SPOTLIGHT GRADES

	Min	Max	Mean	Median
Class average	8.79	9.55	9.32	9.26
Average (class average, instructor score)	8.74	9.68	9.46	9.35

Excellent job!

MIDTERM

- 11/8 Wednesday in class 1-2:15pm
- Open book/notes, no digital device
- Refer to midterm review and sample questions
- All ML algorithms: basic algorithm (intermediate results of each step); decision boundary; impact of key parameters; distance metrics; regularization effect; bias and variance tradeoff
- Data preprocessing; model assessment and selection strategies

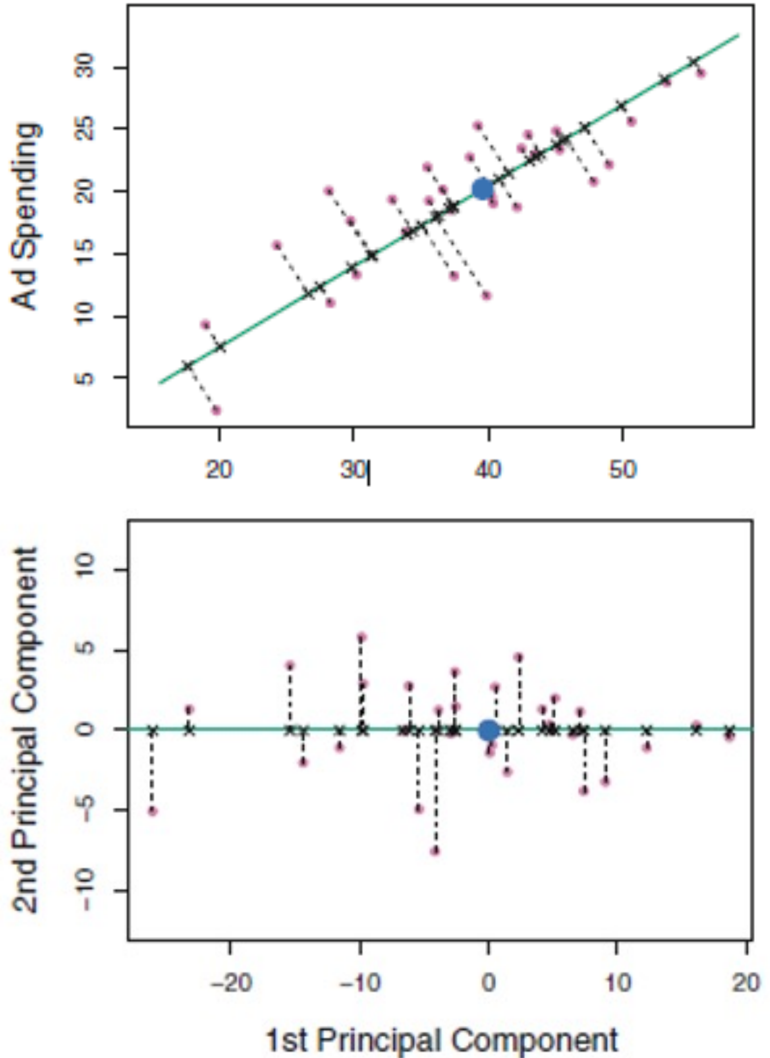
TYPES OF UNSUPERVISED LEARNING

Clustering — identify unknown structure in the data

Dimensionality Reduction — use structural characteristics to simplify data

REVIEW: PRINCIPAL COMPONENT ANALYSIS (PCA)

- First principal component
  - Yields the highest variance of the projection
- Second principal component
  - Orthogonal to the first principal component
  - And has largest variance
- In general, we can construct up to p principal components (p features)



REVIEW: PCA: PROBLEM FORMULATION

- Given a feature matrix **X** with n data points, find **W** such that ||**W**||$_2$ = 1 and the Var(**XW**) is maximized and **W** consists of orthonormal vectors

$$\begin{aligned}\mathrm{Var}(\mathbf{XW}) &= \frac{1}{N}(\mathbf{W}^\top(\mathbf{X}-\mu_\mathbf{X})^\top(\mathbf{X}-\mu_\mathbf{X})\mathbf{W}) \\ &= \mathbf{W}^\top \boxed{\mathbf{\Sigma_X}} \mathbf{W}\end{aligned}$$

Sample covariance matrix

- The solution is the Eigenvectors of the covariance matrix
    - PCs: k eigenvectors with highest eigenvalues (variance)

PCA: # OF PCS?

- How many components are sufficient to summarize the data?

PROPORTION VARIANCE EXPLAINED

- Total variance in data (assuming zero mean):

$$\mathrm{Var}(\mathbf{X}) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$$

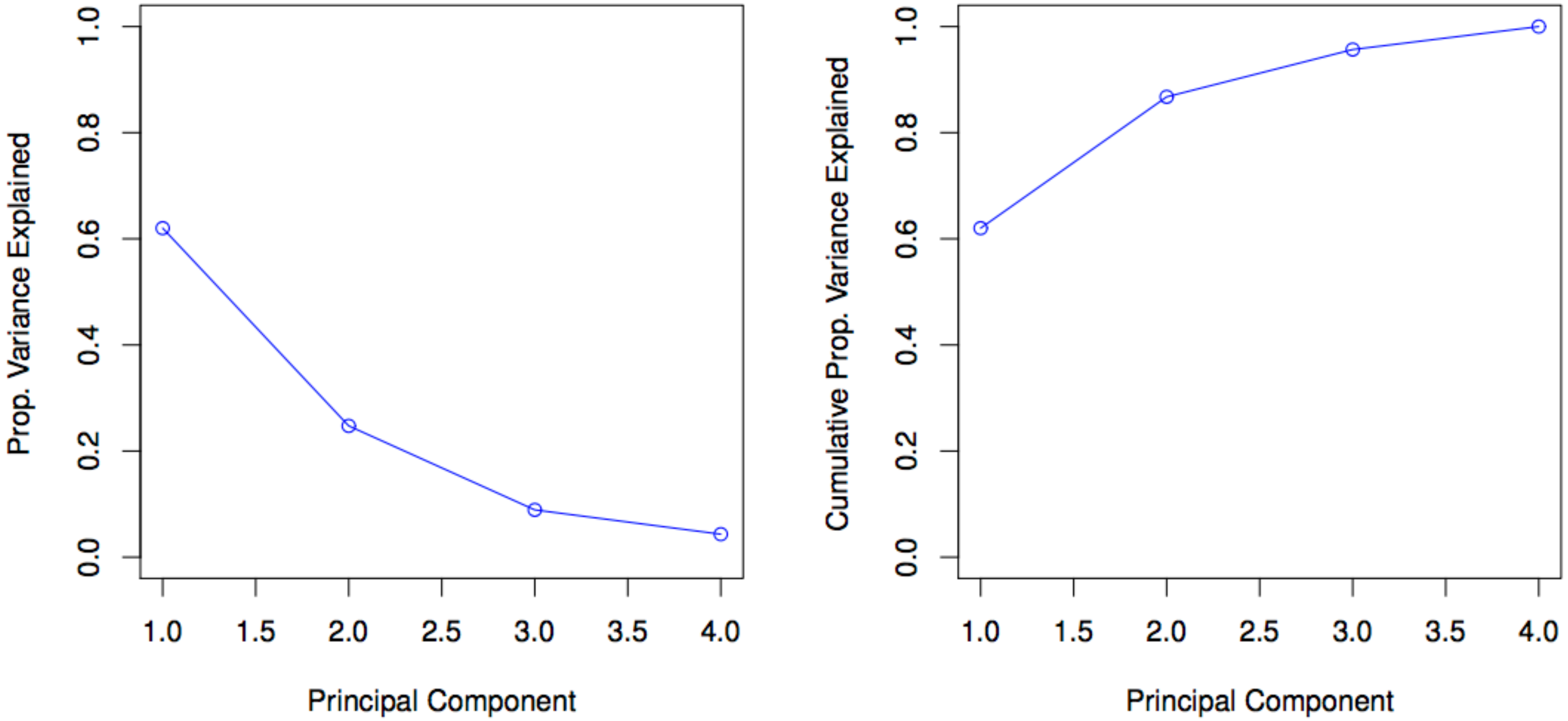
- Variance explained by the m$^{\text{th}}$ component:

$$\mathrm{Var}(\mathbf{W}_m) = \frac{1}{n} \sum_{i=1}^{n} w_{im}^2$$

- Proportion of variance explained by m$^{\text{th}}$ component :

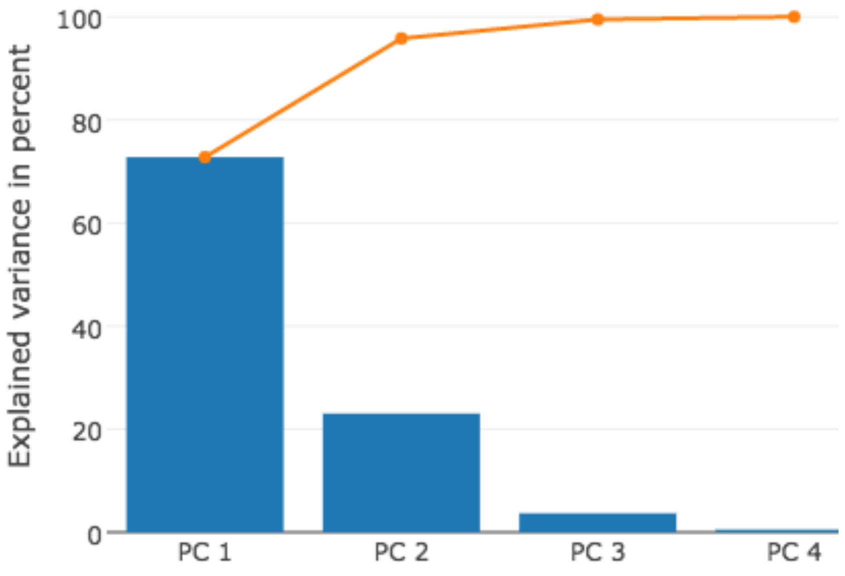
$$\mathrm{PVE}_m = \frac{\mathrm{Var}(\mathbf{W}_m)}{\mathrm{Var}(\mathbf{X})}$$

PCA: SCREE PLOT (UNSUPERVISED)



