Data Mining

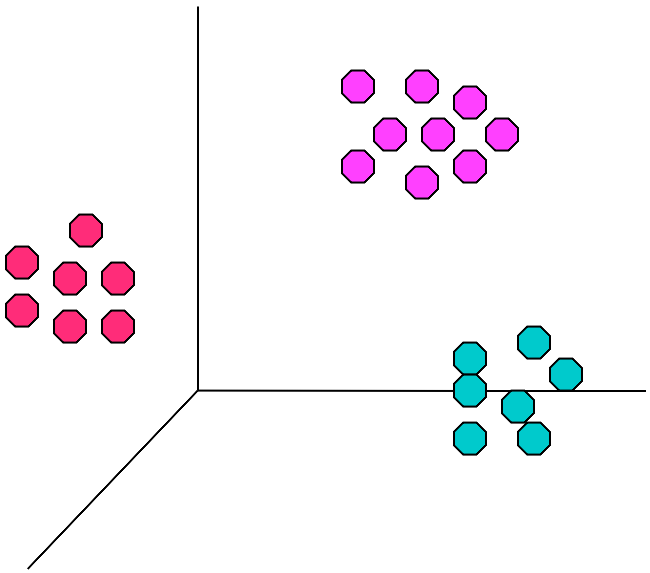
Introduction to Clustering

Mauro Sozio

some slides from Tan,Steinbach, Kumar, Introduction to Data Mining

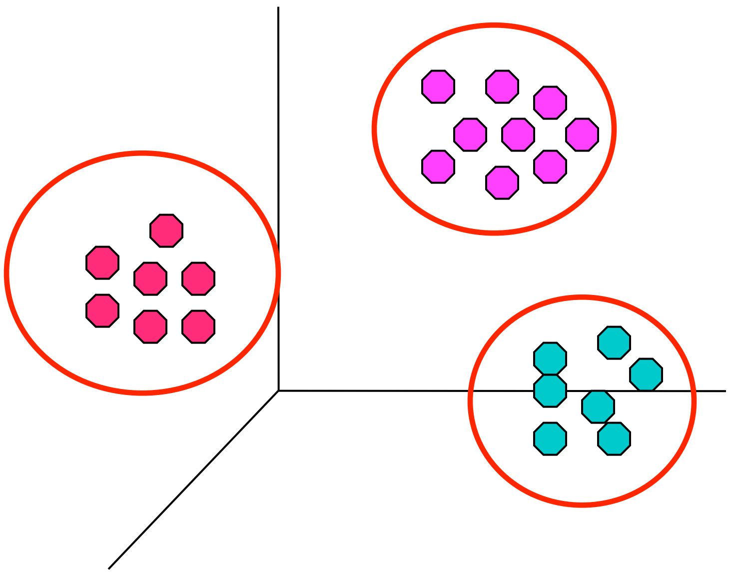
What is Cluster Analysis?

- 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



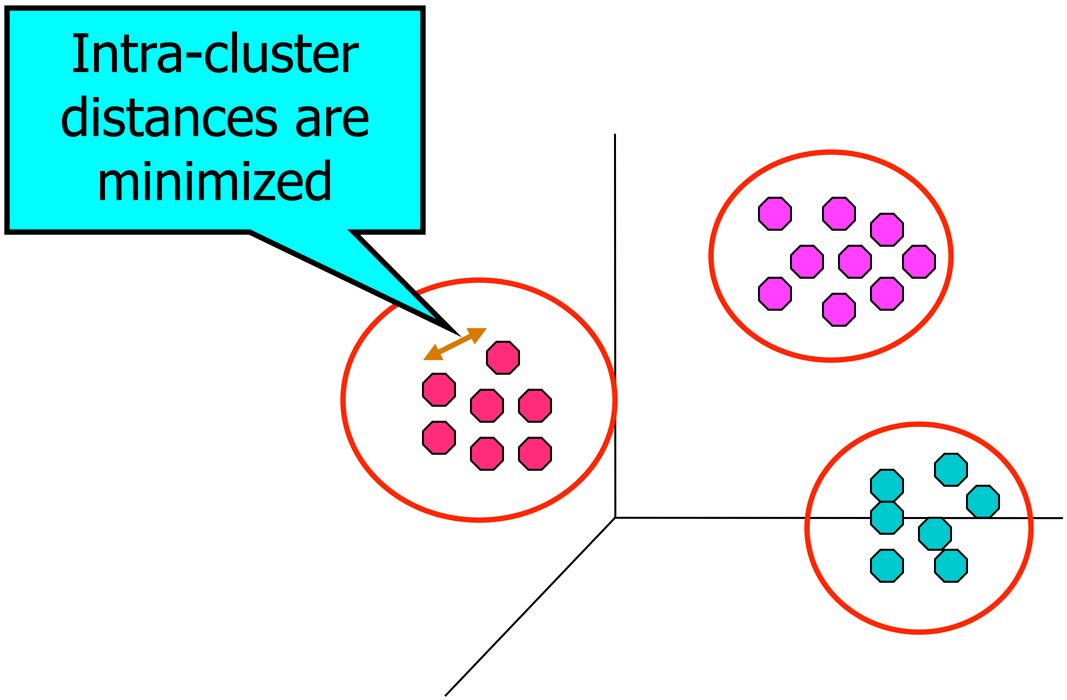
What is Cluster Analysis?

- 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



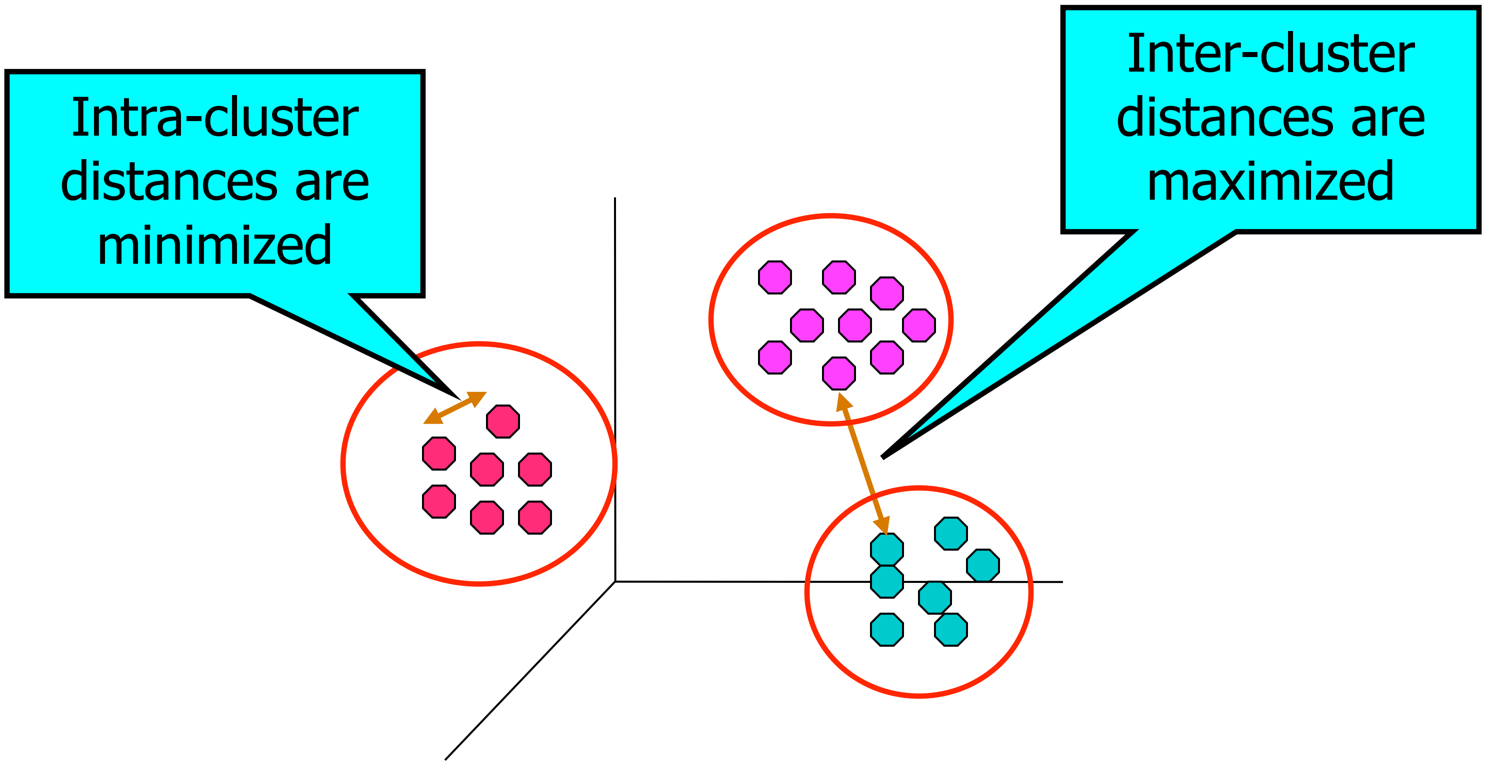
What is Cluster Analysis?

- 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



What is Cluster Analysis?

- 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

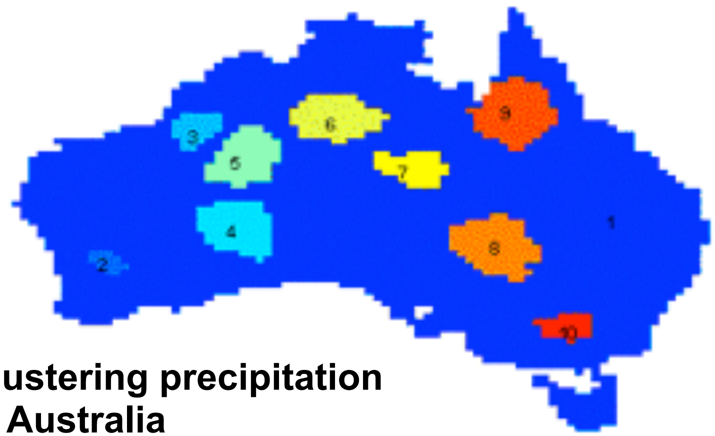


Applications of Cluster Analysis

- **Understanding**
  - Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations

	Discovered Clusters	*Industry Group*
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP

- **Summarization**
  - Reduce the size of large data sets



Clustering precipitation in Australia

What is not Cluster Analysis?

