2.
  - Further, $\frac{1}{4}(I_i - I_j)^2$ is 1 if X_i belongs to one cluster and X_j is in the other cluster.

Spectral Clustering: Example

- Note that $Cut(C_1, C_2)$ considers addition of similarities s_{ij} where $X_i \in C_1$ and $X_j \in C_2$.
- We can select such s_{ij}'s by considering $\frac{1}{4}s_{ij}(I_i - I_j)^2$ in the place of s_{ij} in the summation.
- So, $Cut(C_1, C_2)$ can be equivalently written as

$$Cut(C_1, C_2) = \frac{1}{4} \sum_{X_i \in C_1,\, X_j \in C_2} s_{ij}(I_i - I_j)^2$$
$$= \frac{1}{8} \sum_{I_i \neq I_j} s_{ij}(I_i - I_j)^2$$

- It is possible to simplify this equation to show that

$$Cut(C_1, C_2) = \frac{1}{4}(I^t W I - I^t S I) = \frac{1}{4} I^t D I$$

  Where $D = W - S$.
- So, minimizing the Cut amounts to finding the index vector I such that $I^t D I$ is minimized.

Spectral Clustering: Example

- Note that $I^t DI = I^t WI - I^t SI$

$$= \sum_{i=1}^{n} w_{ii} l_i^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} l_i l_j s_{ij}$$

$$= \frac{1}{2} \left[\sum_{i=1}^{n} w_{ii} l_i^2 - 2 \sum_{i=1}^{n} \sum_{j=1}^{n} l_i l_j s_{ij} + \sum_{j=1}^{n} w_{jj} l_j^2 \right]$$

$$= \frac{1}{2} \left[\sum_{i} \sum_{j} s_{ij} l_i^2 - 2 \sum_{i} \sum_{j} l_i l_j s_{ij} + \sum_{i} \sum_{j} s_{ij} l_j^2 \right]$$

$$= \frac{1}{2} \left[\sum_{i} \sum_{j} (l_i^2 - 2 l_i l_j + l_j^2) s_{ij} \right] = \frac{1}{2} \sum_{i} \sum_{j} s_{ij} (l_i - l_j)^2$$

Spectral Clustering: Example

- Once I is known it is possible to obtain the clusters based on the polarity of the entries in I.
- So the problem of obtaining the *mincut* amounts to

$$min_I\ I^t DI \text{ such that } I_i \in \{-1, 1\} \text{ for all } i \in \{1, 2, \cdots, n\}$$

- Because this is a combinatorially difficult problem to solve, we relax the selection of elements in I to real numbers which leads to

$$min_I\ I^t DI \text{ such that } I^t I = n$$

- It is possible to see that D is symmetric as S and W are symmetric.
- The smallest eigenvalue of D is 0 and the corresponding eigenvector is $\mathbf{1} = (1, 1, \cdots, 1)^t$ because $D\mathbf{1} = 0 = 0\mathbf{1}$.
- By choosing the value of I as the eigenvector $\mathbf{1}$, it is possible to show that $I^t DI$ is equal to 0 as $DI = D\mathbf{1} = 0$.
- However, this value of I does not generate a 2-partition as there is only a positive cluster.
- So, instead of the smallest eigenvalue, consider the next smallest.

Spectral Clustering: Example

- So, $I^t DI$ is still small where I is the eigenvector corresponding to the second smallest eigenvalue.
- Further, *because D is symmetric, eigenvectors of D are orthogonal and the eigenvalues are all real.*
- So, by choosing I to be the eigenvector corresponding to the second smallest eigenvalue, we get an I that is orthogonal to $\mathbf{1}$.
- This means that there will be both negative and positive entries in I.
- So, I is the eigenvector corresponding to the second smallest.
- We illustrate this algorithm using the example shown in the Figure. The matrix $D = W - S$ is given by

$$D = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

Spectral Clustering: Example

- The eigenvalues of D are $0, \frac{5-\sqrt{17}}{2}, 3, 3, 3, \frac{5+\sqrt{17}}{2}$. The first two eigenvectors are $\mathbf{1}$ and $(1, \frac{-3+\sqrt{17}}{2}, 1, \frac{3-\sqrt{17}}{2}, \frac{-7+\sqrt{17}}{4}, \frac{3-\sqrt{17}}{2})^t$.
- Note that in the second eigenvector, the first three entries are positive and the remaining three are negative.
- So, the clusters are $C_1 = \{D, E, F\}$ *and* $C_2 = \{A, B, C\}$ where C_1 is the negative cluster and C_2 is the positive cluster.
- Also note that this clustering is intuitively appealing as points in each cluster are completely connected.
- In the example shown in the Figure we have considered the possibility of a two-partition.
- It is possible in general that the number of clusters K is greater than 2.
- In such a case we consider the K eigenvectors corresponding to the K smallest eigenvalues.
- Note that each eigenvector is n-dimensional.
- So, the K eigenvectors provide a K-dimensional representation of the n patterns by viewing the K eigenvectors as K columns in a matrix.

Spectral Clustering: Example

- This matrix will be of size $n \times K$. Also these K eigenvectors are orthogonal to each other.
- So, we can cluster the n rows (data points) into K clusters.
- By considering the first two eigenvectors as two columns in a matrix we get the n two-dimensional patterns shown below.
- By employing K-means algorithm on this data with a value of 2 for K will give us the same clusters as we got earlier.

$$
\begin{array}{ll}
(1, & 1) \\
(1, & \frac{-3+\sqrt{17}}{2}) \\
(1, & 1) \\
(1, & \frac{3-\sqrt{17}}{2}) \\
(1, & \frac{-7-\sqrt{17}}{2}) \\
(1, & \frac{3-\sqrt{17}}{2})
\end{array}
$$

Table: Two-Dimensional Representation of the Six Points

Spectral Clustering: Shi and Malik

- *Spectral clustering* gets its name from the word spectrum. The set of all eigenvalues of a matrix is called its *spectrum*.
- The magnitude of the maximum eigenvalue of the matrix is called the *spectral radius*.
- Here we have examined how clustering can be performed by using the eigenvalues and eigenvectors of the matrix D which is obtained from the weight matrix W and the similarity matrix S.
- It is possible to consider other variants of D to realize several other spectral clustering algorithms.
- For example, consider $L_{sym} = W^{-\frac{1}{2}}(W - S)W^{-\frac{1}{2}} = I - W^{-\frac{1}{2}}SW^{-\frac{1}{2}}$.

Spectral Clustering: Ng, Jordan, and Weiss (2002)

1. Compute the normalized Laplacian $L_{sym} = W^{-\frac{1}{2}}(W - S)W^{-\frac{1}{2}} = I - W^{-\frac{1}{2}}SW^{-\frac{1}{2}}$.
2. **Compute the first k eigenvectors $u_1, \cdots, u_k$.**
3. Let $U \in \Re^{n \times k}$ be the matrix containing the vectors $u_1, \cdots, u_k$ as columns.
4. **Form the matrix $T \in \Re^{n \times k}$ from U by normalizing the rows to norm** 1, that is set $t_{ij} = \frac{u_{ij}}{(\sum_k u_{ik}^2)^{\frac{1}{2}}}$.
5. For $i = 1, \cdots, n$, let $y_i \in \Re^k$ be the vector corresponding to the i^{th} row of T.
6. Cluster the y_i's using the k-means algorithm into clusters $C_1, \cdots, C_k$.

Output: Clusters $A_1, \cdots, A_k$ with $A_i = \{j : y_j \in C_i\}$.