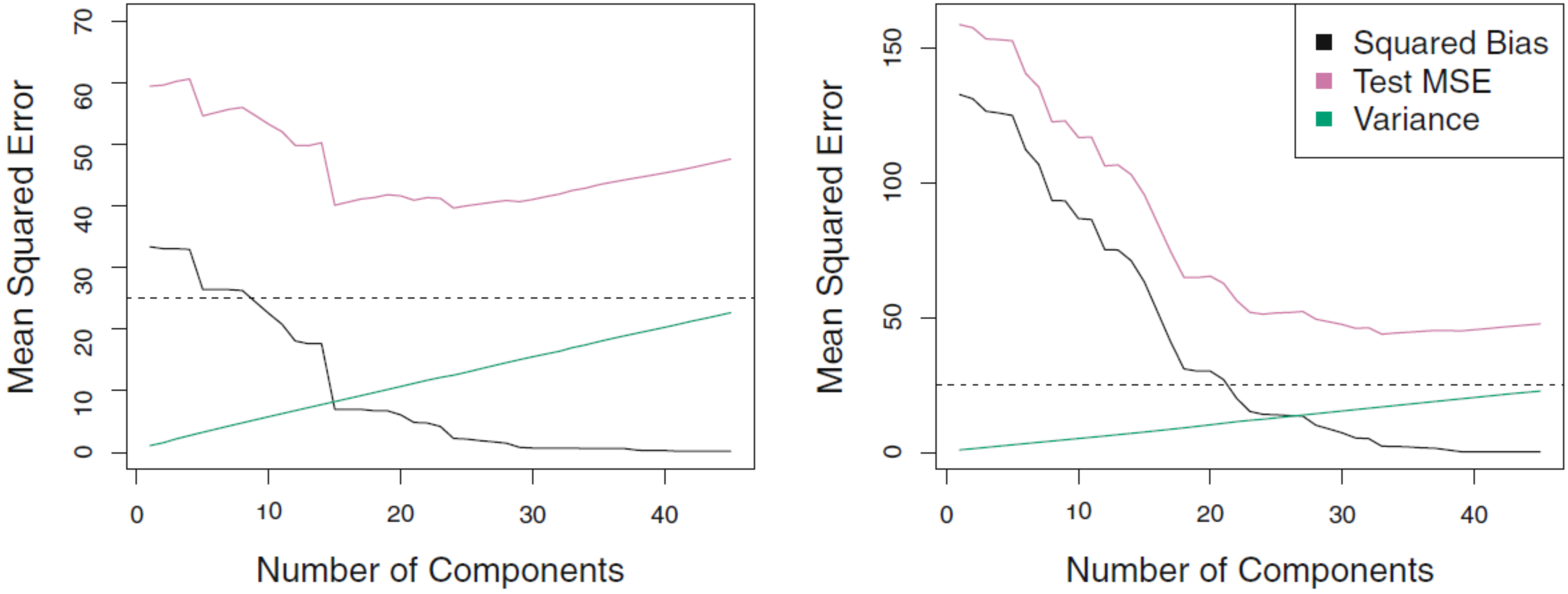
PCA: INTERPRETATION

- If variances of PCs drop off quickly, then X is highly collinear
- Reduce dimensionality of data by keeping only the PCs with highest variance



https://plot.ly/ipython-notebooks/principal-component-analysis/

USE PCA FOR SUPERVISED LEARNING



PCA: SKLEARN

```
from sklearn.decomposition import PCA as sklearnPCA
sklearn_pca = sklearnPCA(n_components=2)
Y_sklearn = sklearn_pca.fit_transform(X_std)
```

REMINDER: HOMEWORK #5

- Due 11/16 @ 11:59 PM ET on Gradescope
- 2 questions
  - PCA
  - Almost Random Forest

DEMO: PCA-EXAMPLE.IPYNB

HTTPS://COLAB.RESEARCH.GOOGLE.COM/DRIVE/12VDO-JD0PVFLVZZLI2FT50DYESJ8V6WW

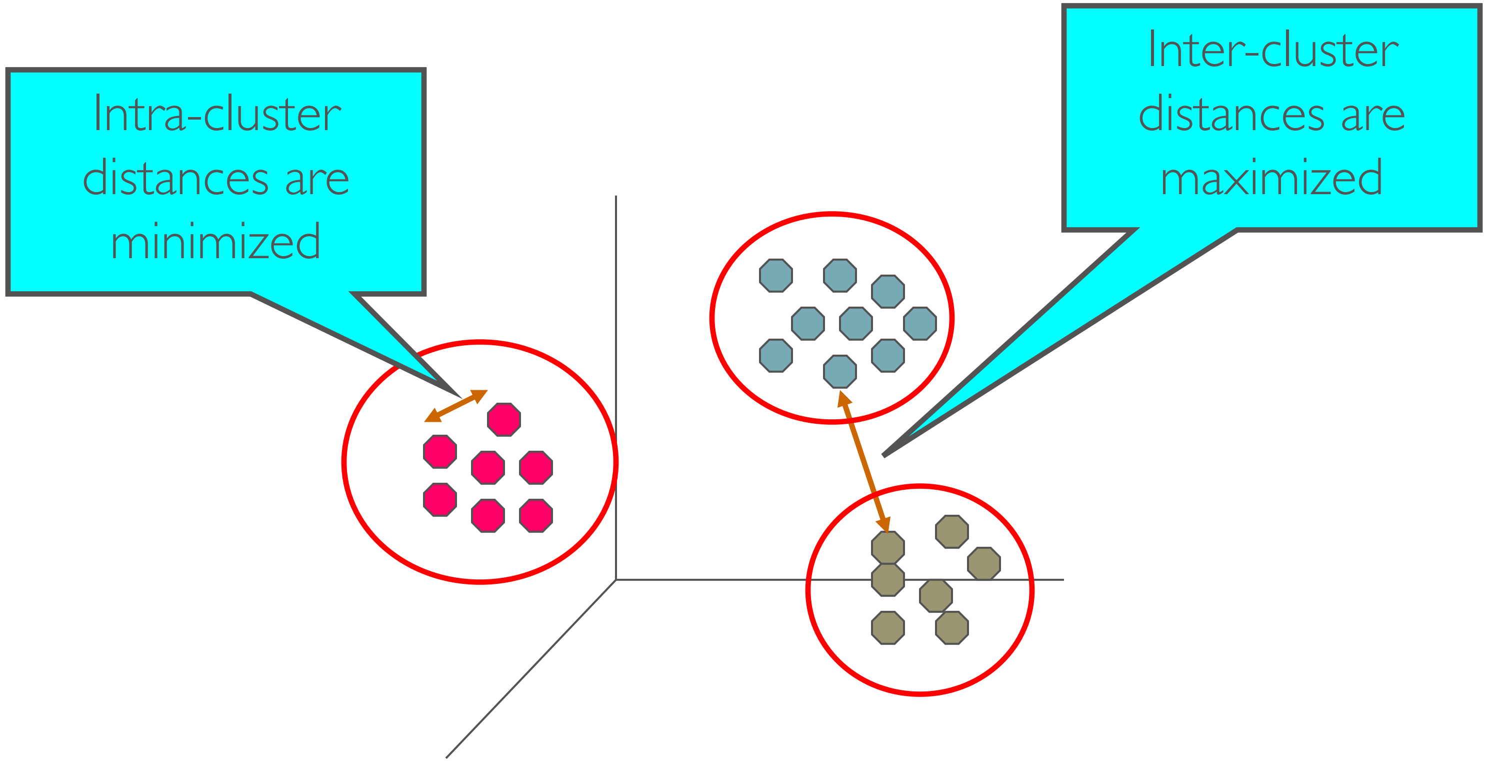
TYPES OF UNSUPERVISED LEARNING

Clustering — identify unknown structure in the data

Dimensionality Reduction — use structural characteristics to simplify data

WHAT IS CLUSTER ANALYSIS?

- Finding groups of objects (clusters)
  - Objects similar to one another in the same group
  - Objects different from the objects in other groups
- Unsupervised learning: no predefined classes



How to define similarity?

APPLICATIONS OF CLUSTER ANALYSIS

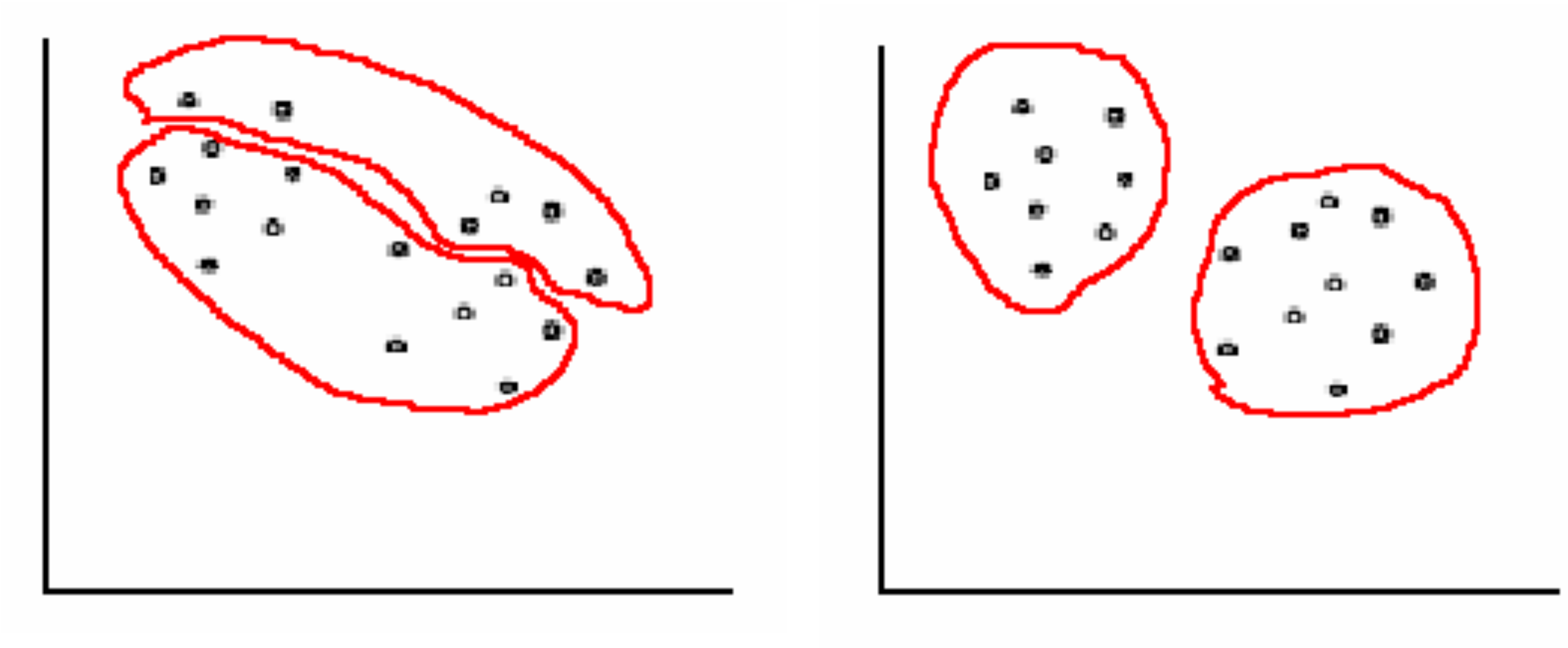
- As a stand-alone tool to get insight into data distribution
  - Cluster into groups – automatic classification
  - Finding k-nearest neighbors
  - Outlier detection
- As a preprocessing step for other algorithms
  - Data cleaning: missing data, noisy data; Data reduction; Data discretization
  - Build supervised models for each cluster