- Supervised classification
  - Have class label information
- Simple segmentation
  - Dividing students into different registration groups alphabetically, by last name
- Results of a query
  - Groupings are a result of an external specification
- Graph partitioning
  - Some mutual relevance and synergy, but areas are not identical

Notion of a Cluster can be Ambiguous

How many clusters?

Notion of a Cluster can be Ambiguous

How many clusters?

Two Clusters

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

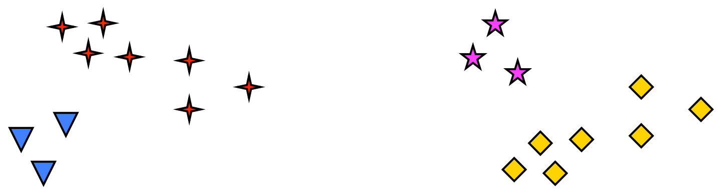
Two Clusters

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Two Clusters

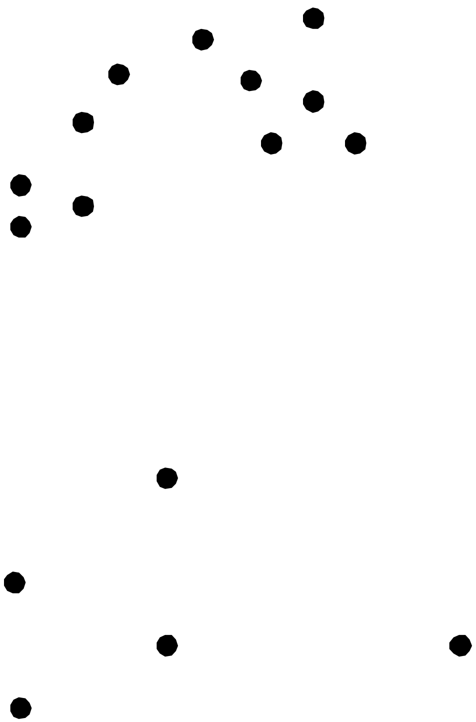


Four Clusters

Types of Clusterings

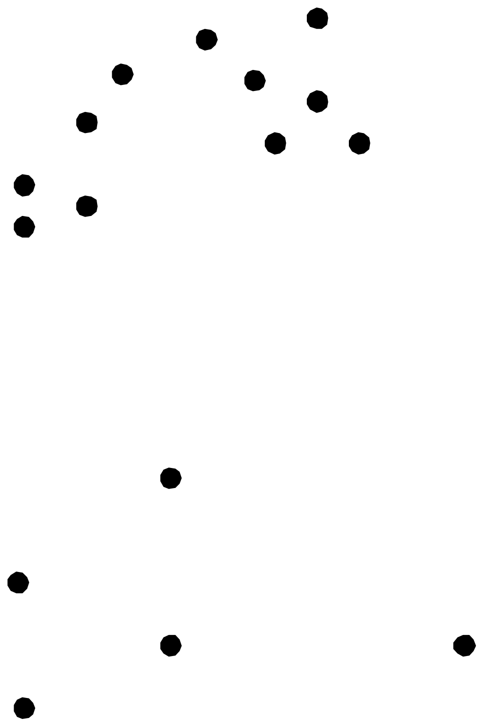
- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested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

Partitional Clustering

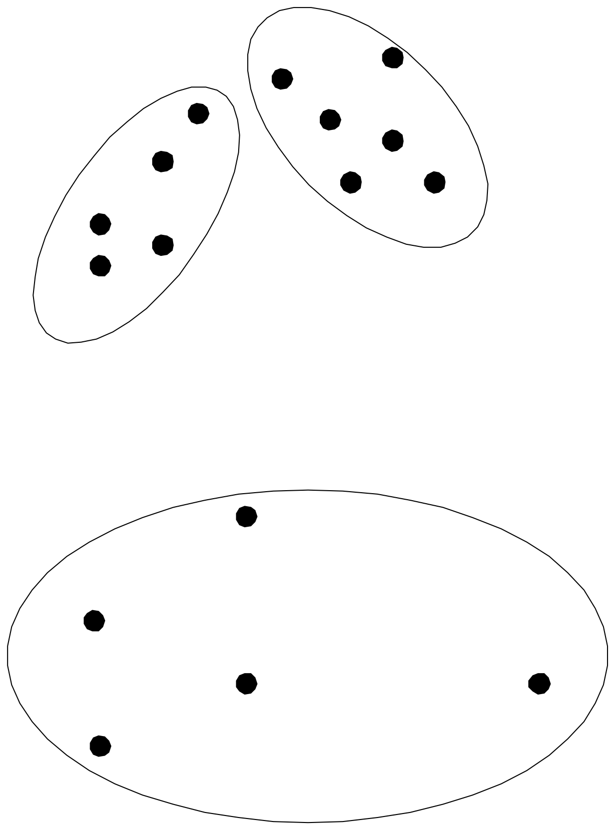


Original Points

Partitional Clustering

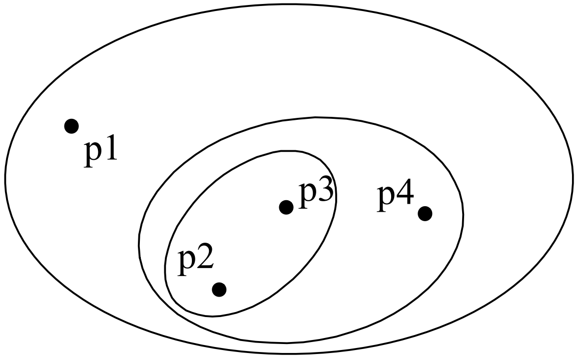


Original Points

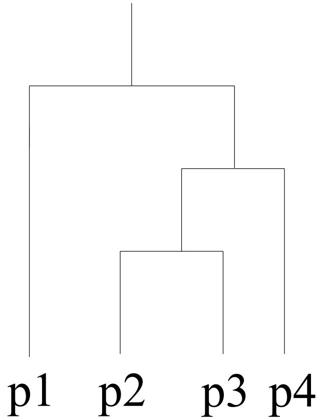


A Partitional Clustering

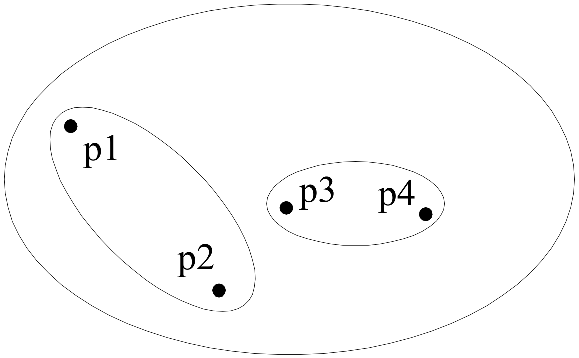
Hierarchical Clustering



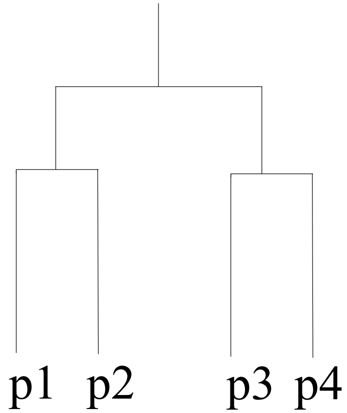
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Other Distinctions Between Sets of Clusters

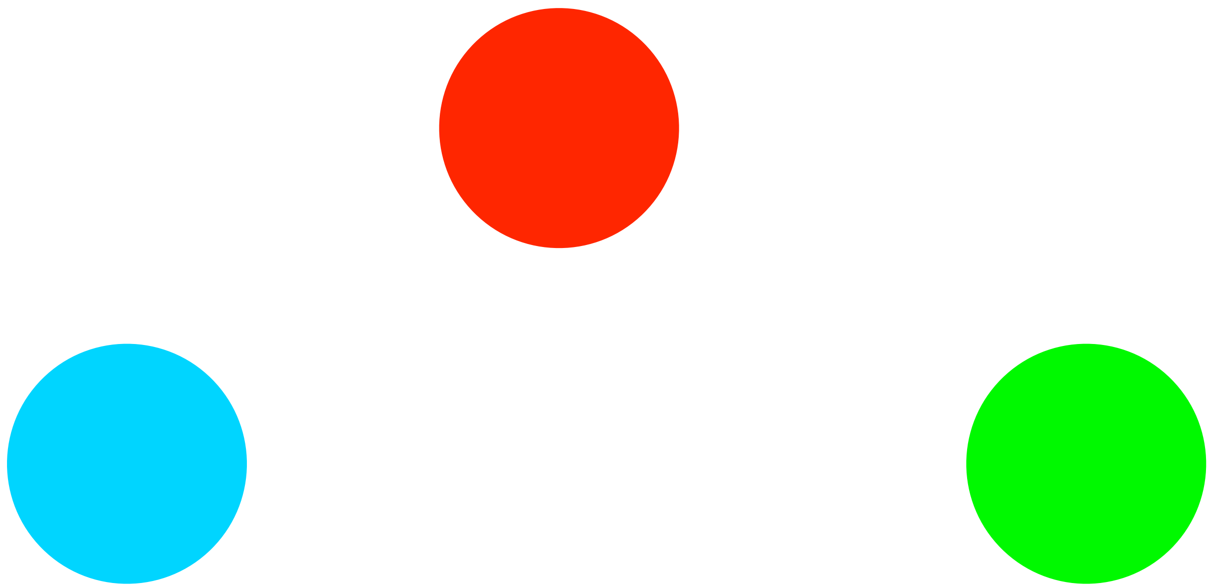
- Exclusive versus non-exclusive
  - In non-exclusive clusterings, points may belong to multiple clusters.
  - Can represent multiple classes or ‘border’ points
- Fuzzy versus non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - Probabilistic clustering has similar characteristics
- Partial versus complete
  - In some cases, we only want to cluster some of the data
- Heterogeneous versus homogeneous
  - Cluster of widely different sizes, shapes, and densities

