CS 405/605 Data Science

Dr. Qianqian Tong

K-means Clustering

Clustering:

- What is clustering?
- Types of clustering
- Clustering algorithm ----- k-means
- How can you do this efficiently?

K-means Clustering

Clustering:

- What is clustering?

K-means Clustering

Clustering:

Task 1 : Group These Set of Document into 3 Groups based on meaning

Doc1 : Health , Medicine, Doctor
Doc 2 : Machine Learning, Computer
Doc 3 : Environment, Planet
Doc 4 : Pollution, Climate Crisis
Doc 5 : Covid, Health , Doctor

K-means Clustering

Clustering:

Task 1 : Group These Set of Document into 3 Groups.

Doc1 : Health , Medicine, Doctor
Doc 2 : Machine Learning, Computer
Doc 3 : Environment, Planet
Doc 4 : Pollution, Climate Crisis
Doc 5 : Covid, Health , Doctor

K-means Clustering

Clustering:

Task 1 : Group These Set of Document into 3 Groups.

Doc1 : Health , Medicine, Doctor

Doc 5 : Covid, Health , Doctor

Doc 3 : Environment, Planet

Doc 4 : Pollution, Climate Crisis

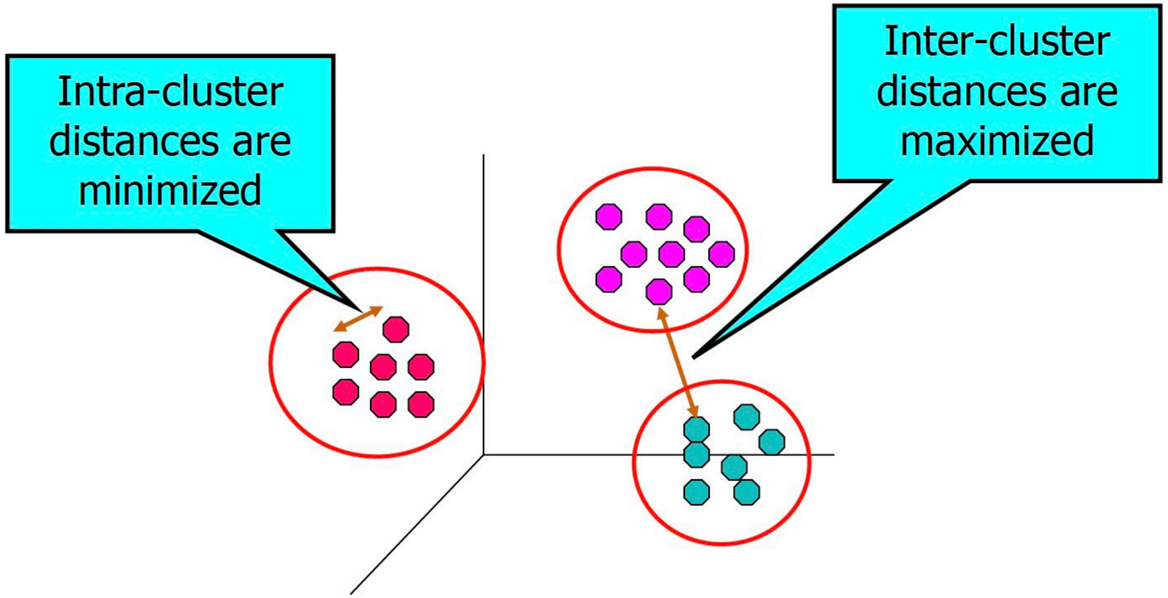
Doc 2 : Machine Learning, Computer

K-means Clustering

Clustering:

- What is clustering?

Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups.



K-means Clustering

Clustering:

- What is clustering?

Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups.

	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

K-means Clustering

Clustering:

- Types of clustering

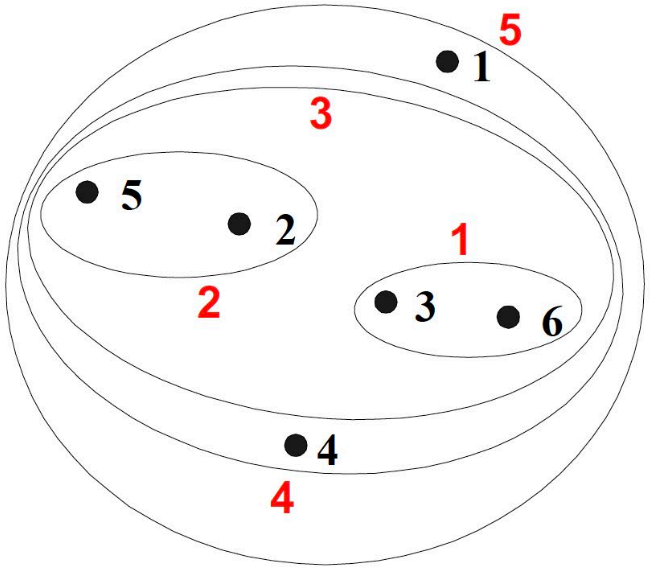
  - *A clustering is a set of clusters*
  - *Important distinction between hierarchical and partitional sets of clusters*
  - *Hierarchical clustering*
    - *A set of nested clusters organized as a hierarchical Tree*
  - *Partitional Clustering*
    - *A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset*