CLUSTERING APPROACHES

- Partitioning approach:
  - Construct various partitions and then evaluate them by some "goodness" criterion
  - Typical methods: **k-means**, k-medoids
- Hierarchical approach:
  - Create a hierarchical decomposition of the objects
  - Typical methods: Diana, Agnes
- Density-based approach:
  - Based on connectivity and density functions (vs. distance)
  - Typical methods: DBSACN
- Others

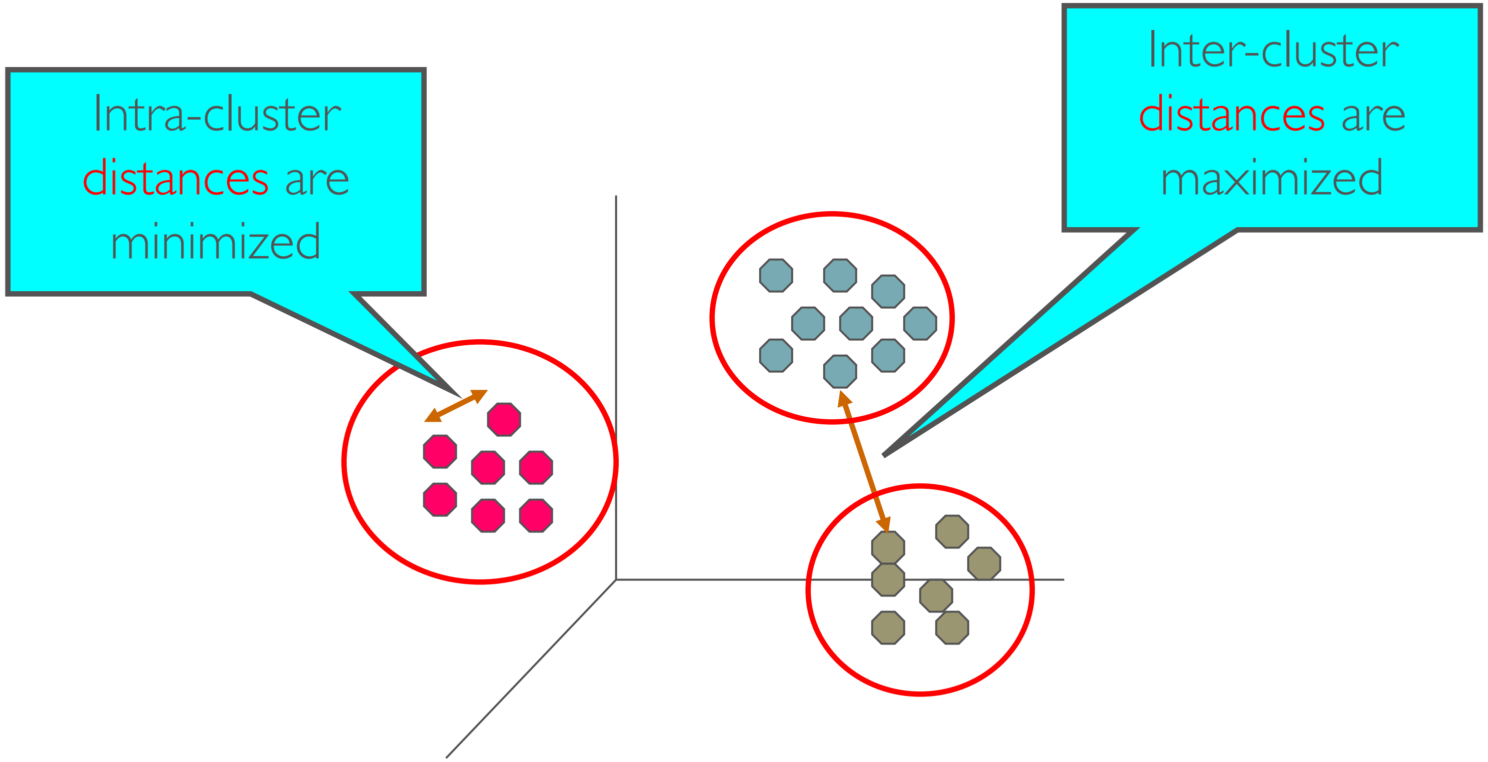
GROUP ACTIVITY

Which is better and why?
What is good clustering criteria?



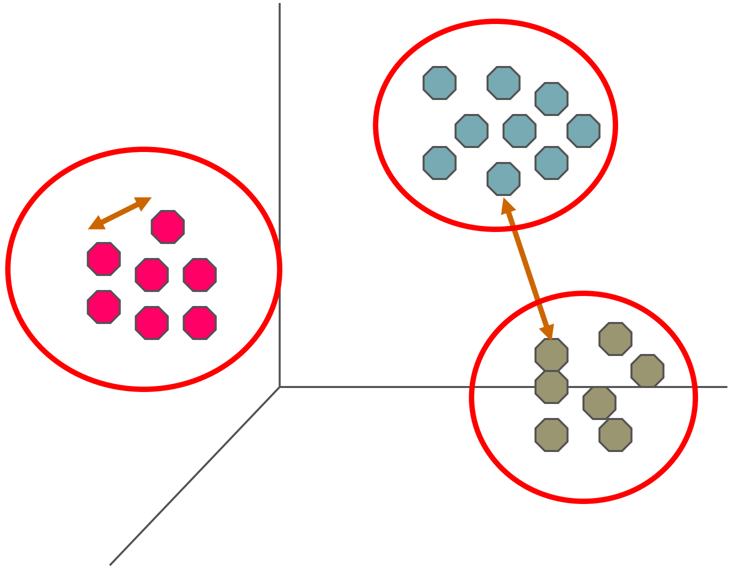
QUALITY: WHAT IS GOOD CLUSTERING?

- Homogeneity - high intra-class similarity
- Separation - low inter-class similarity



PARTITIONING ALGORITHMS: BASIC CONCEPT

- Partitioning method: Construct a partition of *n* objects (into *k* clusters), s.t. intra-cluster similarity maximized and inter-cluster similarity minimized



- One objective: minimize the sum of squared distance from cluster centroid (intra-class distance)

$$\Sigma_{i=1}^{k}\Sigma_{p\in C_i}(p-m_i)^2$$

- How to find optimal partition?

NUMBER OF PARTITIONINGS

- Stirling partition number – number of ways to partition n objects into k non-empty subsets

$$\left\{ {n+1 \atop k} \right\} = k \left\{ {n \atop k} \right\} + \left\{ {n \atop k-1} \right\}$$

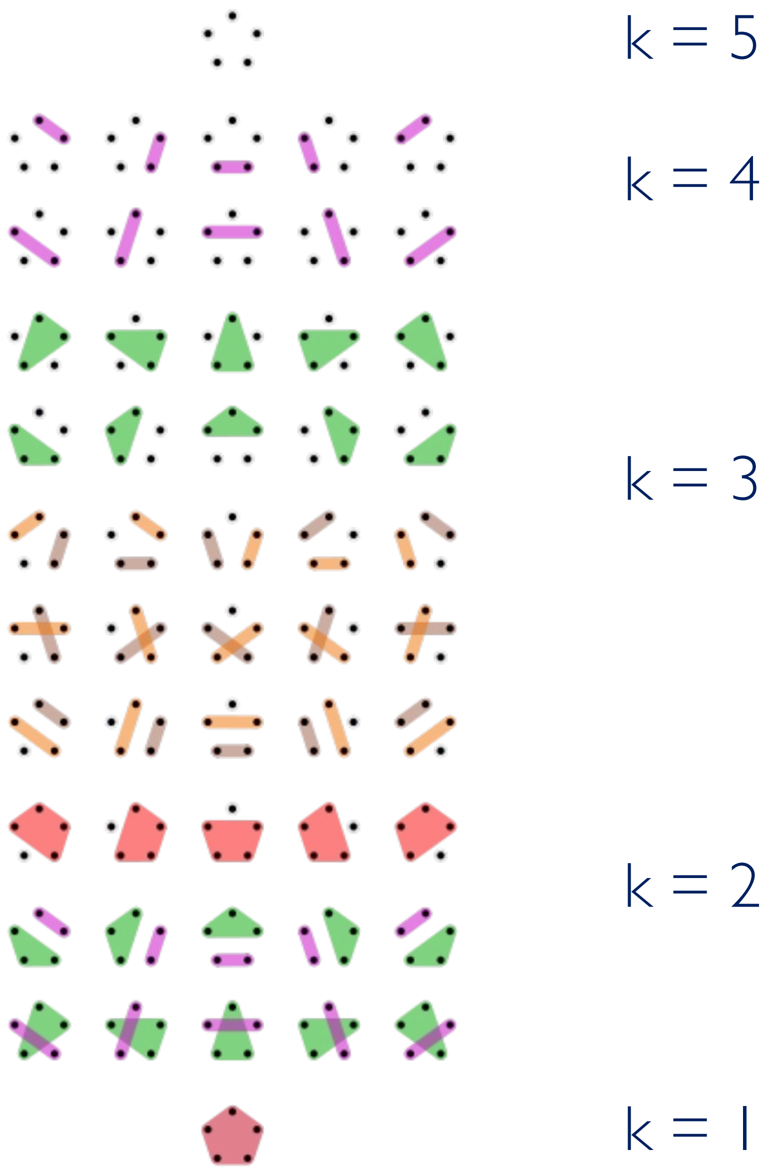
(n= 5, k = 1, 2, 3, 4, 5): 1, 15, 25, 10, 1

(n=10, k = 1, 2, 3, 4, 5, …): 1, 511, 9330, 34105, 42525, …

- Bell numbers – number of ways to partition n objects

$$B_n = \sum_{k=0}^{n} \left\{ {n \atop k} \right\}.$$

(n = 0, 1, 2, 3, 4, 5, …): 1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975, 678570, 4213597, 27644437, 190899322, 1382958545, 10480142147, 82864869804, 682076806159, 5832742205057, ...