Types of Clusters

- Well-separated clusters
- Center-based clusters
- Contiguous clusters
- Density-based clusters
- Property or Conceptual
- Described by an Objective Function

Types of Clusters: Well-Separated

- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

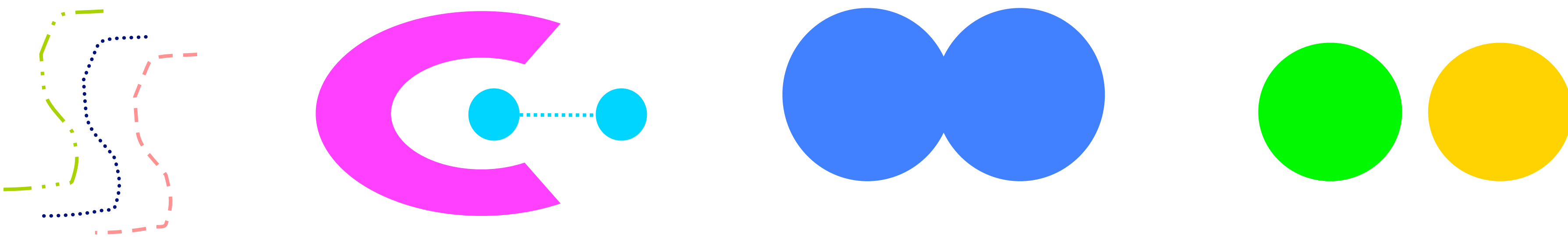
Types of Clusters: Center-Based

- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the “center” of a cluster, than to the center of any other cluster
  - The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most “representative” point of a cluster

4 center-based clusters

Types of Clusters: Contiguity-Based

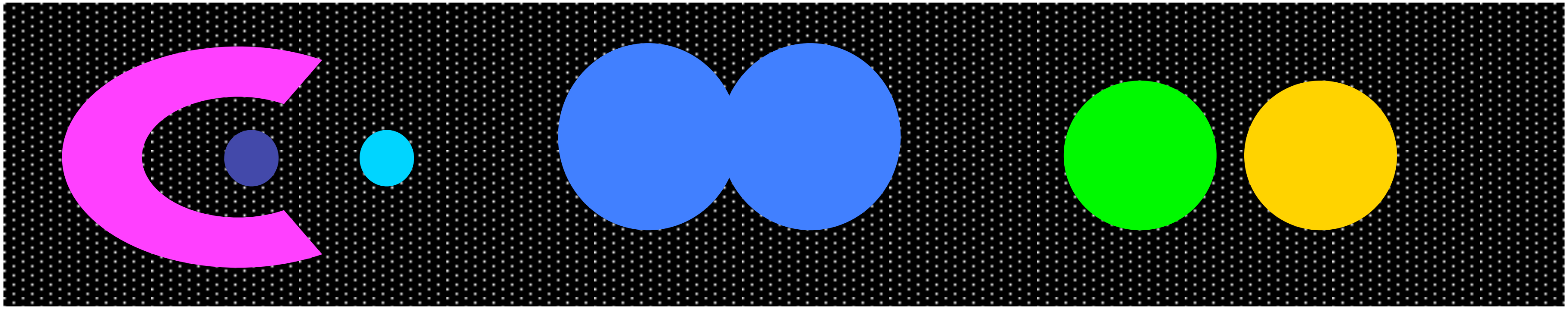
- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



8 contiguous clusters

Types of Clusters: Density-Based

- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Used when the clusters are irregular or intertwined, and when noise and outliers are present.

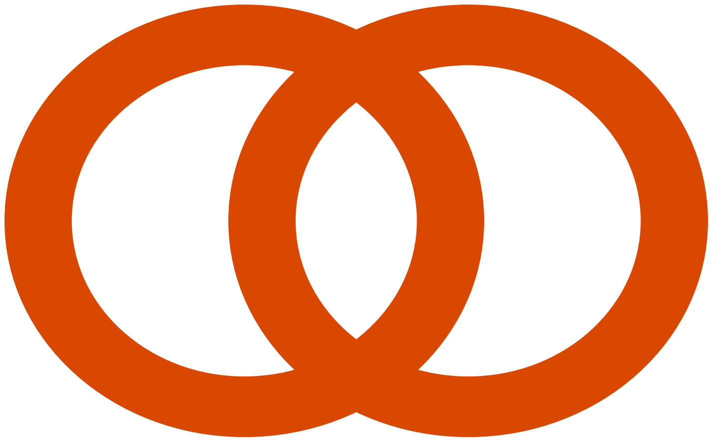


6 density-based clusters

Types of Clusters: Conceptual Clusters

- Shared Property or Conceptual Clusters
  - Finds clusters that share some common property or represent a particular concept.

.



2 Overlapping Circles

Clustering Algorithms

- K-means
- K-means++
- Hierarchical clustering

K-means Clustering

Input: integer k>0, set S of points in the euclidean space
Output: A (partitional) clustering of S

1. Select k points in S as the initial centroids
2. Repeat until the centroids do not change
   Form k clusters by assigning points to the closest centroids
   For each cluster recompute its centroid

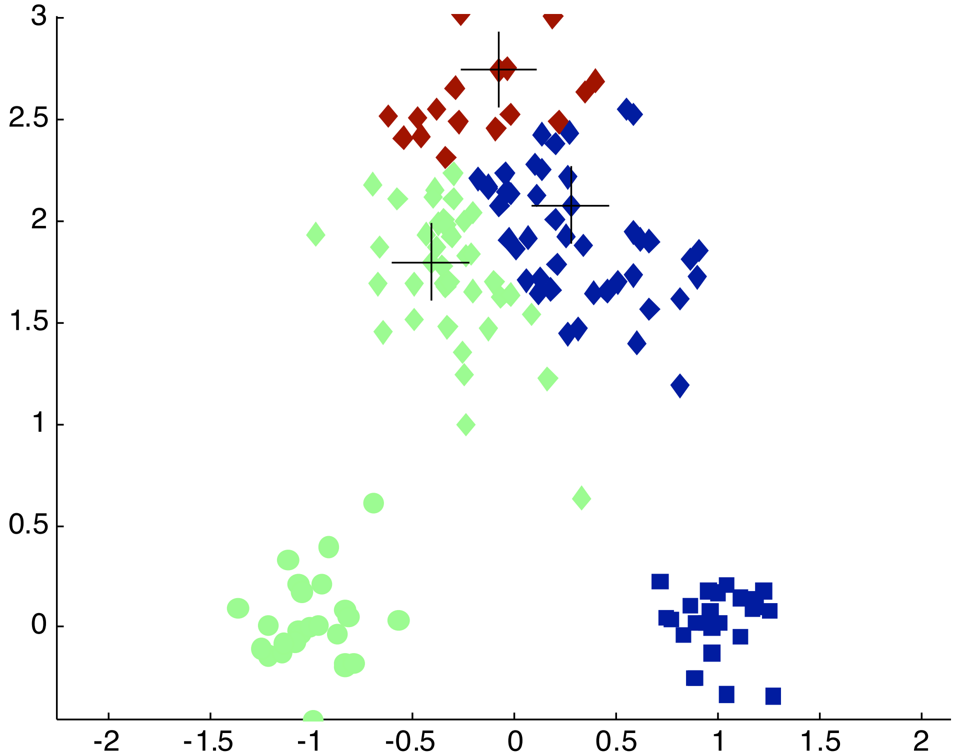
K-means Clustering

Input: integer k>0, set S of points in the euclidean space
Output: A (partitional) clustering of S

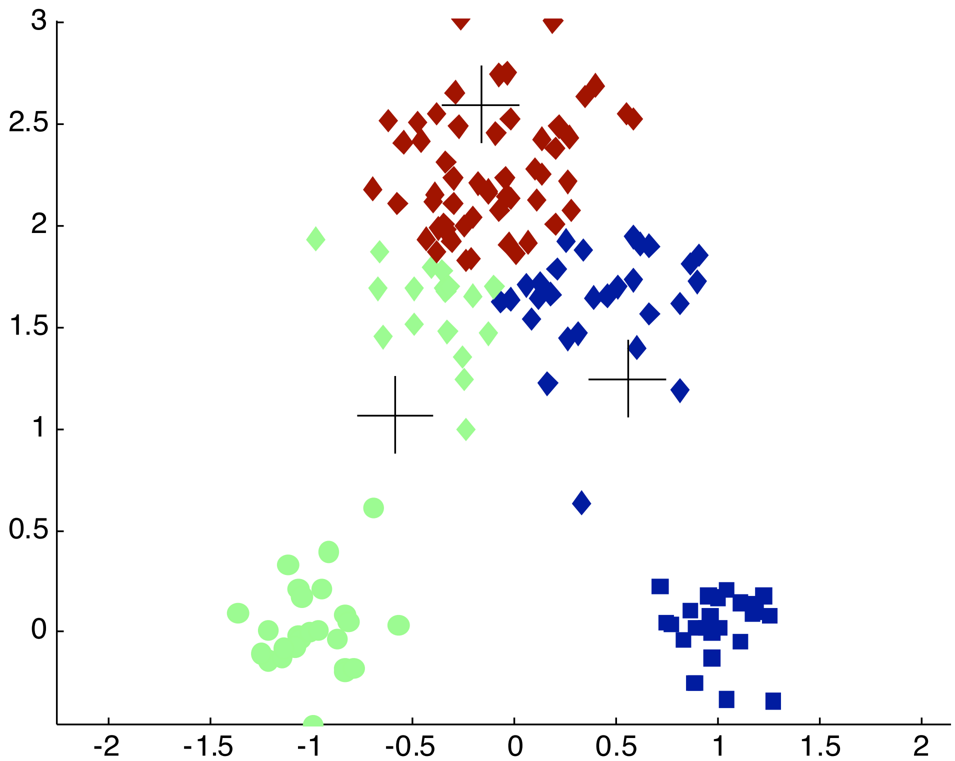
1. Select k points in S as the initial centroids
2. Repeat until the centroids do not change
   Form k clusters by assigning points to the closest centroids
   For each cluster recompute its centroid

- Initial centroids are often chosen randomly.
- Centroids are often the mean of the points in the cluster.
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.