K-means Clustering

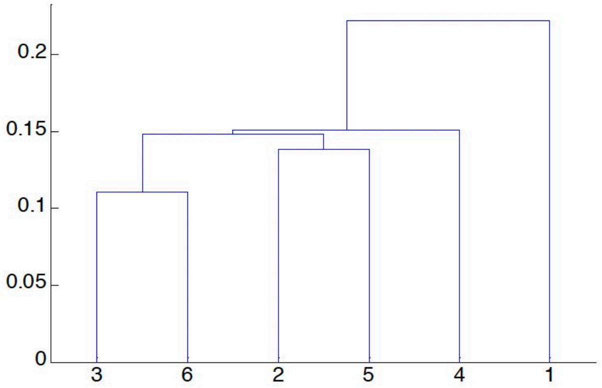
Clustering:

- Types of clustering

Hierarchical clustering



Nested Clusters



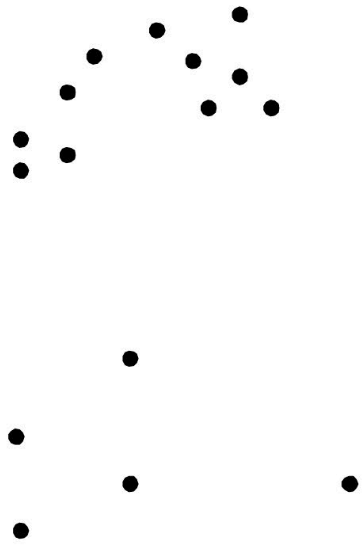
Dendrogram

K-means Clustering

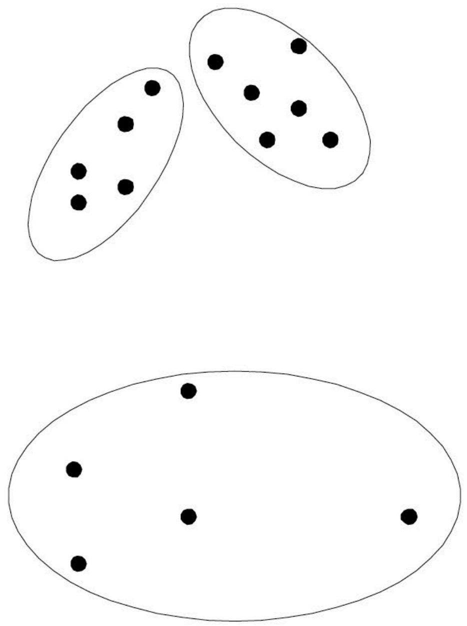
Clustering:

- Types of clustering

Partitional Clustering



Original Points



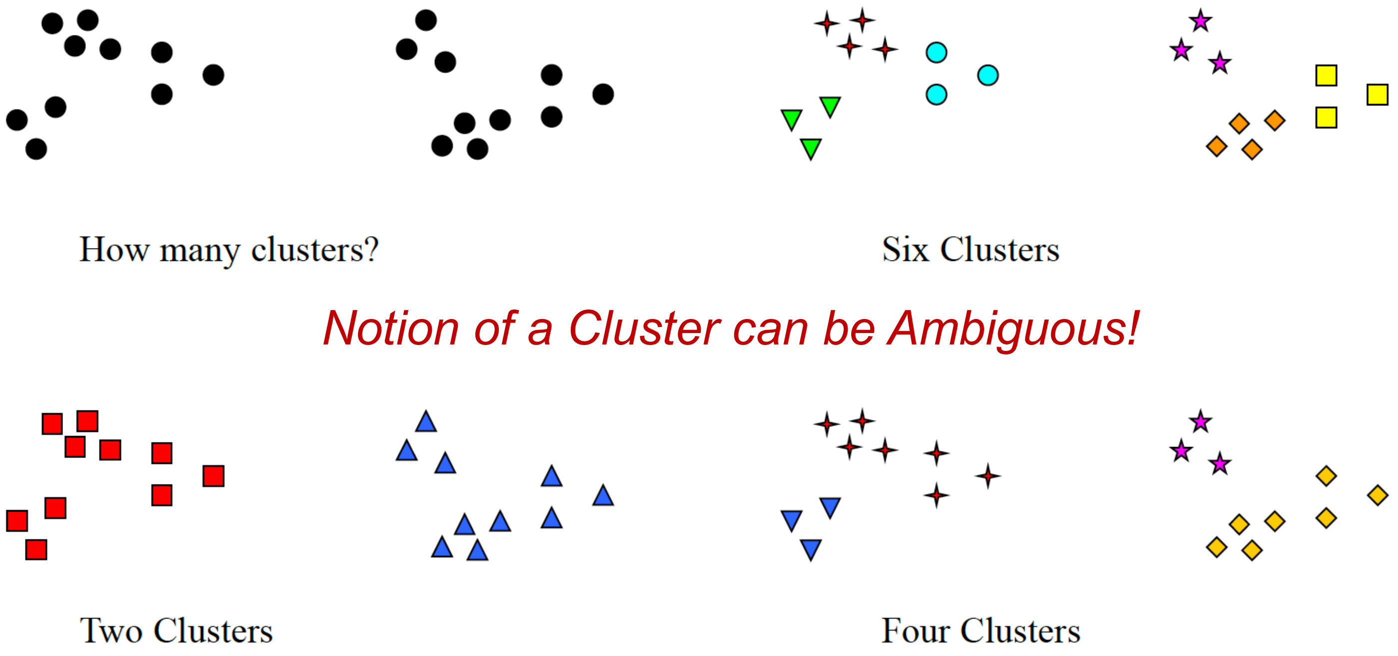
A Partitional Clustering

K-means Clustering

Clustering:

- Types of clustering

Partitional Clustering



K-means Clustering

Clustering:

- *Clustering Algorithms*

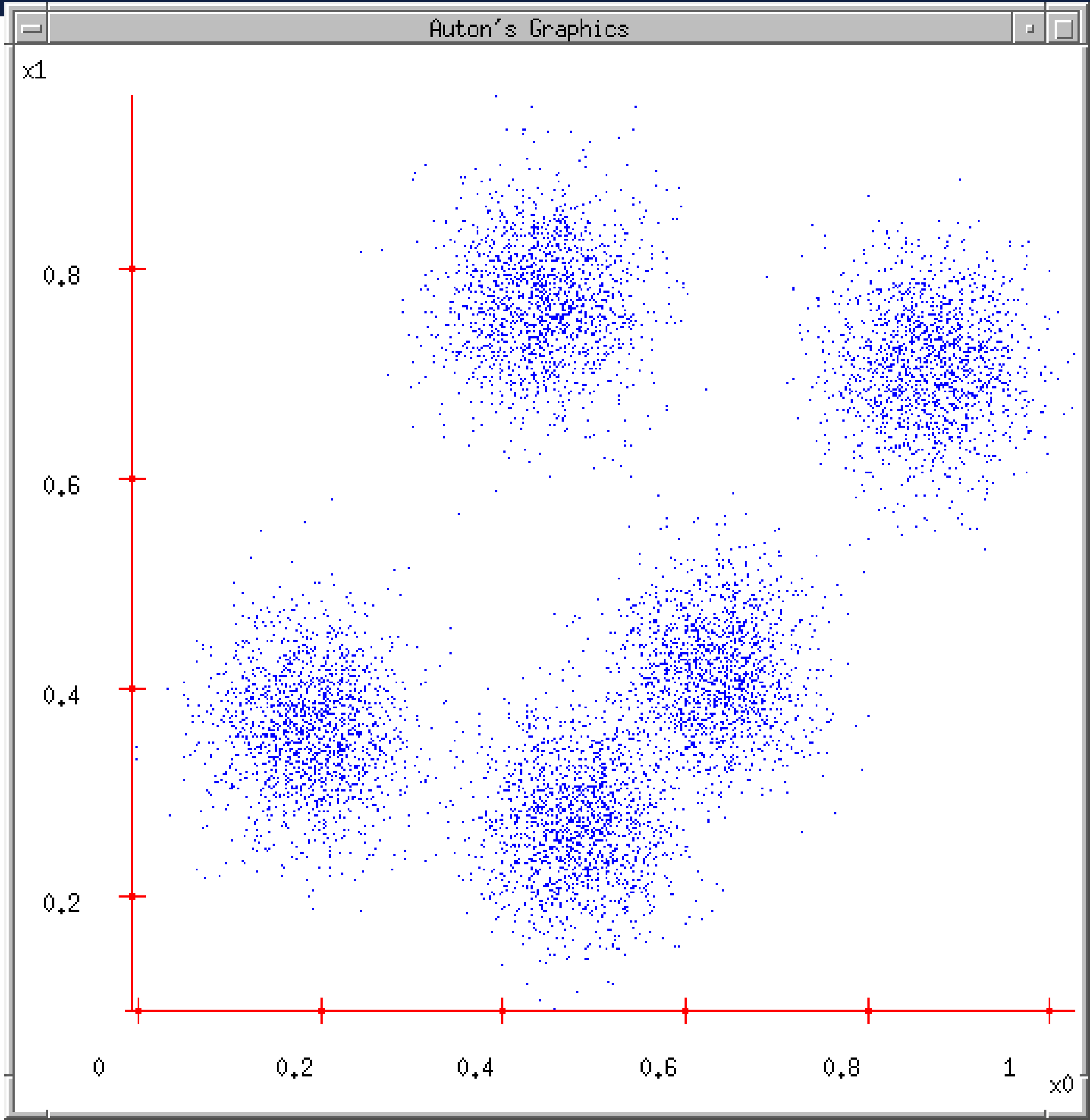
    - *K-means and its variants*
    - *Hierarchical clustering*
    - *Density-based clustering*
    - *Spectral clustering*

K-means Clustering

Clustering:

- *Clustering Algorithms*
  - *K-means*

1. Ask user how many clusters they'd like. *(e.g. k=5)*



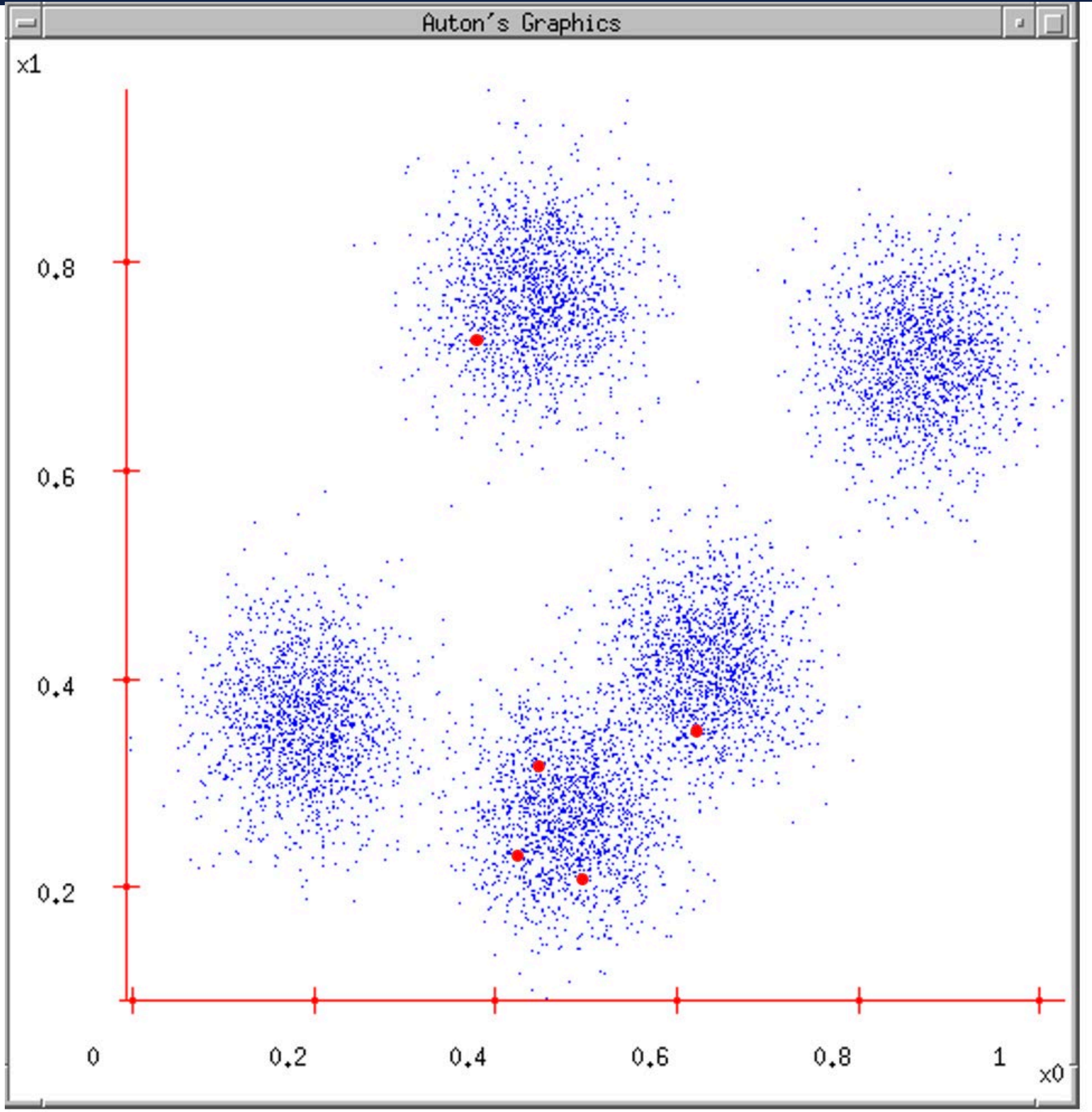
UNC GREENSBORO

K-means Clustering

Clustering:

- *Clustering Algorithms*
  - *K-means*

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



K-means Clustering

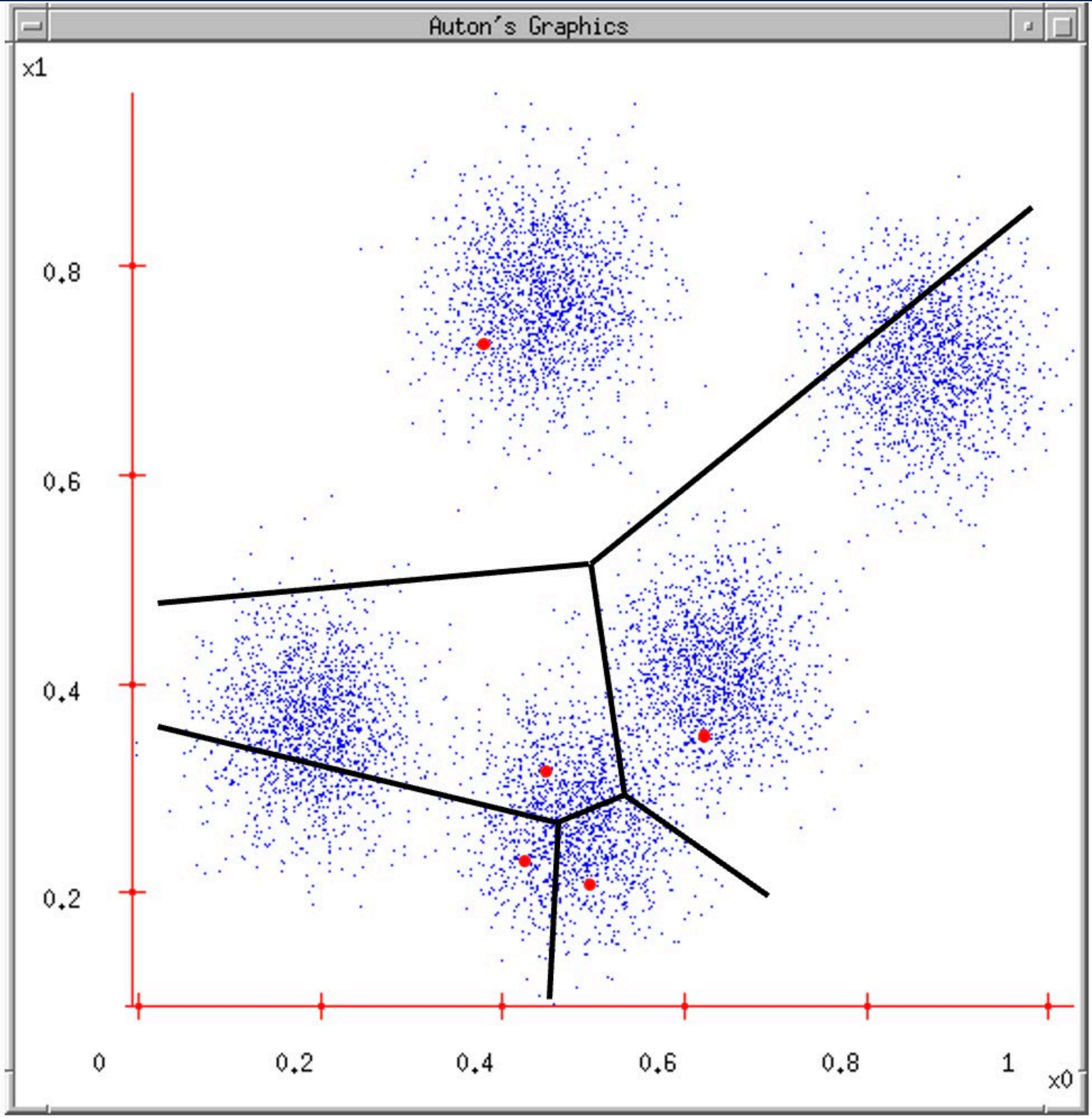
Clustering:

- *Clustering Algorithms*

  - *K-means*

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.

   (Thus each Center "owns" a set of datapoints)



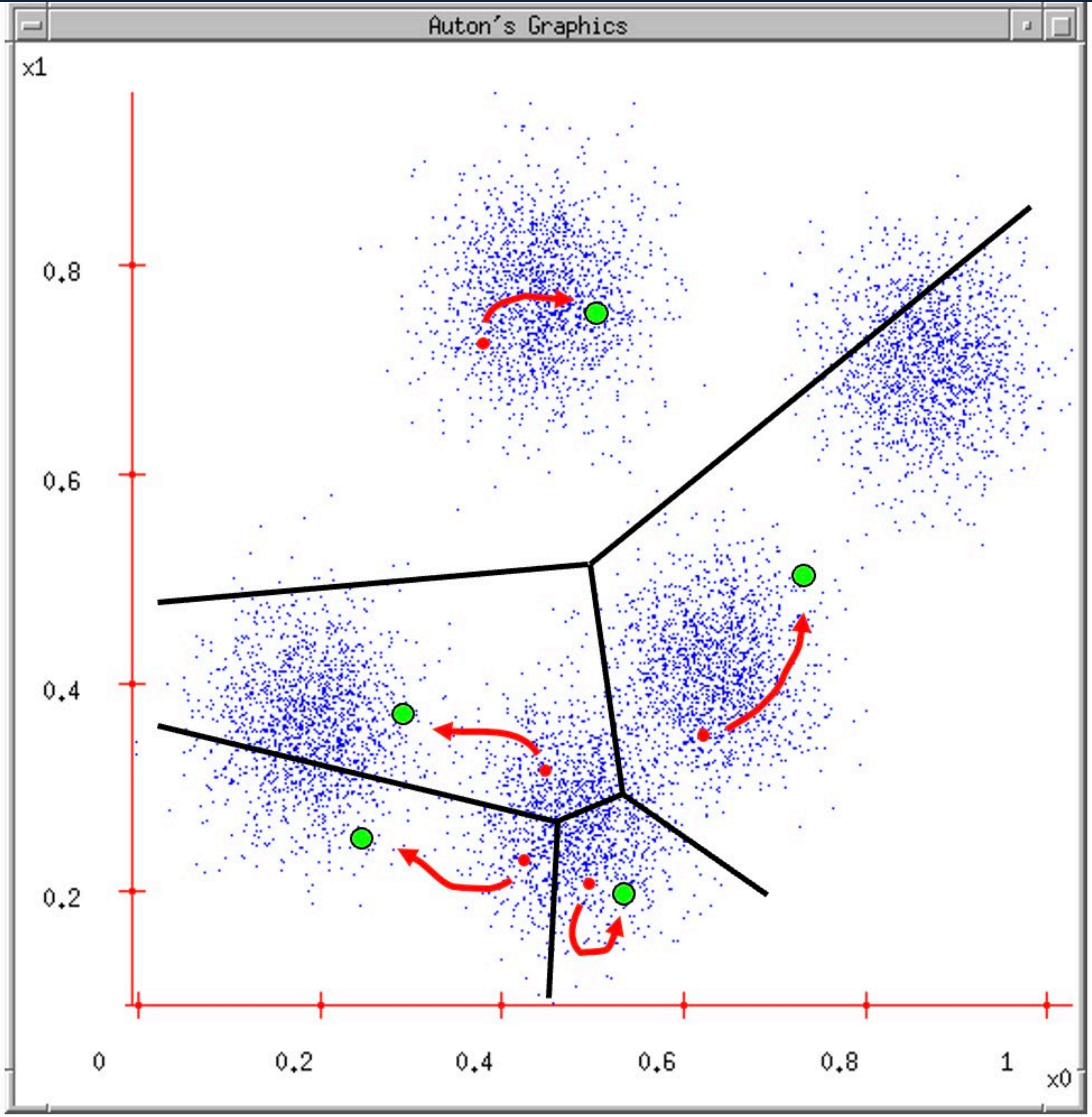
UNC GREENSBORO

K-means Clustering

Clustering:

- *Clustering Algorithms*
  - *K-means*

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



UNC GREENSBORO

K-means Clustering

Clustering:

- *Clustering Algorithms*
  - *K-means*

1.Ask user how many clusters they'd like. *(e.g. k=5)*

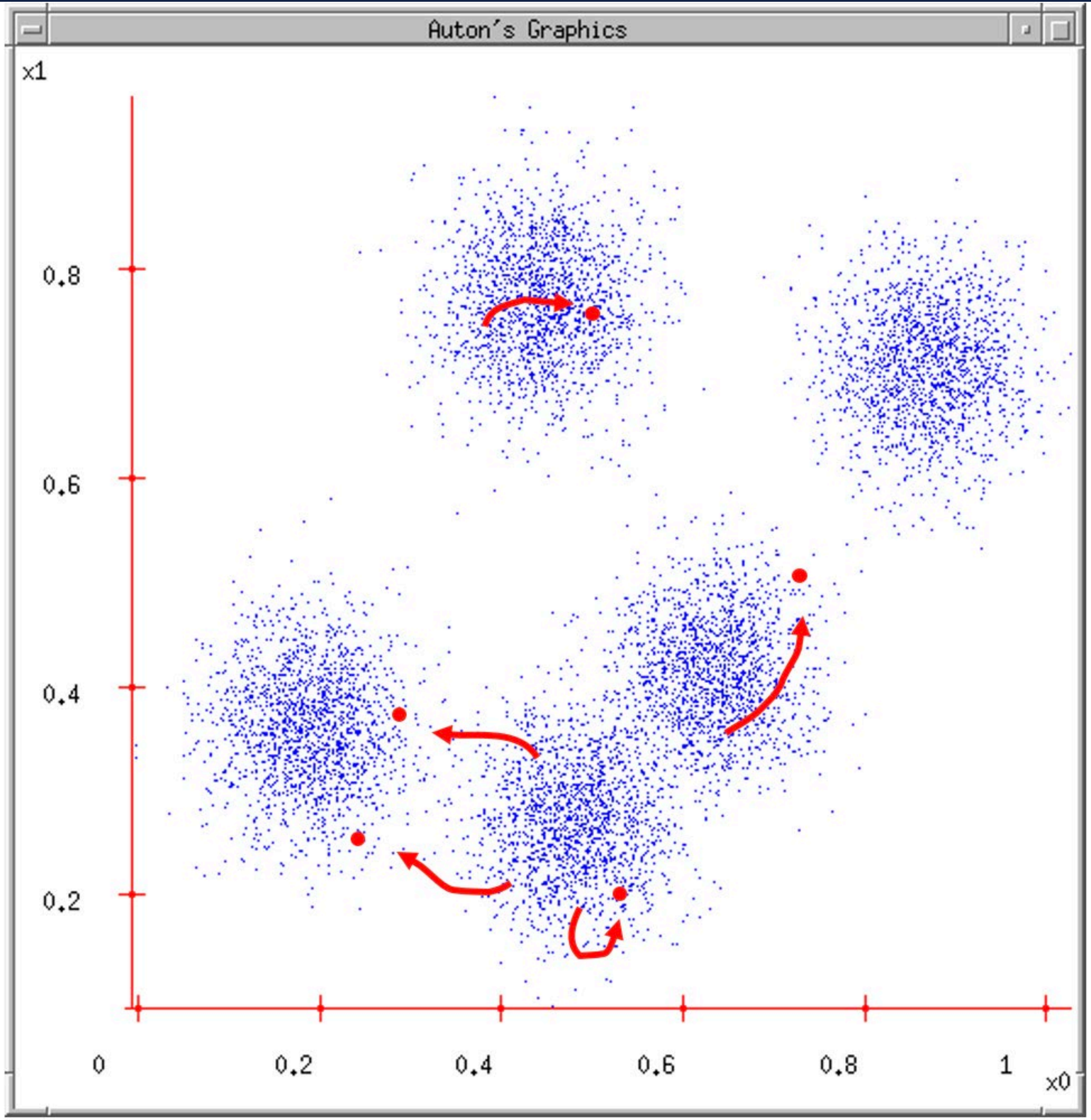
2.Randomly guess k cluster Center locations

3.Each datapoint finds out which Center it's closest to.

4.Each Center finds the centroid of the points it owns

5.…and jumps there

6.…Repeat until terminated!



UNC GREENSBORO

K-means Clustering

Clustering:

- *Clustering Algorithms*
  - *K-means*

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

K-means Clustering

Clustering:

- *Clustering Algorithms*

  - *K-means*

    1. *Partitional clustering approach*
    2. *Each cluster is associated with a centroid (center point)*
    3. *Each point is assigned to the cluster with the closest centroid*
    4. *Number of clusters, K, must be specified*
    5. *The basic algorithm is very simple*

K-means Clustering

Clustering:

- *Clustering Algorithms*
  - *K-means*
    - Strengths
      - Simple iterative method
      - User provides “K”
    - Weaknesses
      - Often too simple → bad results
      - Difficult to guess the correct “K”

K-means Clustering

Details:

- ❑ *Initial centroids are often chosen randomly.*
  - *Clusters produced vary from one run to another.*
- ❑ *The centroid is (typically) the mean of the points in the cluster.*
- ❑ *'Closeness' is measured by Euclidean distance, cosine similarity, correlation, etc.*
- ❑ Iterate:
  - Calculate distance from objects to cluster centroids.
  - Assign objects to closest cluster
  - Recalculate new centroids
- ❑ Stop based on convergence criteria
  - No change in clusters
  - Max iterations
- ❑ *Complexity is O(n * K * I * d)*

  n = number of points, K = number of clusters,

  I = number of iterations, d = number of attributes

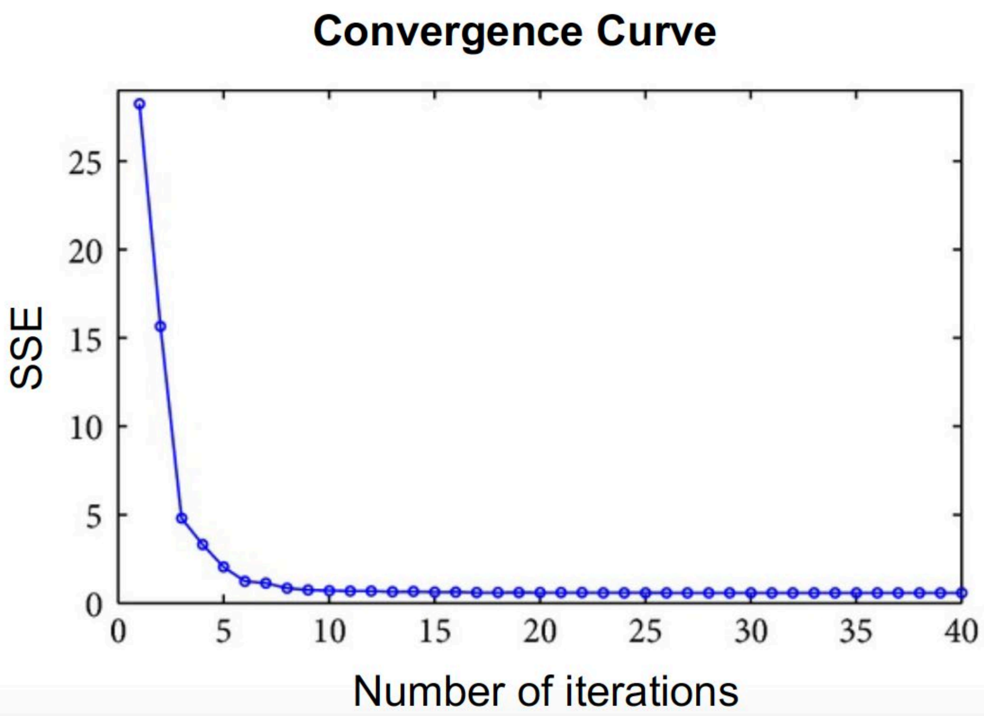
K-means Clustering

Objective function in K-means

Sum of Squared Errors (SSE)

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

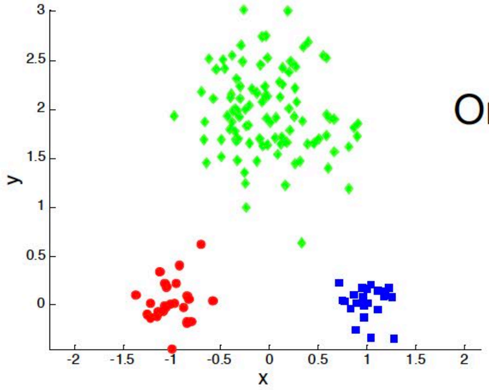
x is a data point in cluster C_i and m_i is the centroid of cluster C_i