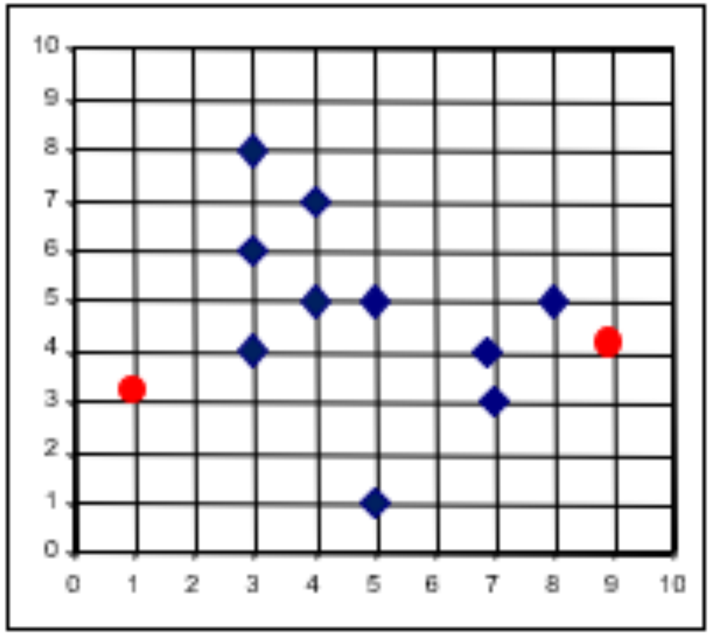


K-MEANS CLUSTERING: LLOYD ALGORITHM

- Lloyd'57, MacQueen'67
- Heuristic EM-style algorithm for the partitioning problem
- Each cluster is represented by the center of the cluster

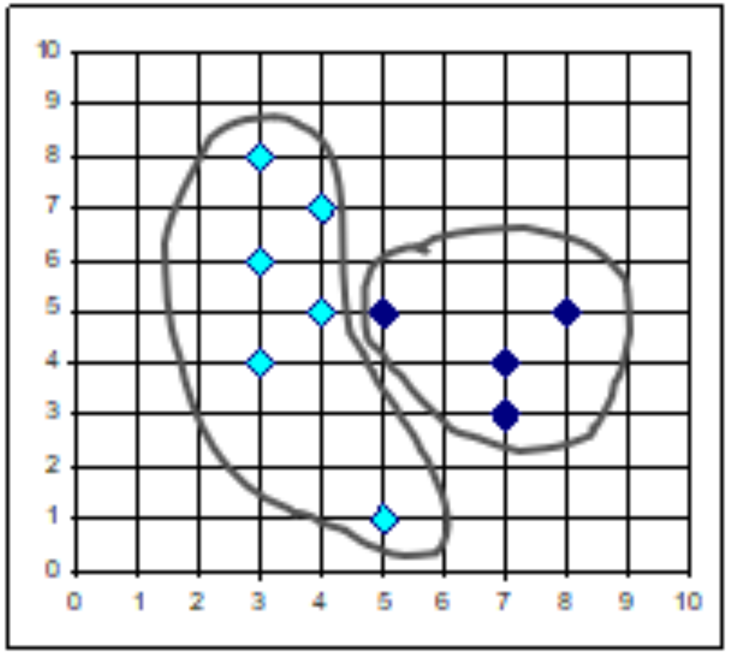
K-MEANS CLUSTERING: LLOYD ALGORITHM
(LLOYD'57, MACQUEEN'67)

- If we know the centroid of each cluster, how do we cluster the points?



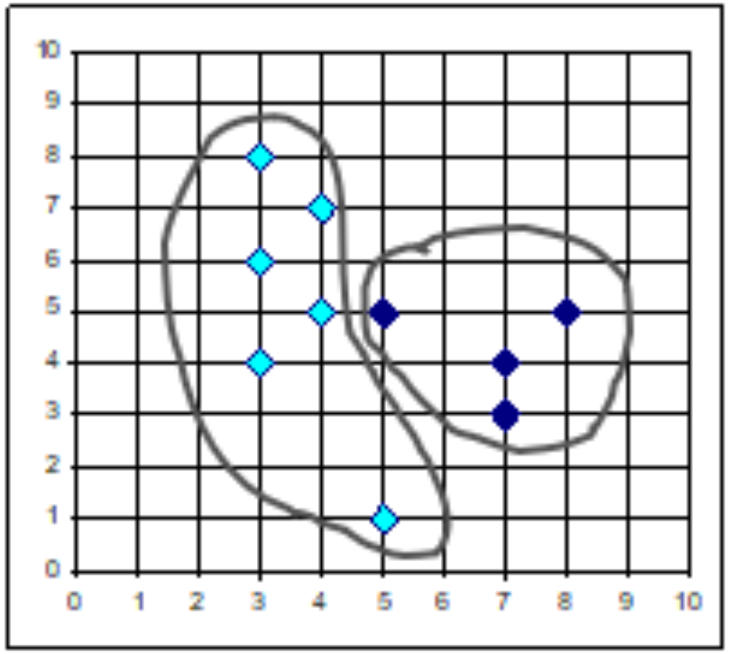
K-MEANS CLUSTERING: LLOYD ALGORITHM
(LLOYD'57, MACQUEEN'67)

- If we know the centroid of each cluster, how do we cluster the points?
- If we know the clusters, how do we compute centroid?



K-MEANS CLUSTERING: LLOYD ALGORITHM
(LLOYD'57, MACQUEEN'67)

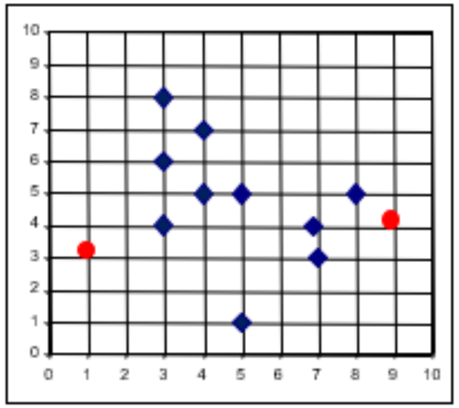
- If we know the centroid of each cluster, how do we cluster the points?
- If we know the clusters, how do we compute centroid?

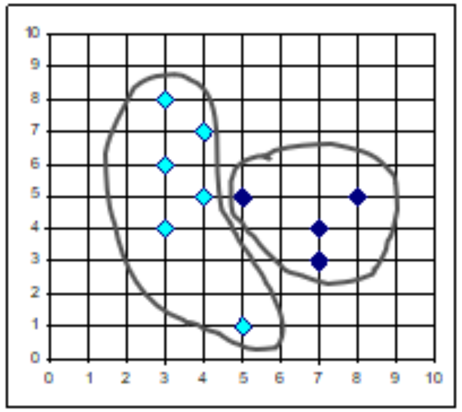


How do we get started?

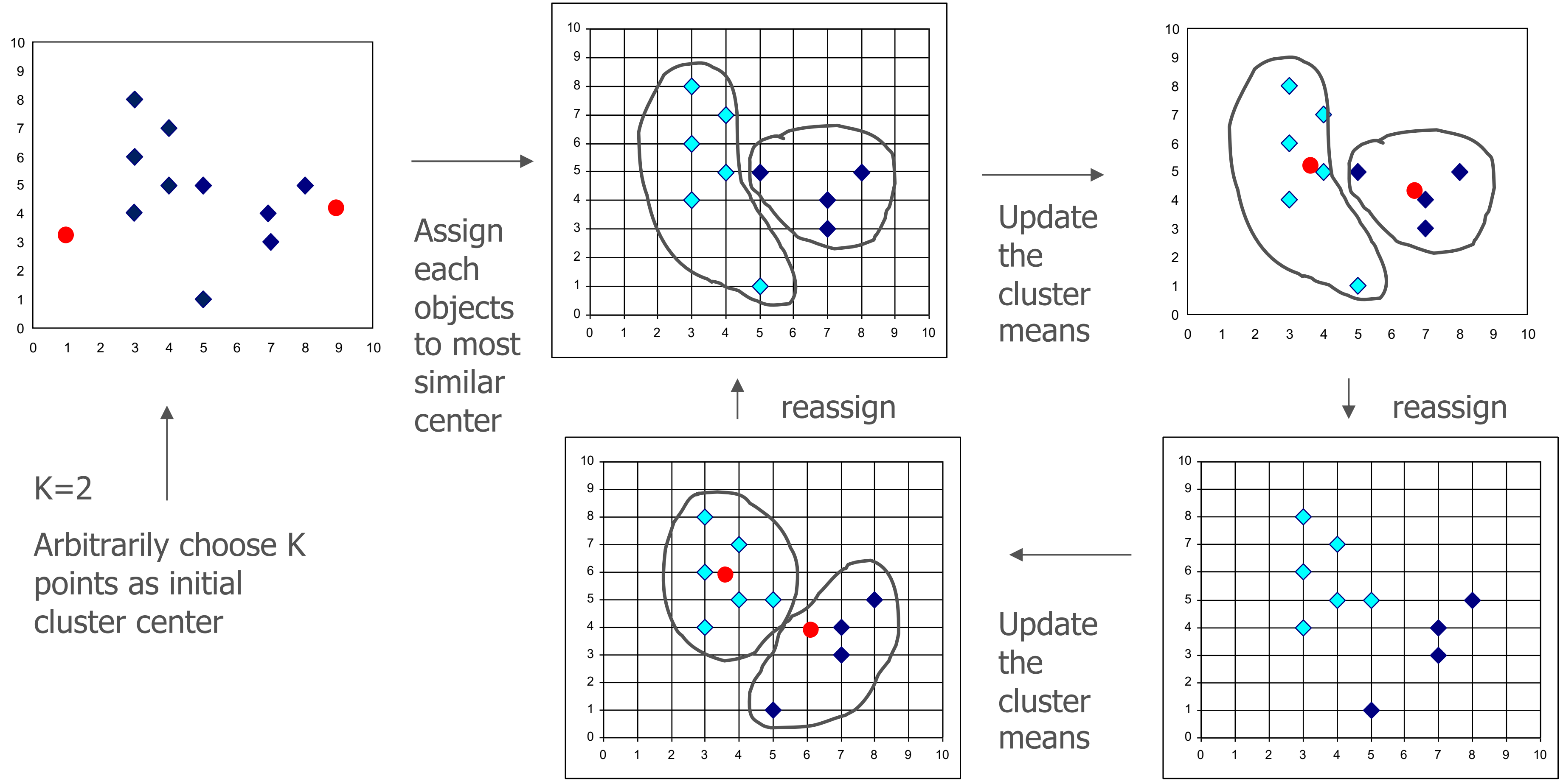
K-MEANS CLUSTERING: LLOYD ALGORITHM
(LLOYD'57, MACQUEEN'67)

- Initialization: Given *k*, randomly choose *k* initial cluster centers
- Assignment: Partition objects into *k* nonempty subsets by assigning each object to the cluster with the nearest centroid
- Update: update centroid, i.e. *mean point* of the cluster
- Go back to Step 2 and repeat, stop when no more new assignment and centroids do not change