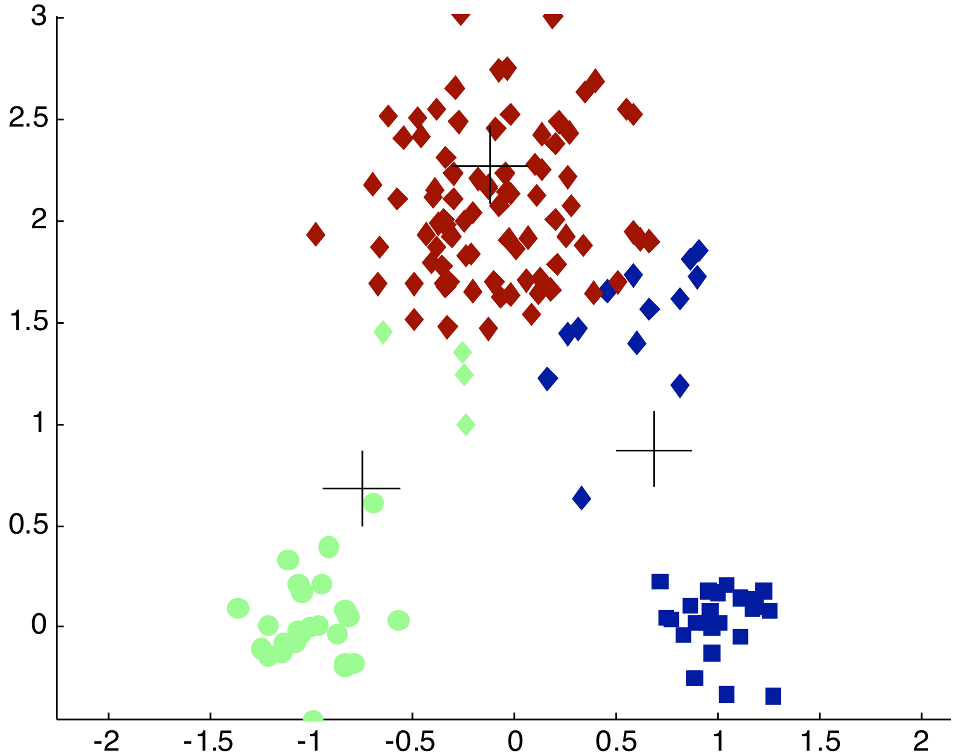
K-means: example



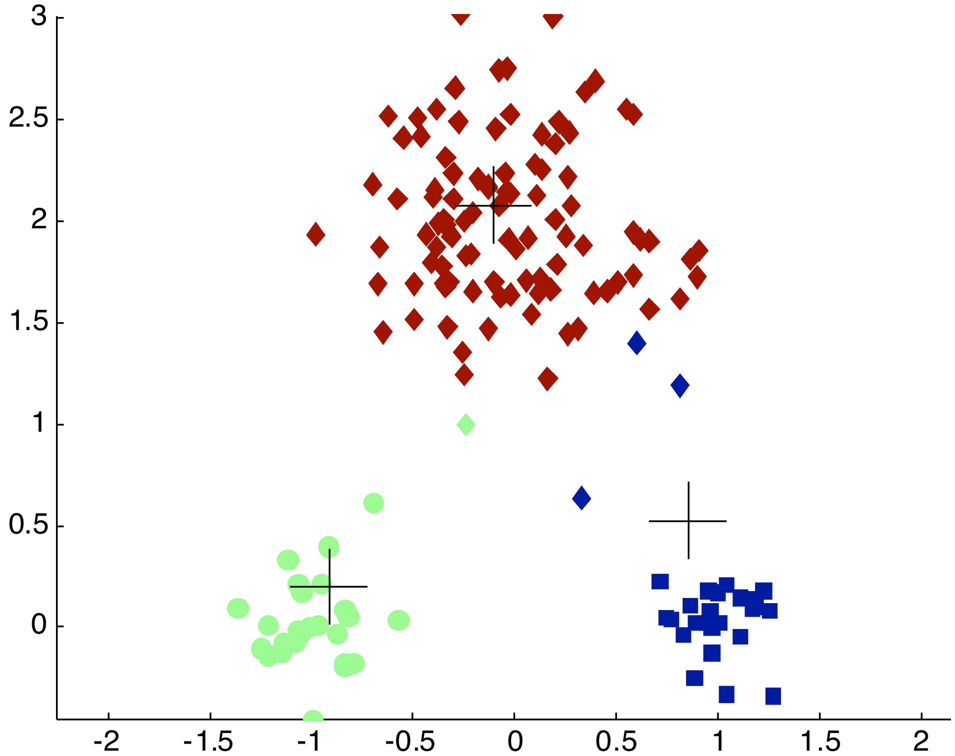
K-means: example



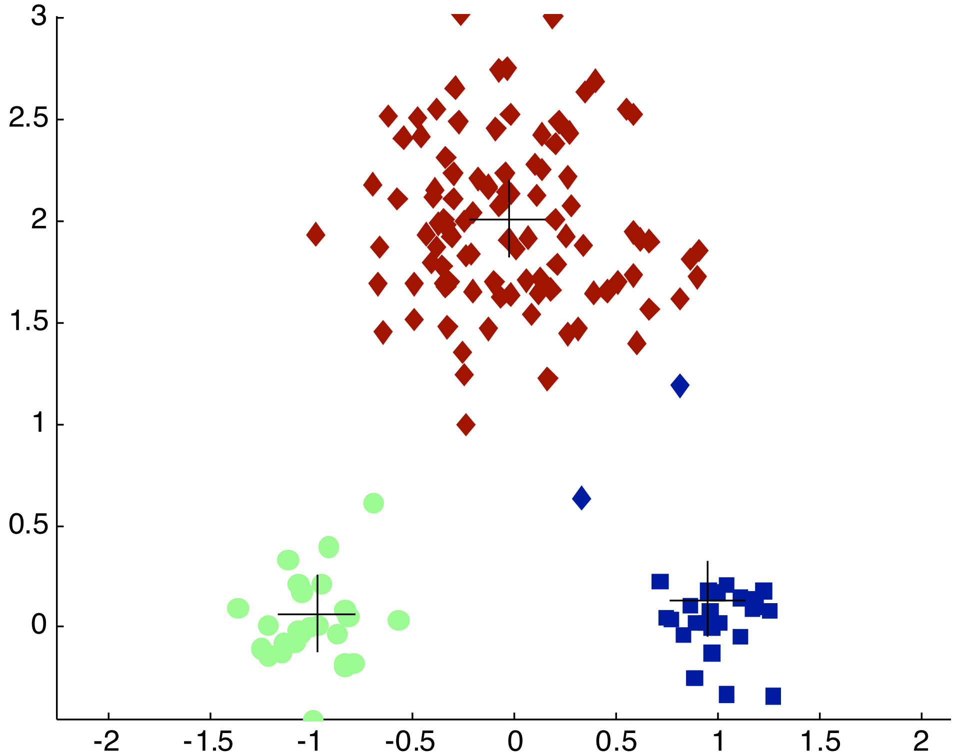
K-means: example



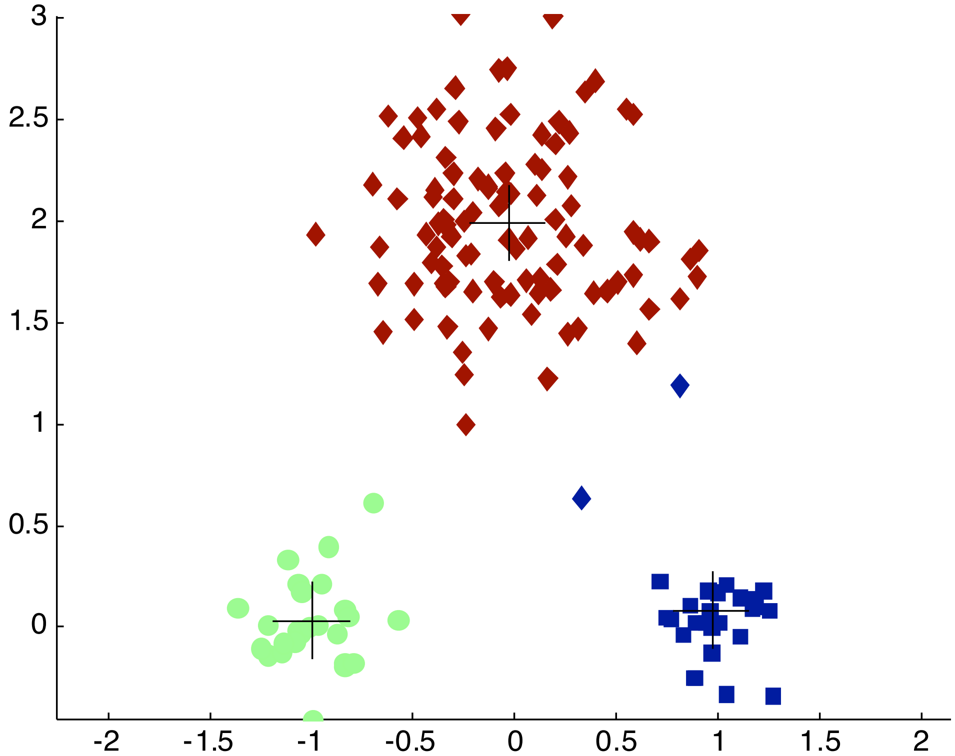
K-means: example



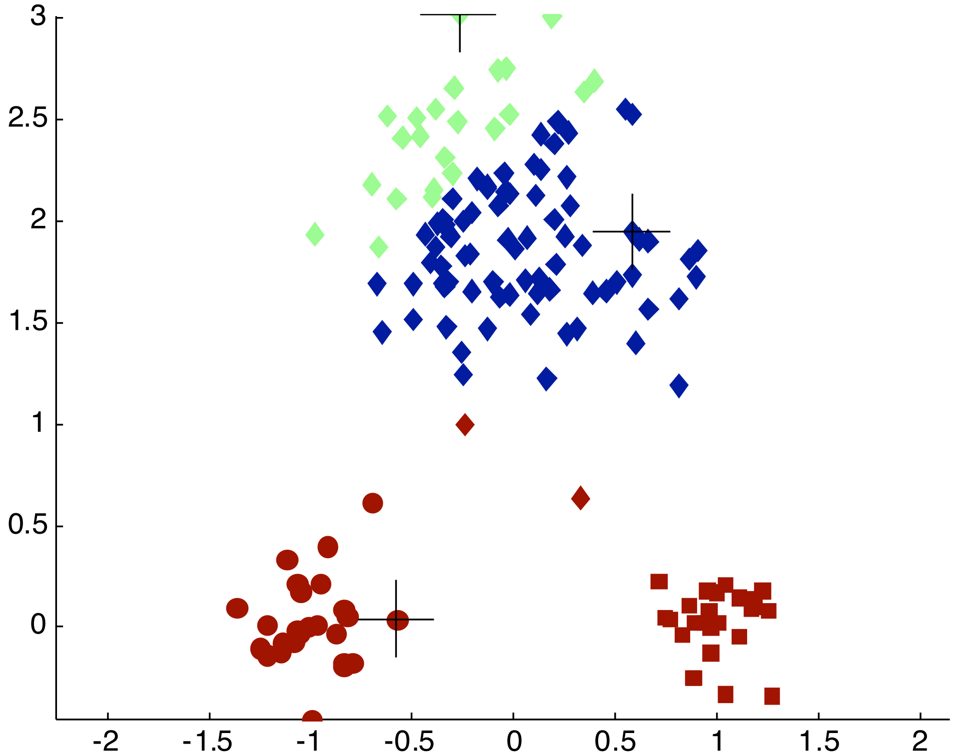
K-means: example