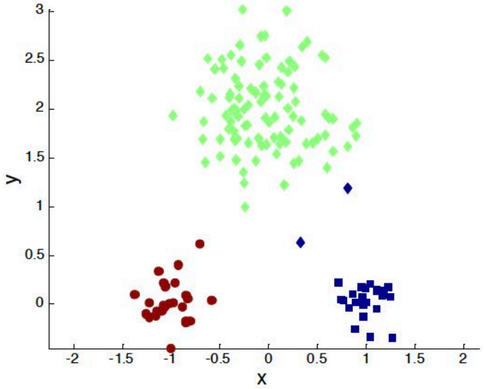
UNC GREENSBORO

K-means Clustering

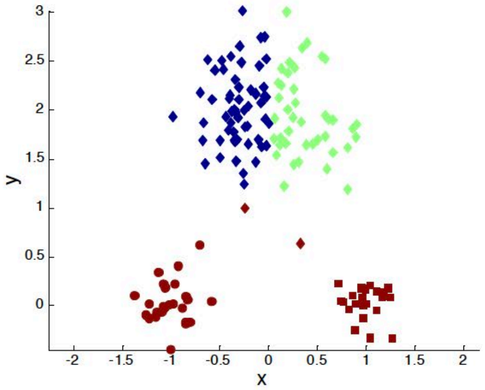
Determine numbers of clusters



Original Points



Optimal Clustering

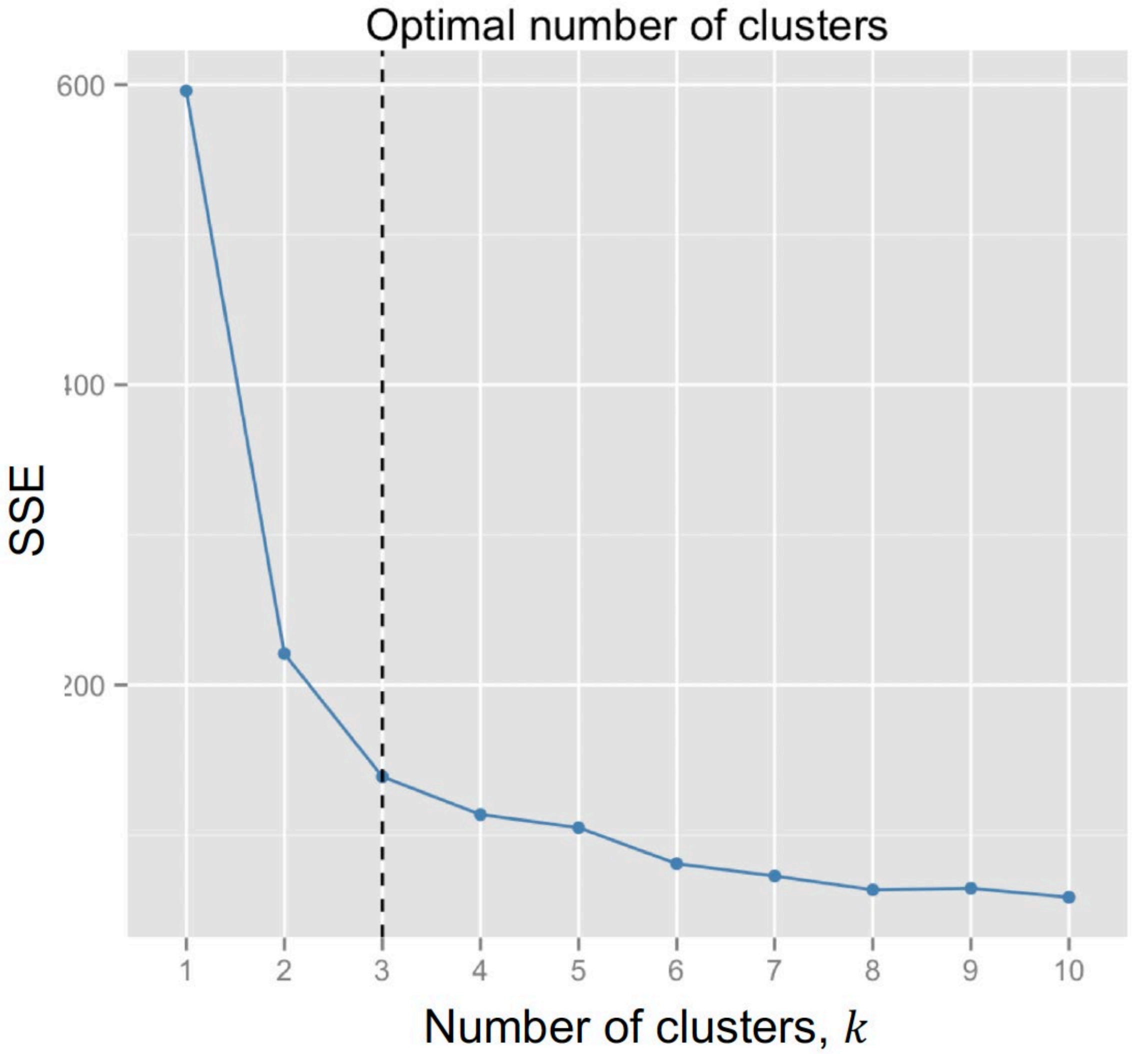


Sub-optimal Clustering

Choose the one with the smallest SSE

K-means Clustering

Determine numbers of clusters



K-means Clustering

How can you do this efficiently?

- Idea 1: Be careful about where you start
- Idea 2: Do many runs of k-means, each from a different random start configuration
- Many other ideas floating around.

K-means Clustering

How can you do this efficiently?

- Idea 1: Be careful about where you start
- Idea 2: Do ... runs of k-means, each from a diff...
- Ma...

Neat trick:
Place first center on top of randomly chosen datapoint.
Place second center on datapoint that's as far away as possible from first center
:
Place j'th center on datapoint that's as far away as possible from the closest of Centers 1 through j-1
:

UNC GREENSBORO

Exercise (15 minutes)

- **Objective**: Understand the k-means clustering algorithm by applying it to the Iris dataset and observe how it clusters the data based on flower features.

1. Import libraries and set up the environment:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
```

2. Load the dataset:

```
iris = datasets.load_iris()
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
```

3. Data Exploration:

```
use df.head(); df.describe()
```

4. Data Preprocessing: (Scale the data in k-means using StandardScaler!)

```
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df)
```

5. Apply k-means clustering (start with k=3)

```
kmeans = KMeans(n_clusters=3, random_state=42)
clusters = kmeans.fit_predict(df_scaled)
df["cluster"] = clusters
```

6. Visualize clusters:

```
plt.scatter(df_scaled[clusters == 0, 0], df_scaled[clusters == 0, 1], label='Cluster 1’)
plt.scatter(df_scaled[clusters == 1, 0], df_scaled[clusters == 1, 1], label='Cluster 2’)
plt.scatter(df_scaled[clusters == 2, 0], df_scaled[clusters == 2, 1], label='Cluster 3’)
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='red’,
label='Centroids’)
plt.xlabel('Sepal Length’)
plt.ylabel('Sepal Width’)
plt.legend()
plt.show()
```

7. Analyze results:

- Compare your results with the actual species in the dataset (provided in ‘iris.target’). How well did k-means cluster the data? Which species got mixed in the same cluster?
- Try different k, how do the clusters change?