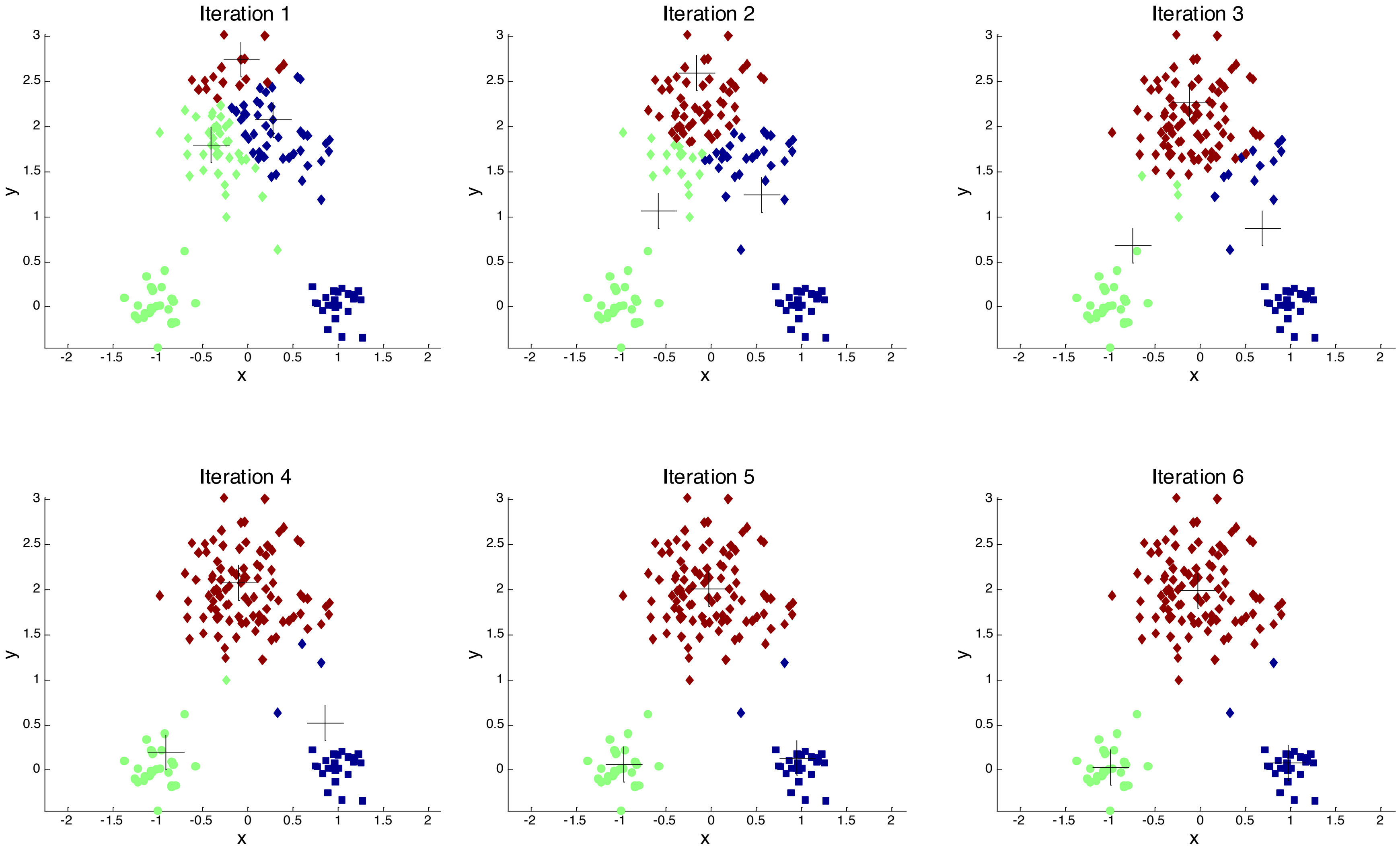




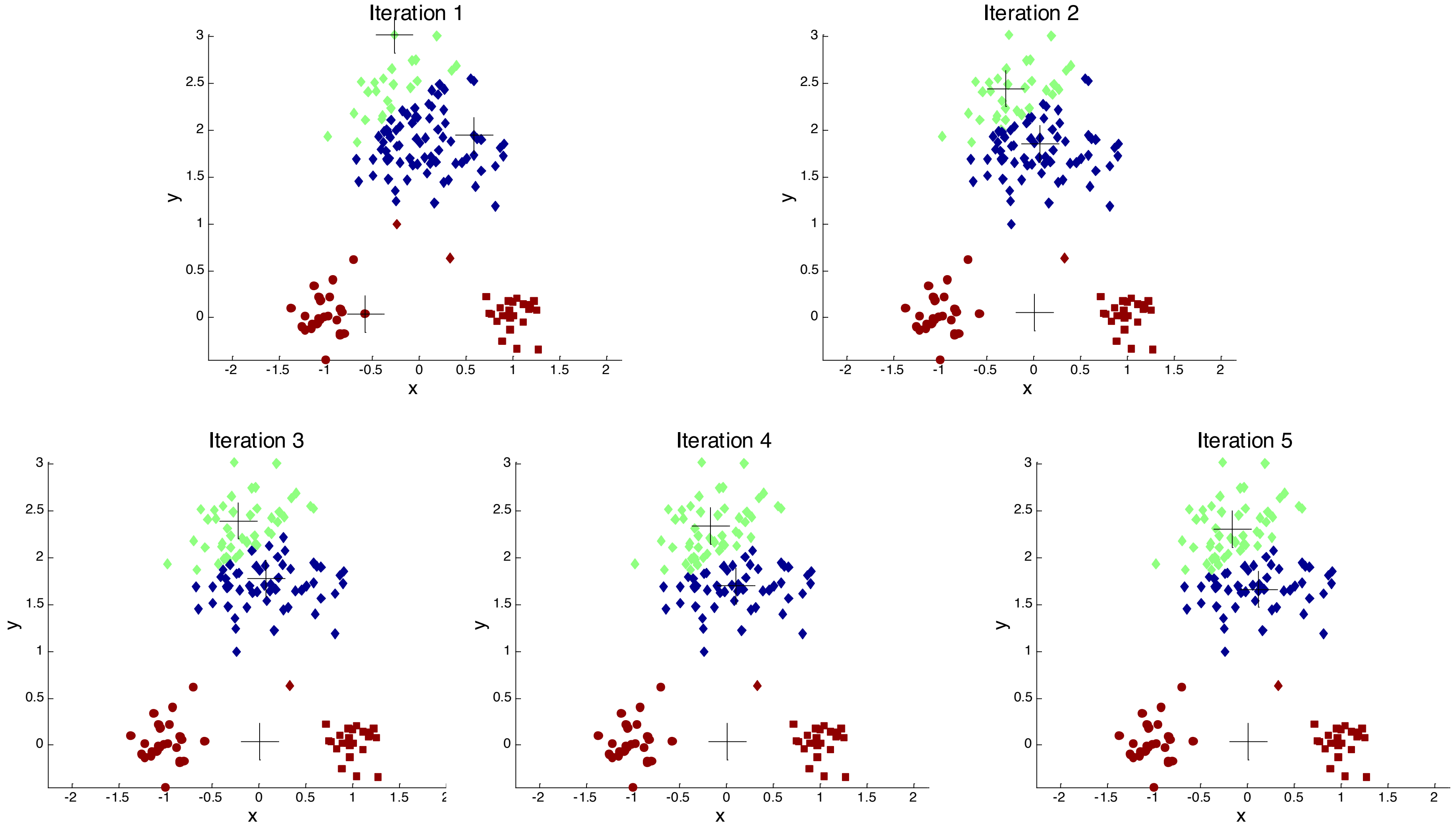
THE *K-MEANS* CLUSTERING METHOD



IMPORTANCE OF CHOOSING INITIAL CENTROIDS – CASE 1



IMPORTANCE OF CHOOSING INITIAL CENTROIDS – CASE 2



How do we avoid bad initial cases?

K-MEANS CLUSTERING – DETAILS

- Initial centroids are often chosen randomly
  - Example: Pick one point at random, then k-1 other points, each as far away as possible from the previous points
  - Run multiple times
- 'Nearest' is measured by Euclidean distance, cosine similarity, correlation, etc.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to 'Until relatively few points change clusters'
- What's the complexity? (Minibatch k-means further reduces complexity)

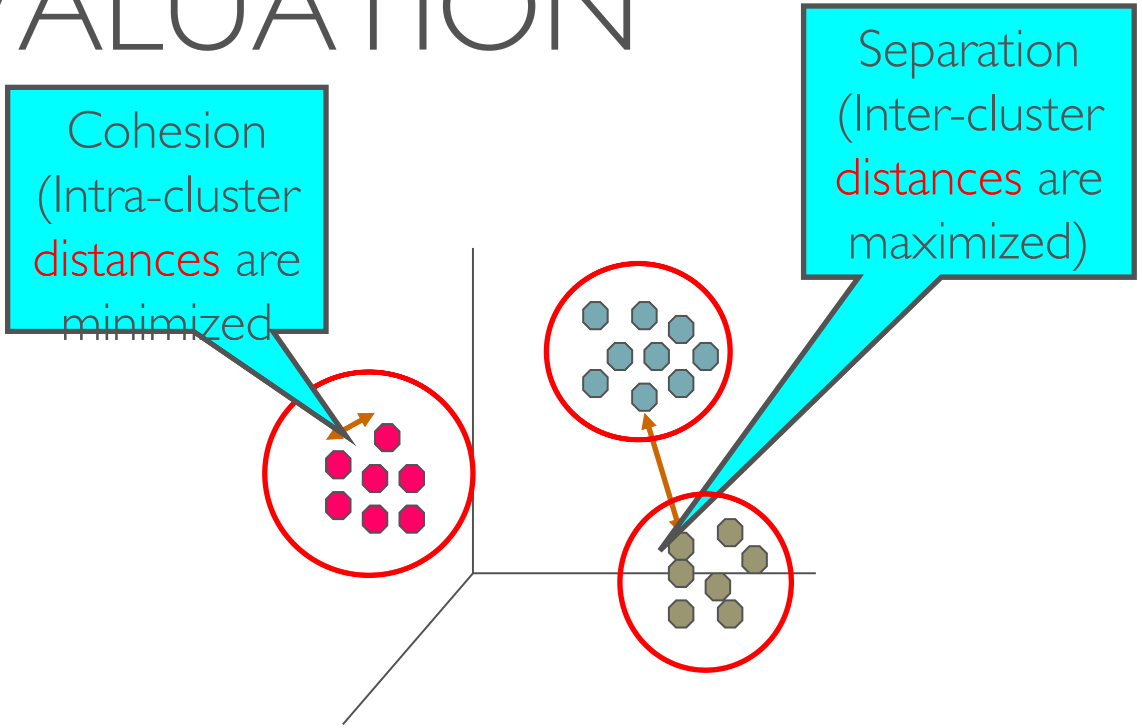
n is # objects, *k* is # clusters, and *t* is # iterations.