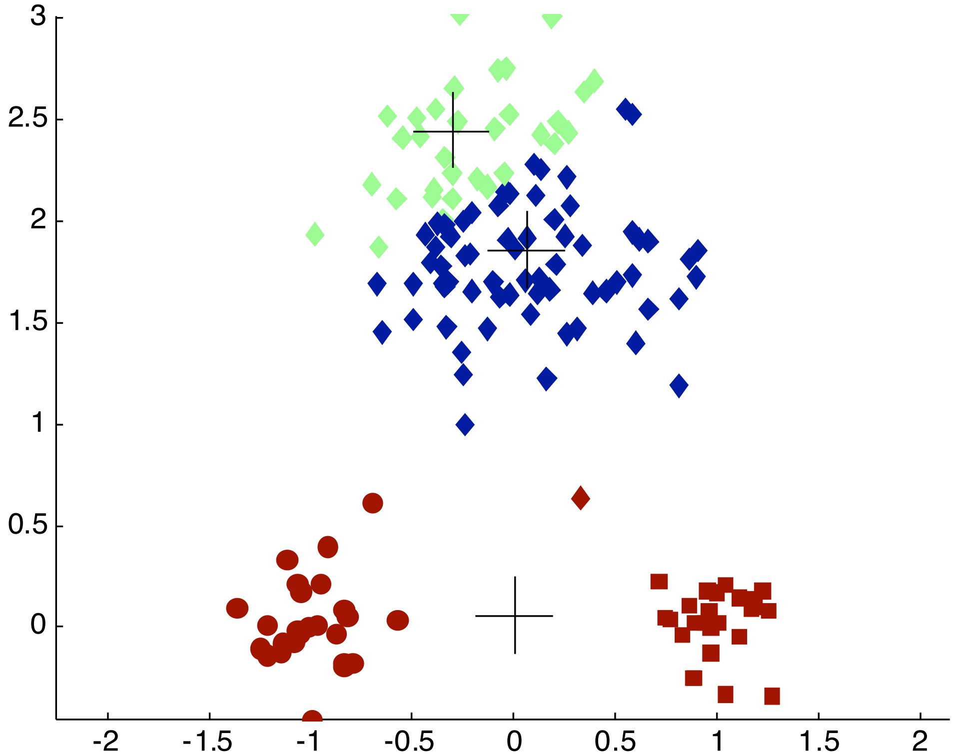
K-means: example



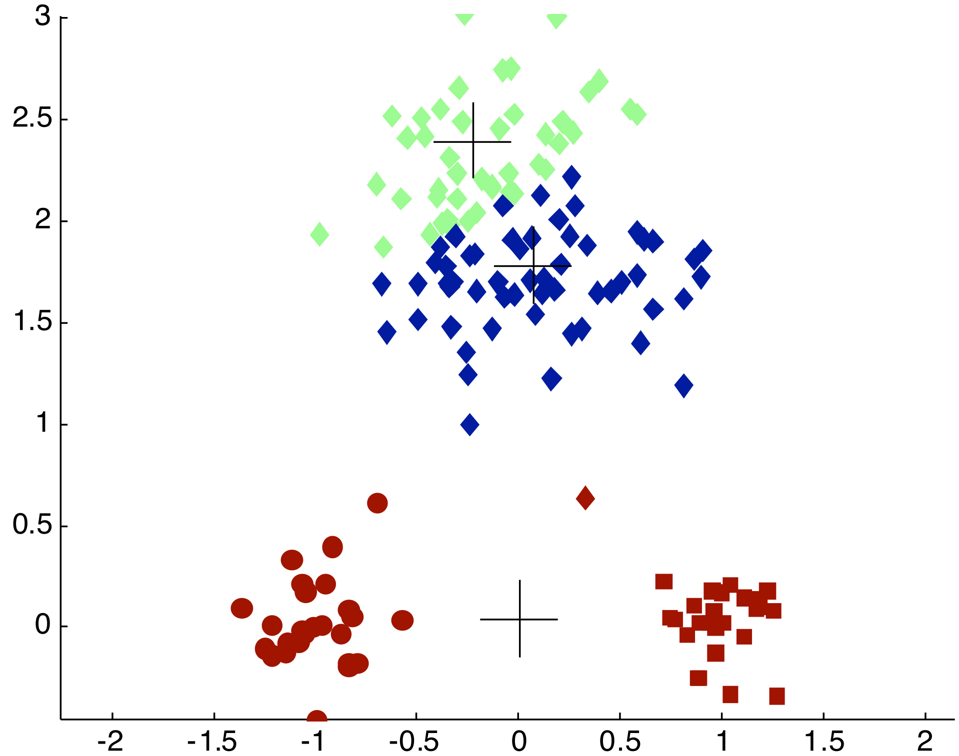
Importance of Choosing Initial Centroids …



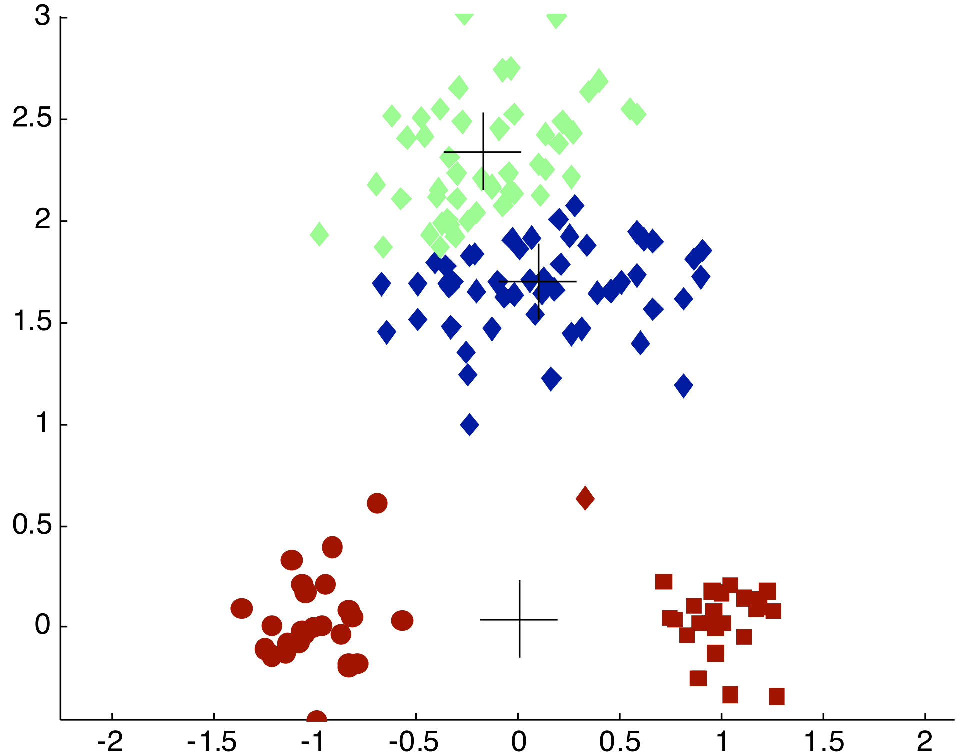
Importance of Choosing Initial Centroids …



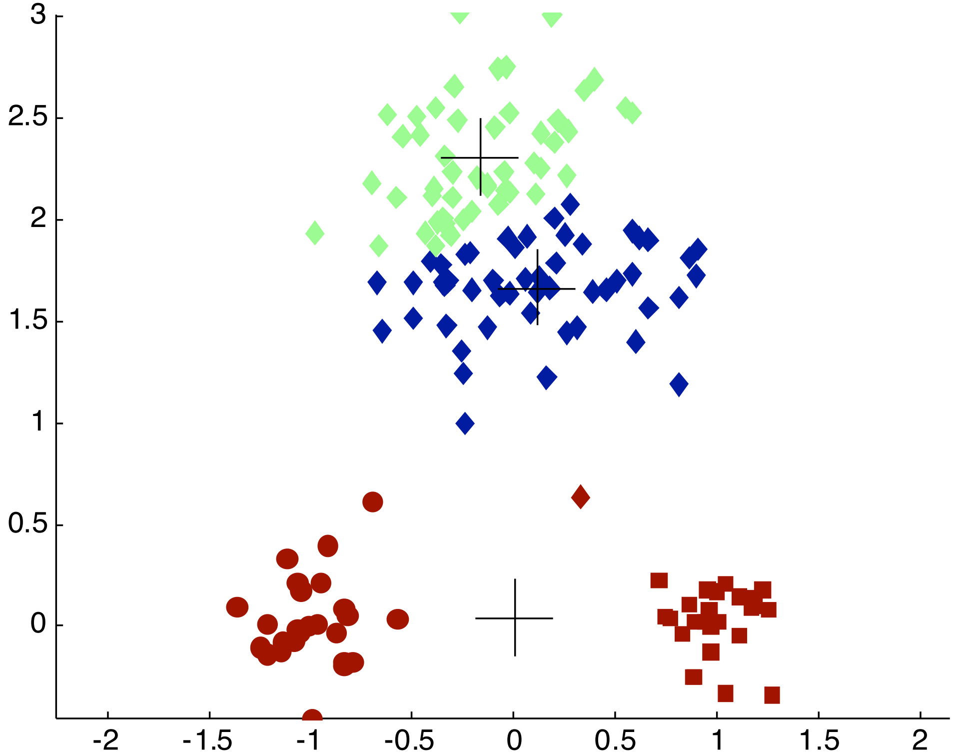
Importance of Choosing Initial Centroids …