CLUSTERING EVALUATION

- SSE (sum of squared error)

$$\Sigma_{i=1}^{k}\Sigma_{p\in C_i}(p-m_i)^2$$

- Silhouette coefficient [-1, 1]



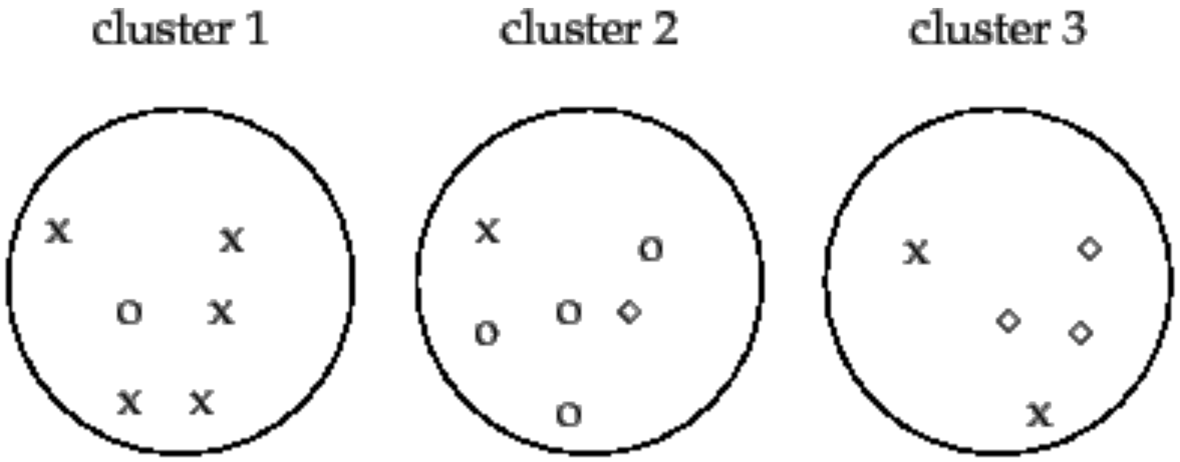
a(i): The mean distance between a sample i and all other points in the same cluster
b(i): The mean distance between a sample i and all other points in the next nearest cluster.

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i,j) \qquad b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j)$$

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_I| > 1$$

CLUSTERING EVALUATION - SUPERVISED

- Compare clusters with "ground truth" clusters
  - Entropy/purity based
  - Precision and recall based
  - Similarity-based



► **Figure 16.1** Purity as an external evaluation criterion for cluster quality. Majority class and number of members of the majority class for the three clusters are: x, 5 (cluster 1); o, 4 (cluster 2); and ⋄, 3 (cluster 3). Purity is $(1/17) \times (5+4+3) \approx 0.71$.

PRECISION AND RECALL BASED

- BCubed Precision and recall – average precision and recall of all objects
  - Precision of an object: proportion of objects in the same cluster belong to the same category
  - Recall of an object: proportion of objects of the same category are assigned to the same cluster

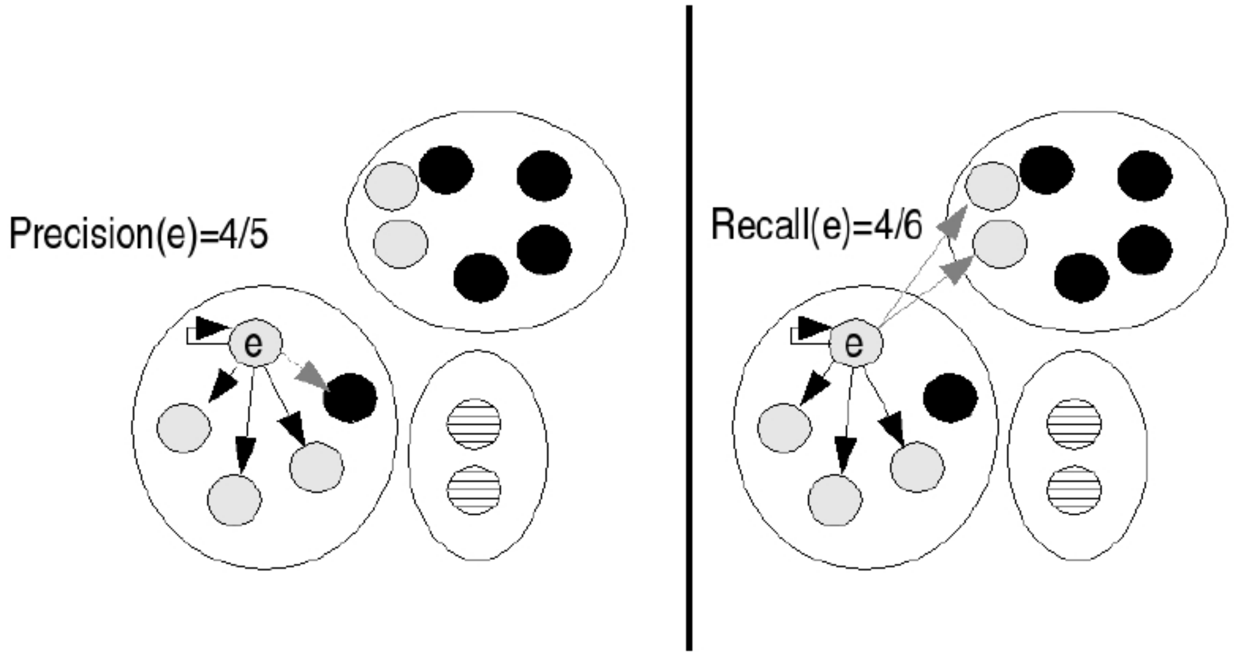


Figure 10: Example of computing the BCubed precision and recall for one item

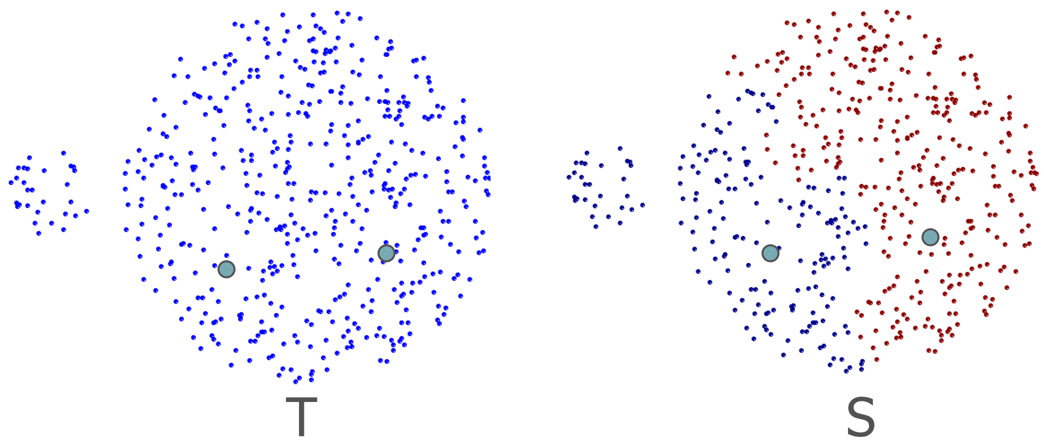
SIMILARITY-BASED MEASURES

Given a reference clustering T and clustering S

- f_{00}: number of pair of points belonging to different clusters in both T and S
- f_{01}: number of pair of points belonging to different cluster in T but same cluster in S
- f_{10}: number of pair of points belonging to same cluster in T but different cluster in S
- f_{11}: number of pair of points belonging to same clusters in both T and S

$$Rand = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

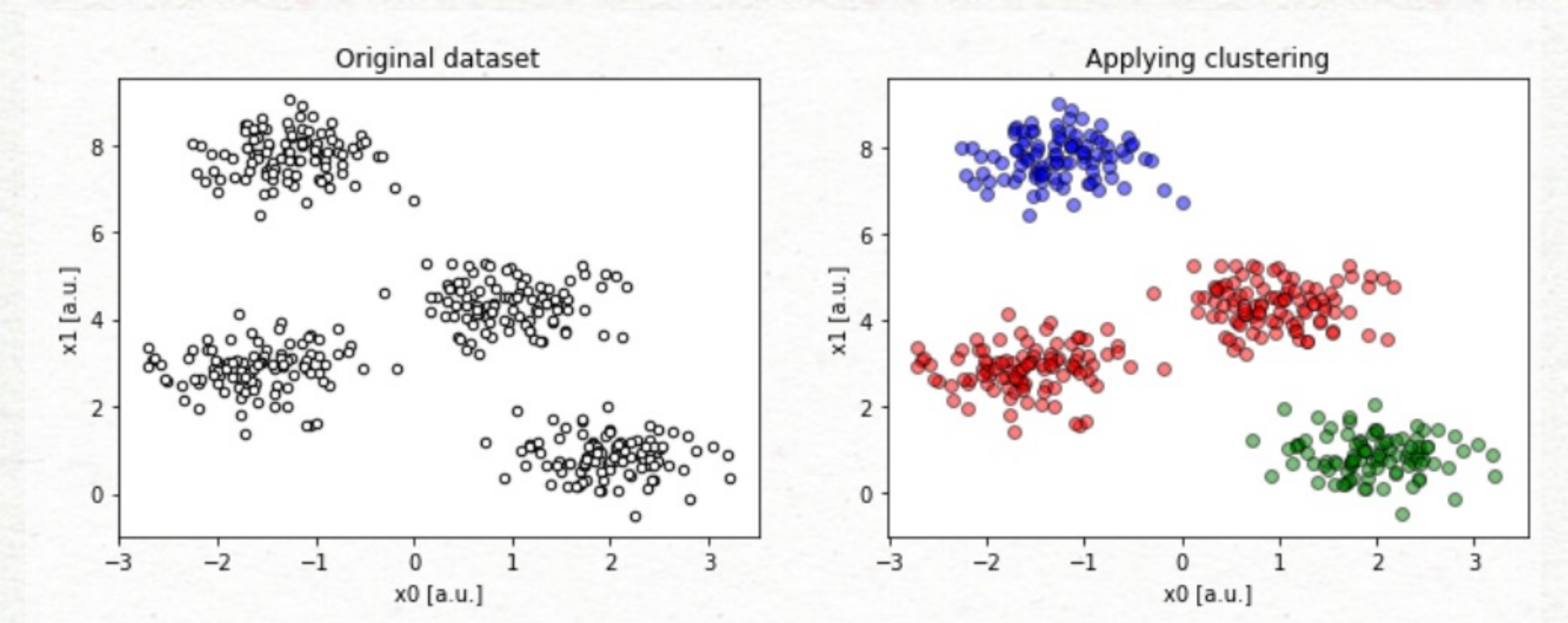
$$Jaccard = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$



K-MEANS CLUSTERING: # OF K?

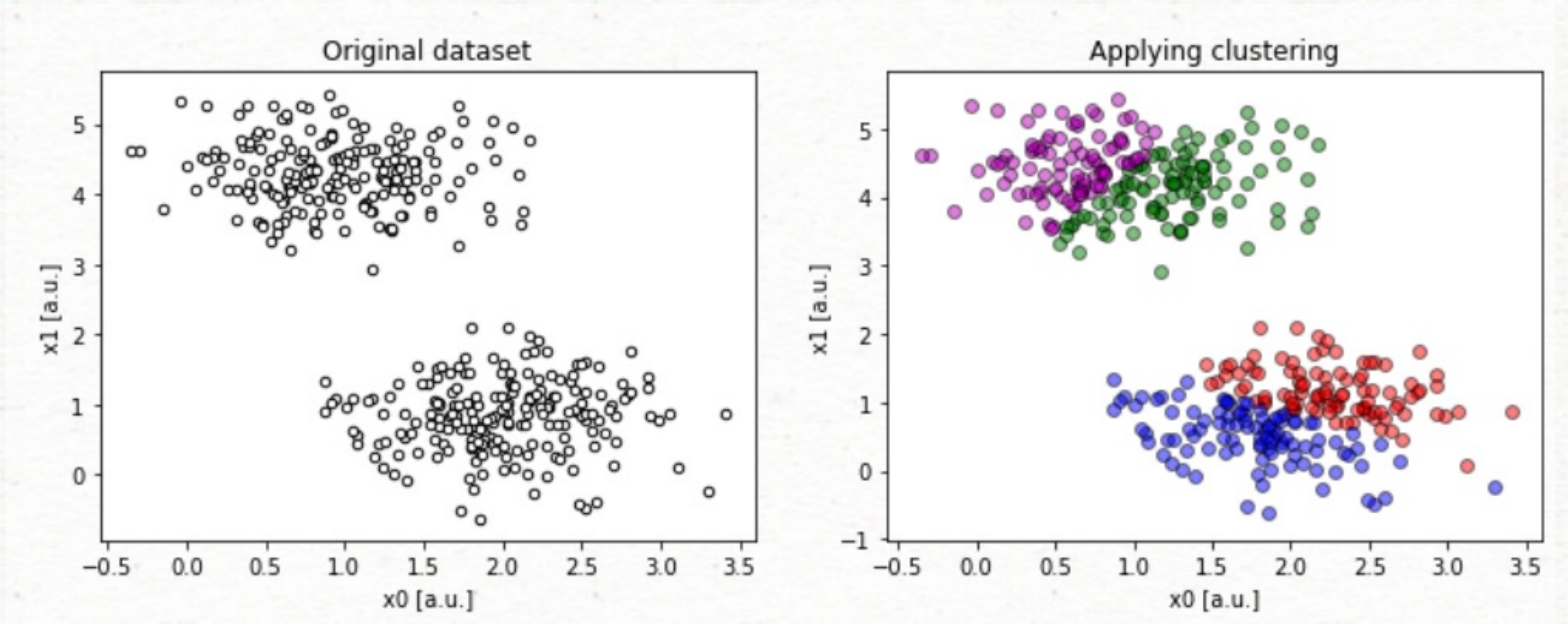
- How to choose the number of clusters K?

K: TOO FEW



Original dataset
x1 [a.u.]
x0 [a.u.]
−3
−2
−1
0
1
2
3
0
2
4
6
8
Applying clustering
x1 [a.u.]
x0 [a.u.]

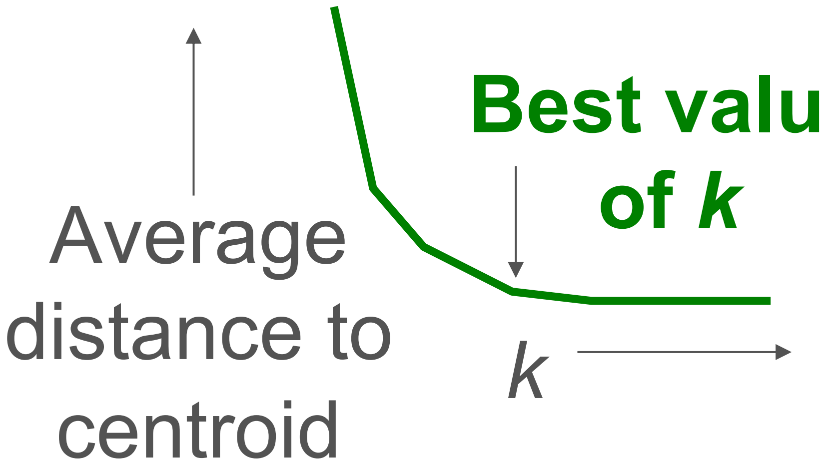
K: TOO MANY



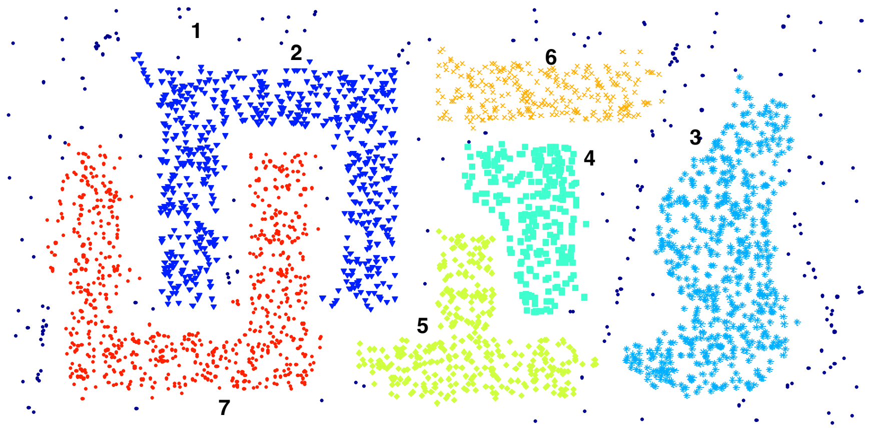
GETTING THE *K* RIGHT – ELBOW METHOD

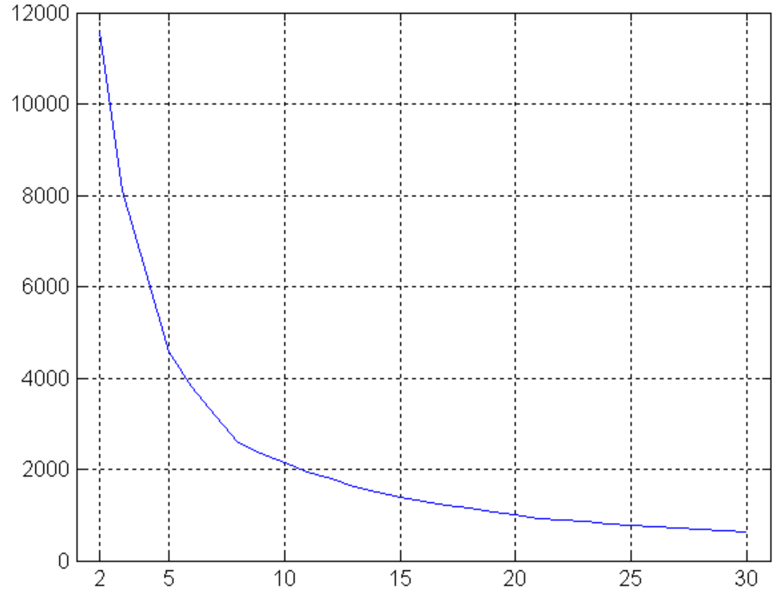
- Try different k, looking at the change in clustering criterion (e.g. average distance to centroid, SSE, or clustering coefficient) as k increases