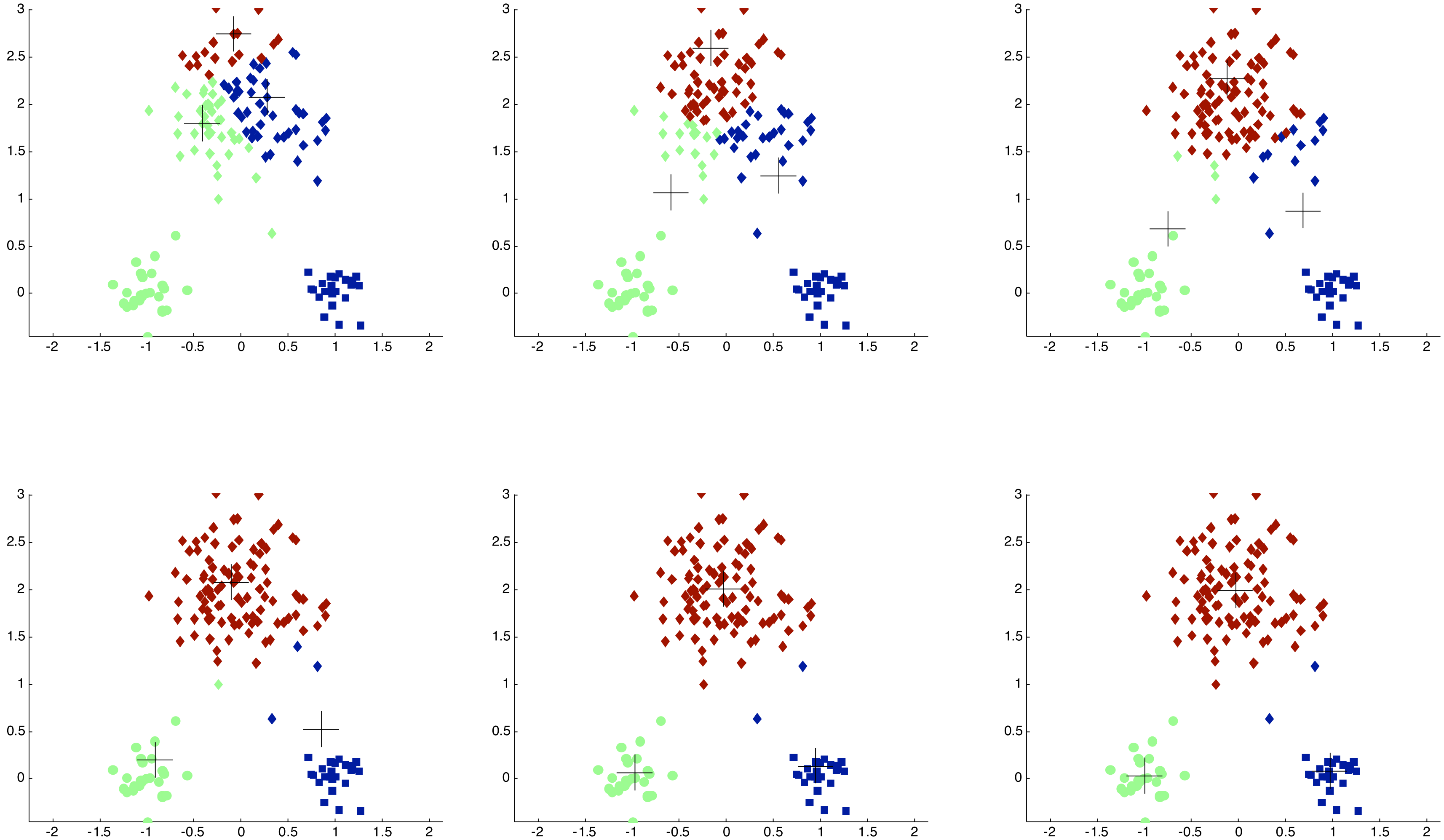
Importance of Choosing Initial Centroids …



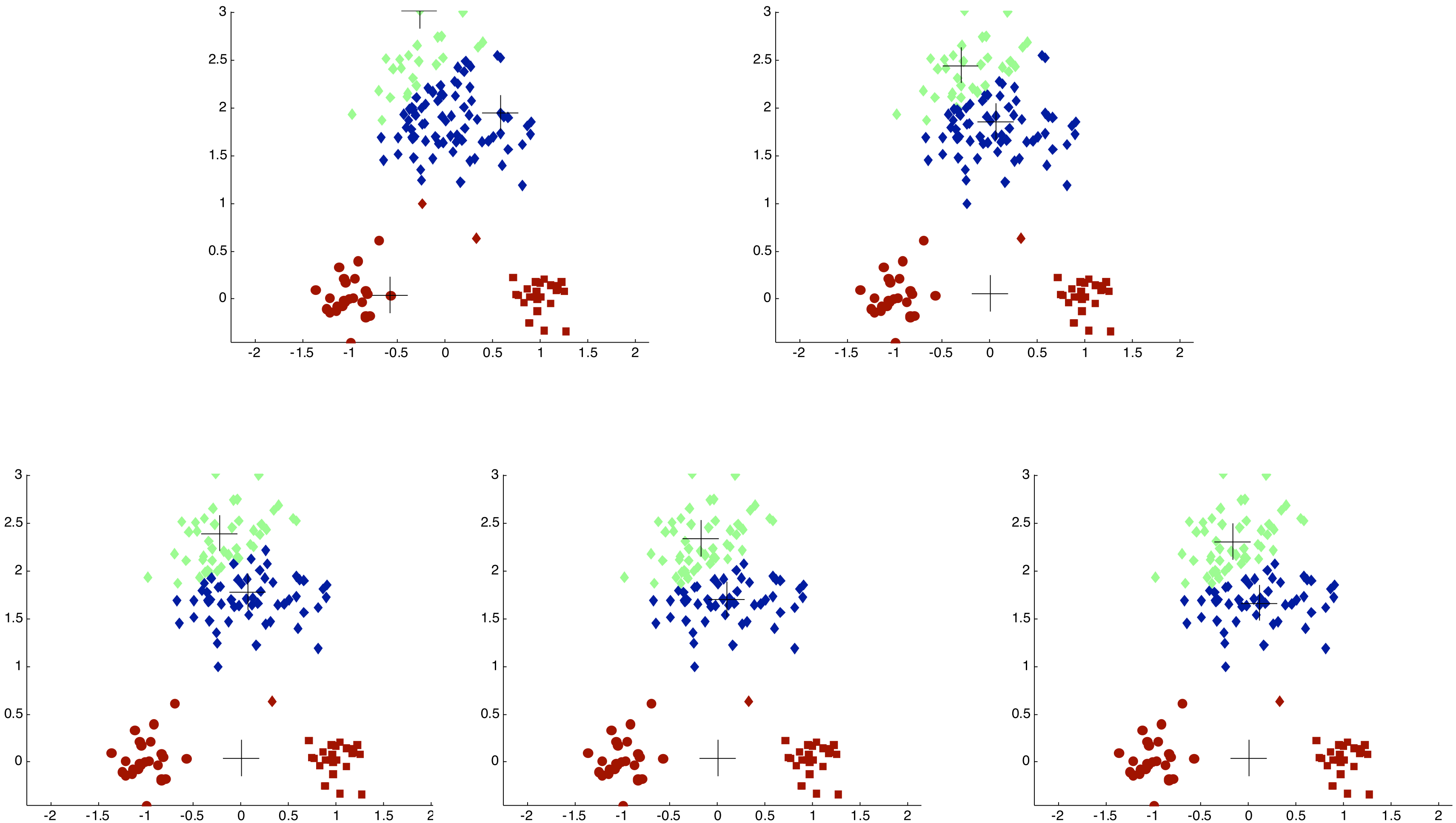
Importance of Choosing Initial Centroids …



Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids ...



Problems with Selecting Initial Points

- **Input**: k clusters with the same size n/k, clusters are ``sufficiently'' spread apart.
- Need to select 1 point per cluster!
- prob. the 2nd point is in a new cluster is (k-1)/k, the 3rd point in a new cluster is (k-2)/k...
- Prob. to select exactly one point per cluster =

$$\frac{k-1}{k} \cdot \frac{k-2}{k} \cdot \quad \dots \quad \cdot \frac{1}{k} = \frac{(k-1)!}{k^{k-1}} = \frac{k!}{k^k}$$

For example, if K = 10, then probability = $10!/10^{10}$ = 0.00036.

Evaluating K-means Clusterings

- Most common measure is Sum of Squared Error (SSE):

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

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

- Given two clusterings, we can choose the one with smallest error

- Decreasing K might decrease SSE. However, good clusterings with small K might have a lower SSE than poor clusterings with higher K.

K-Means Always Terminates

- **Theorem.** K-means with Euclidean distance as distance always terminates.
- Proof follows from the following lemmas.

- **Lemma 1**. The point y that minimizes the SSE in a cluster C is the mean of all points in C.

- **Lemma 2**. SSE strictly decreases.

- **Lemma 3**. The total number of possible clusterings is finite (< n^k).

K-Means Always Terminates

- **Lemma 1**. Given a cluster C the point y that minimizes the SSE in cluster C is the mean of all points in C.

- **Proof**.

$$SSE(C) = \sum_{\vec{x} \in C} d^2(\vec{y}, \vec{x}) = \sum_{\vec{x} \in C} \sum_{j=1}^{m} (y_j - x_j)^2$$

$$\frac{\partial SSE(C)}{\partial y_j} = \sum_{\vec{x} \in C} 2(y_j - x_j)$$

$$\frac{\partial SSE(C)}{\partial y_j} = 0 \implies y_j = \frac{1}{|C|} \sum_{x \in C} x_j.$$

K-Means Always Terminates

- **Lemma 2**. Let SSE(t) be the SSE at step t. We have: SSE(t+1) < SSE (t).
- **Proof (Sketch).** At step t+1, each point x is assigned to a new cluster C only if the distance between x and the centroid of C strictly decreases.
- The next step is to recompute the centroids which cannot increase the total SSE (follows from Lemma 1).

K-Means Always Terminates

- **Theorem.** K-means always terminates.

- The proof follows from Lemma 2 and Lemma 3. We cannot obtain the same clustering more than once, otherwise we get the same SSE value.

- Observe that we need both Lemma 2 and 3.

Solutions to Initial Centroids Problem

- Multiple runs (helps but low success probability)
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
- Postprocessing
- K-Means++

Handling Empty Clusters

- Basic K-means algorithm can yield less than k clusters (so called empty clusters). (**Exercise**)

- Several strategies:
  - Pick the points that contributes most to SSE and move them to empty cluster.
  - Pick the points from the cluster with the highest SSE
  - If there are several empty clusters, the above can be repeated several times.

Updating Centers Incrementally

- In the basic K-means algorithm, centroids are updated after all points are assigned to a centroid.
- An alternative is to update the centroids after each assignment (incremental approach).

- More precisely, let $C_1,C_2,...,C_k$ be the current clusters. Re-assign all points one by one to the best cluster. Let p in C_i be the current point and suppose we re-assign it to C_j. Then, after that, recompute the centroid of C_i and C_j.

  - \+ Never get an empty cluster
  - \- Introduces an order dependency
  - \- More expensive

Pre-processing and Post-processing

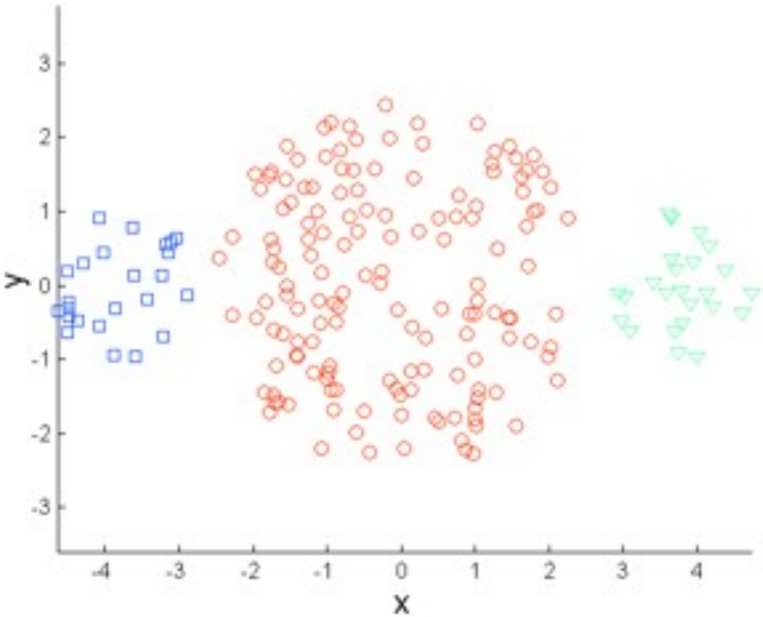
- Pre-processing
  - Normalize the data
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE

Limitations of K-means

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

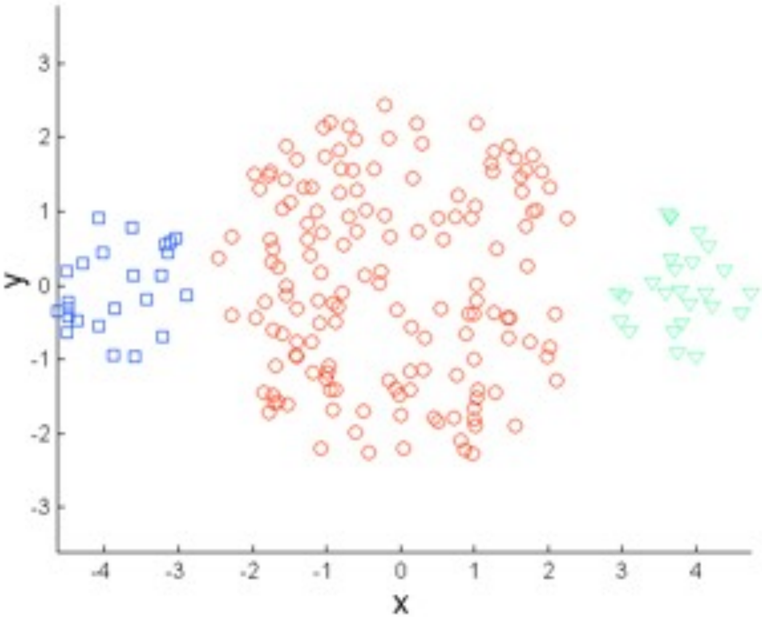
- K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes



Original Points

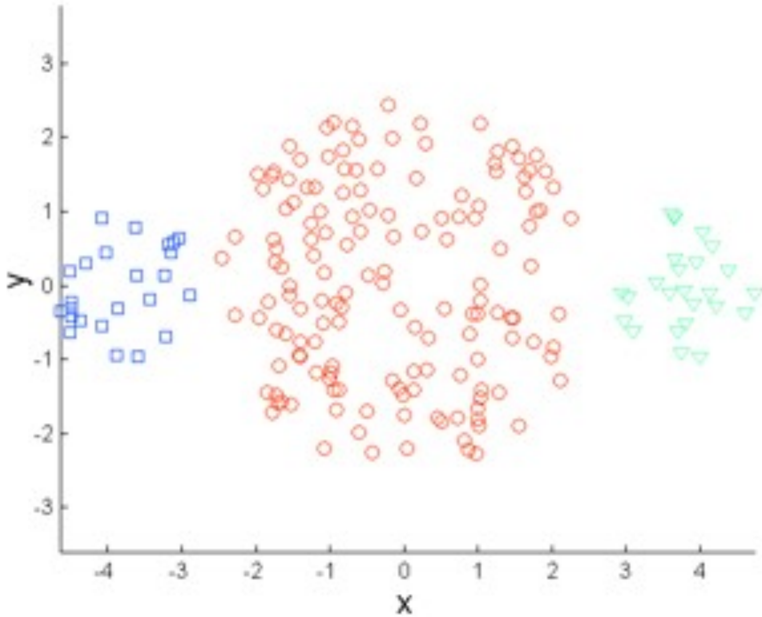
Limitations of K-means: Differing Sizes



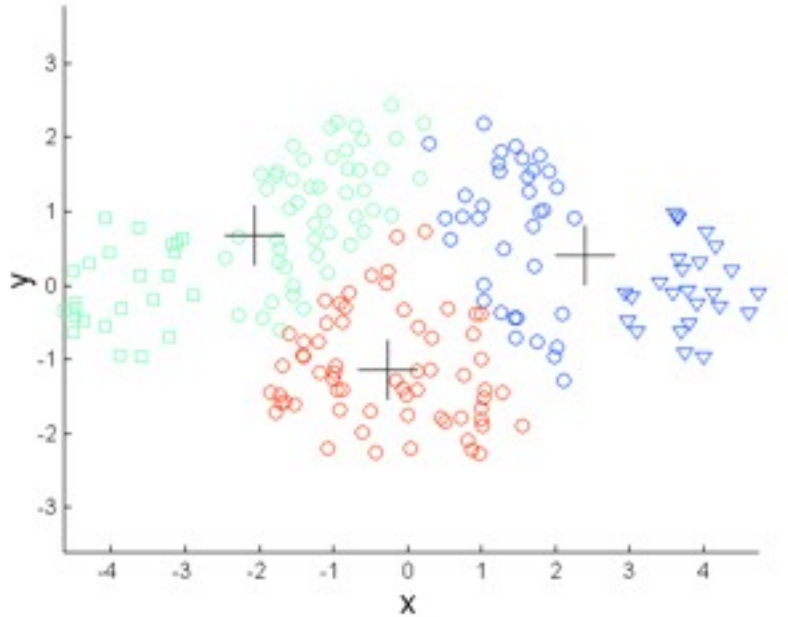
Original Points

K-means (3 Clusters)