- Average distance typically falls rapidly until right k, then changes little



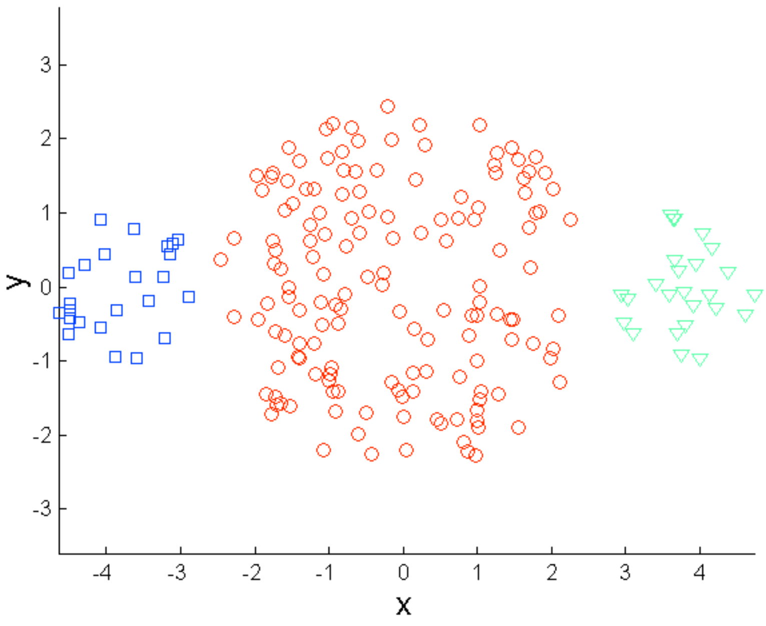
CHOOSING K - ELBOW METHOD



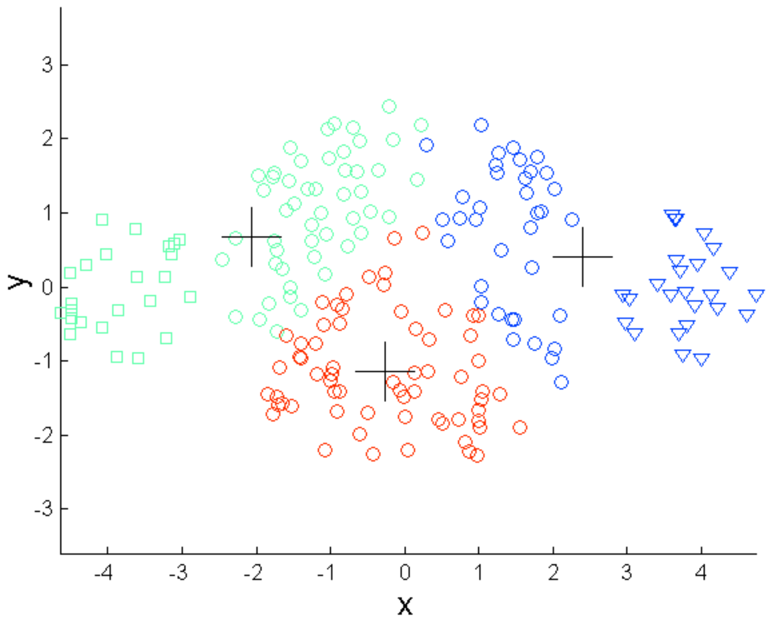


SSE of clusters found using K-means

LIMITATIONS OF K-MEANS: DIFFERING SIZES

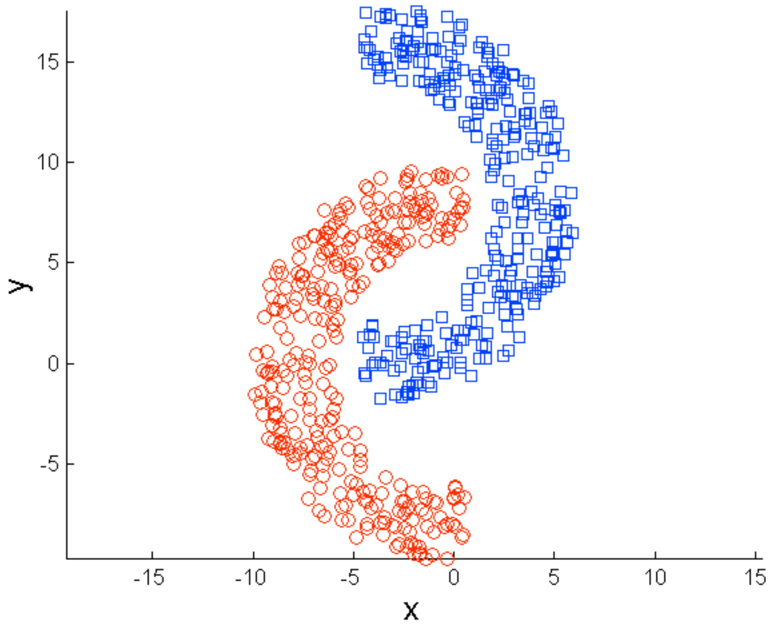


Original Points

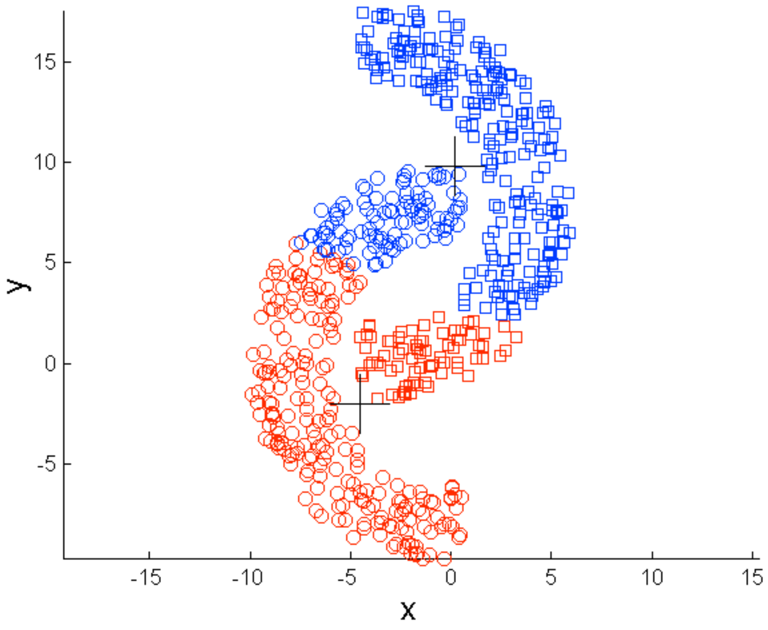


K-means (3 Clusters)

LIMITATIONS OF K-MEANS: NON-CONVEX SHAPES

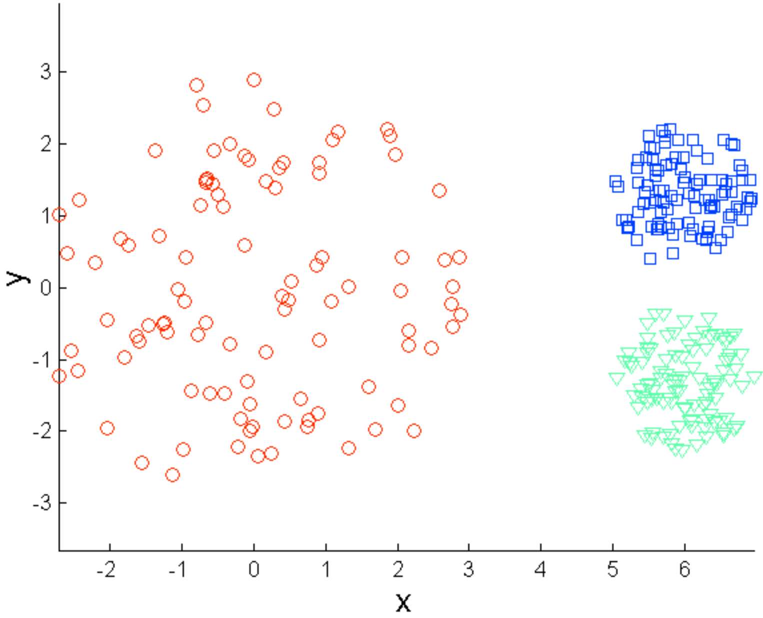


Original Points

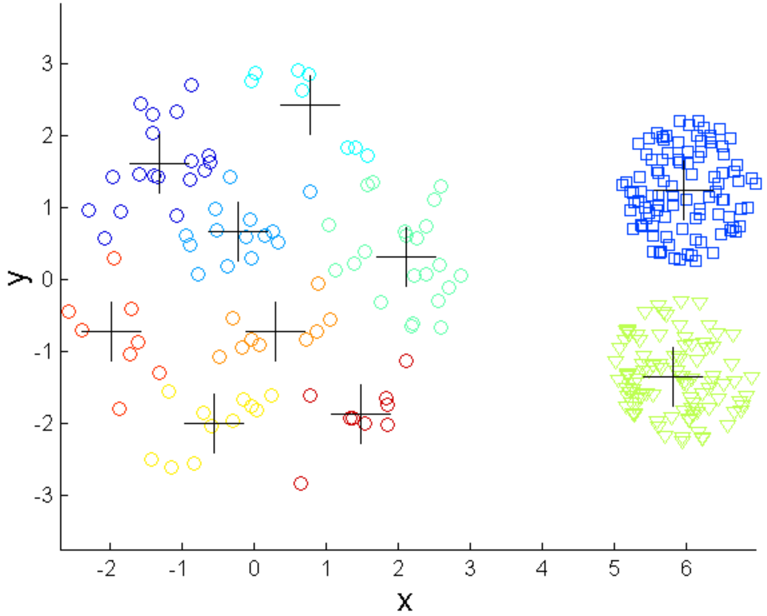


K-means (2 Clusters)

OVERCOMING K-MEANS LIMITATIONS

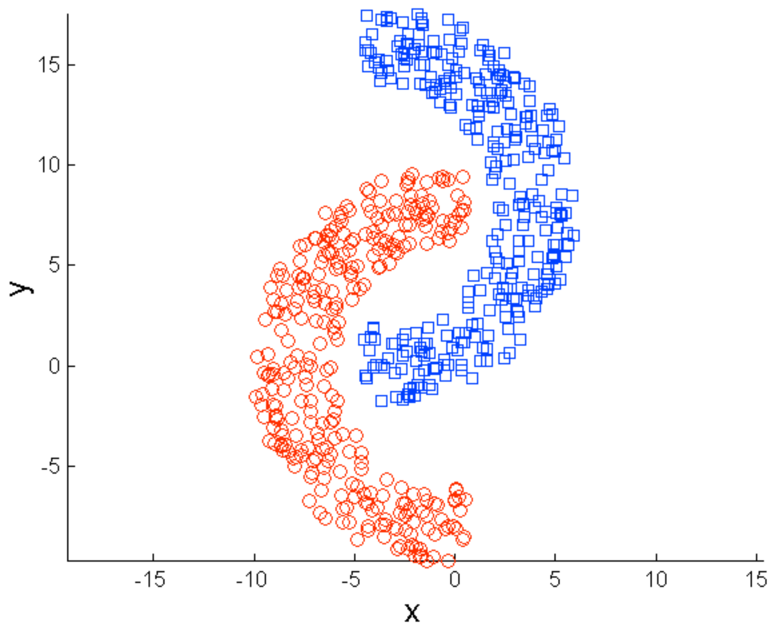


Original Points

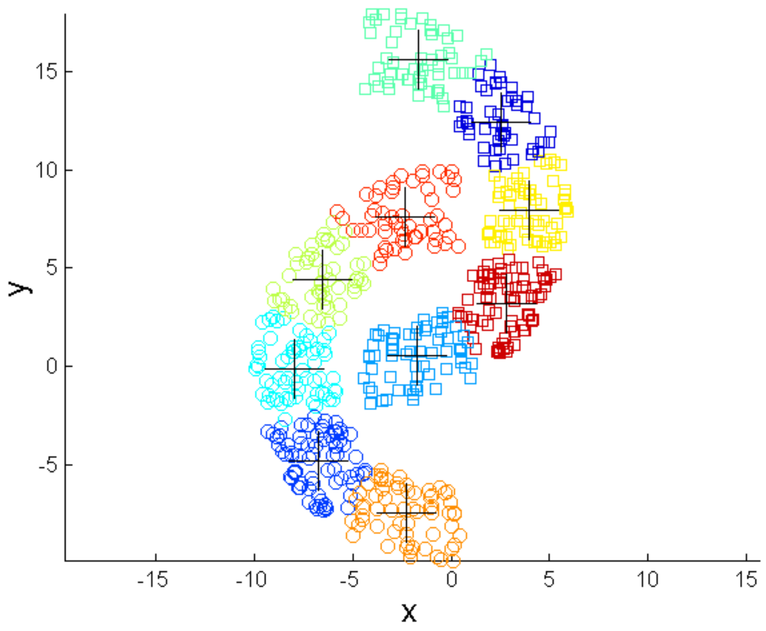


K-means Clusters

OVERCOMING K-MEANS LIMITATIONS



Original Points



K-means Clusters

Or use hierarchical or density based clustering

K-MEANS CLUSTERING: SKLEARN

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=8, n_init=10)
kmeans.fit(X)
print(kmeans.labels_)
```

- n_clusters: number of clusters
- n_init: number of times k-means is run