Limitations of K-means: Differing Sizes

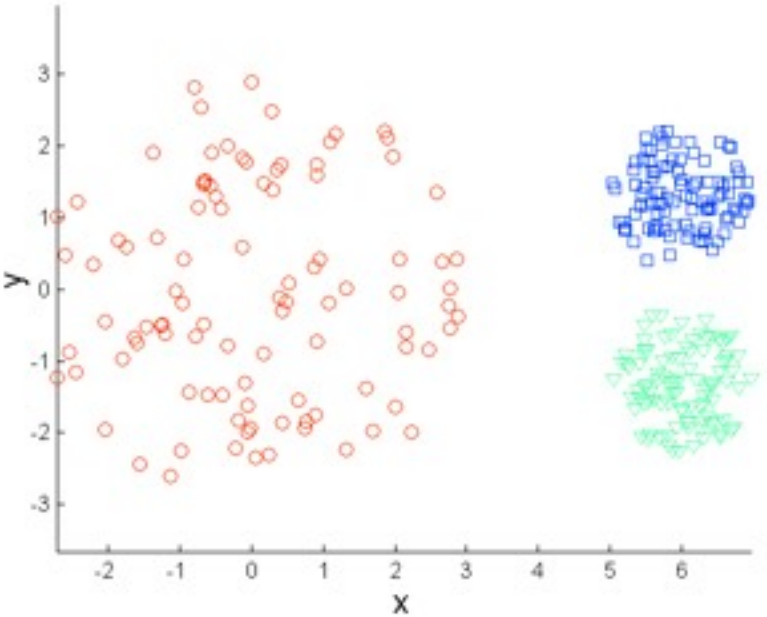


Original Points



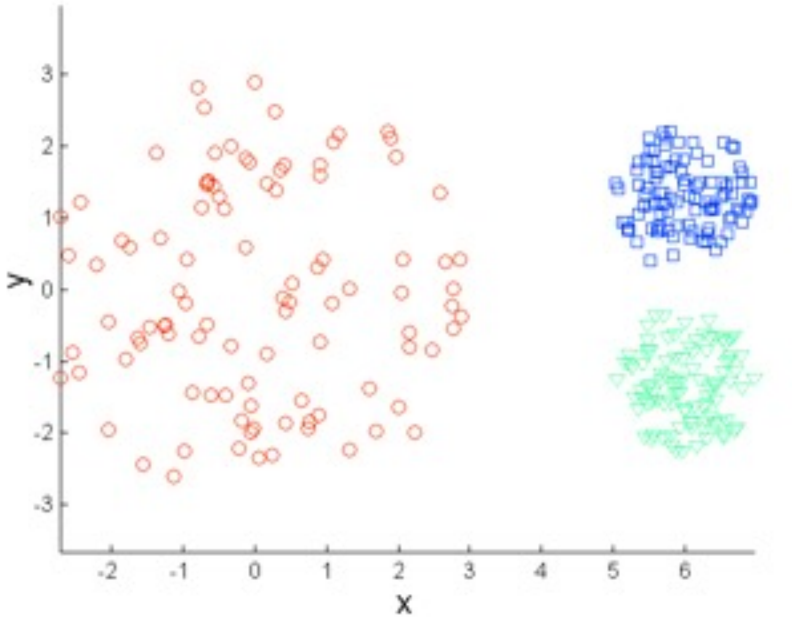
K-means (3 Clusters)

Limitations of K-means: Differing Density



Original Points

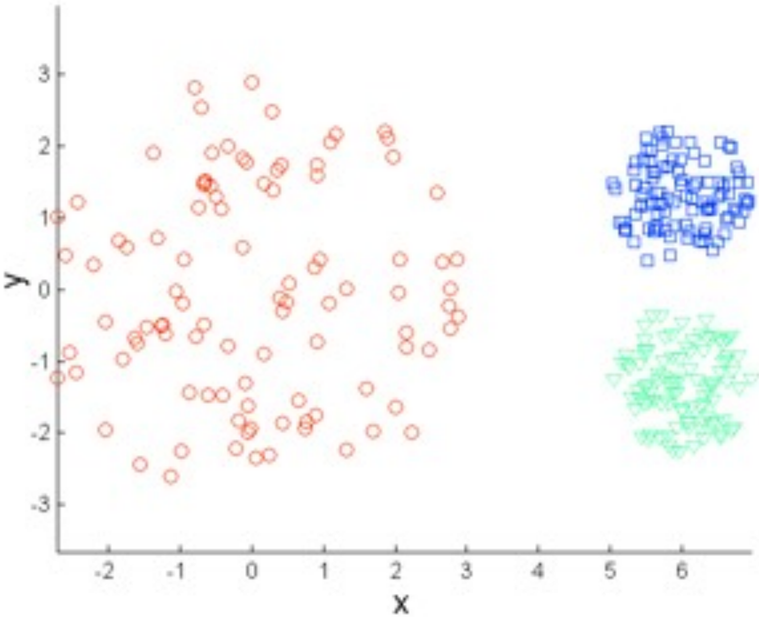
Limitations of K-means: Differing Density



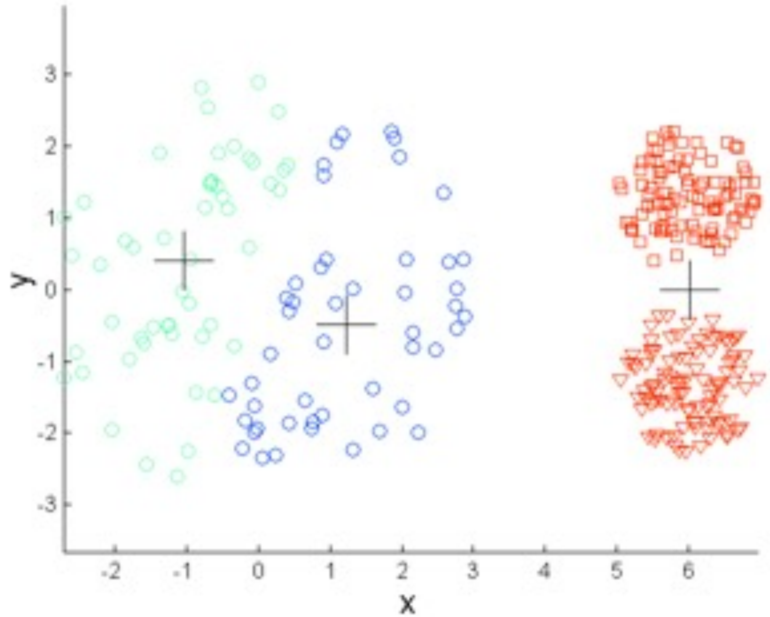
Original Points

K-means (3 Clusters)

Limitations of K-means: Differing Density

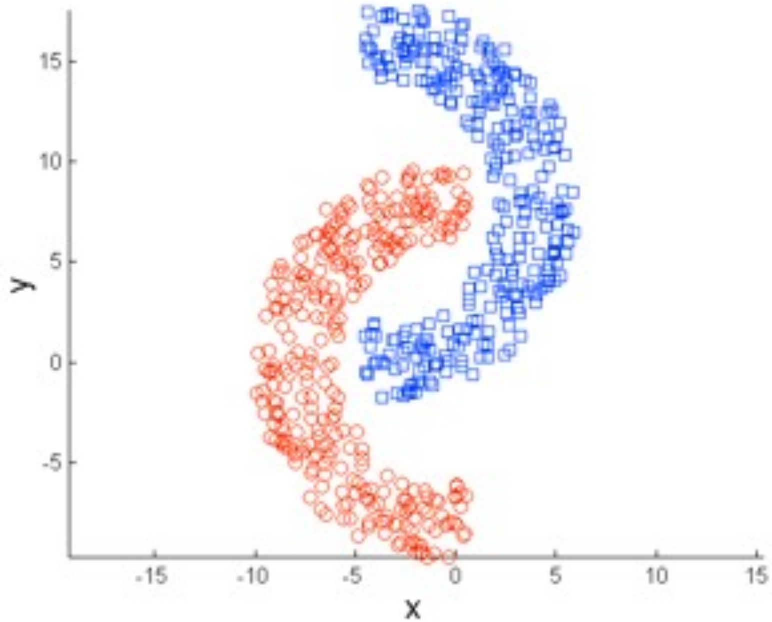


Original Points



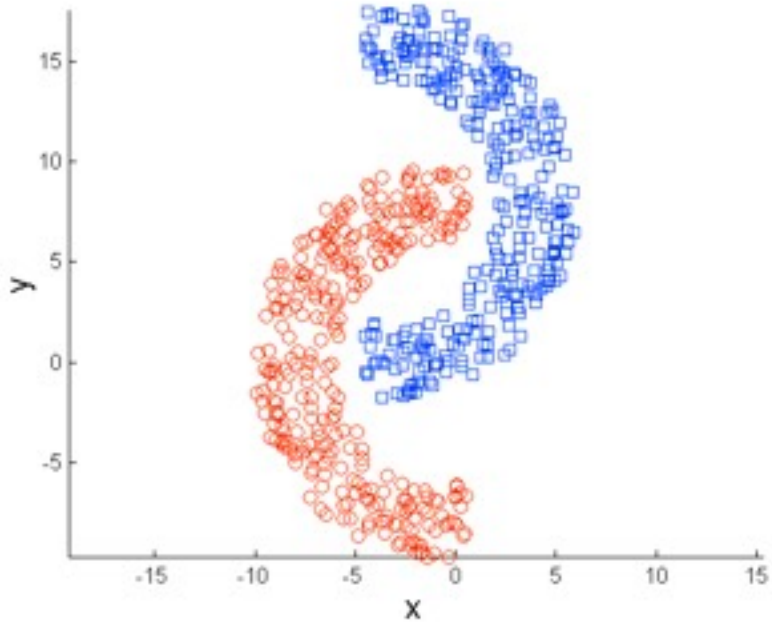
K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



Original Points

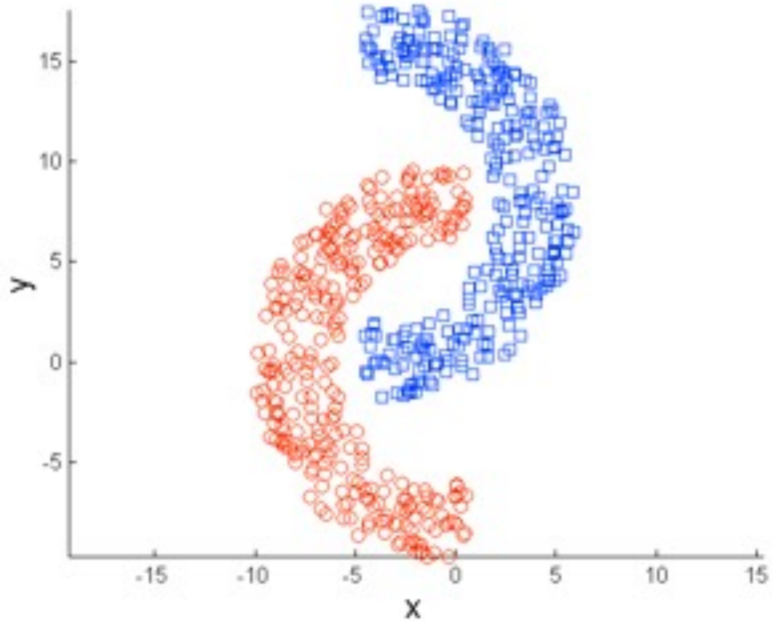
Limitations of K-means: Non-globular Shapes



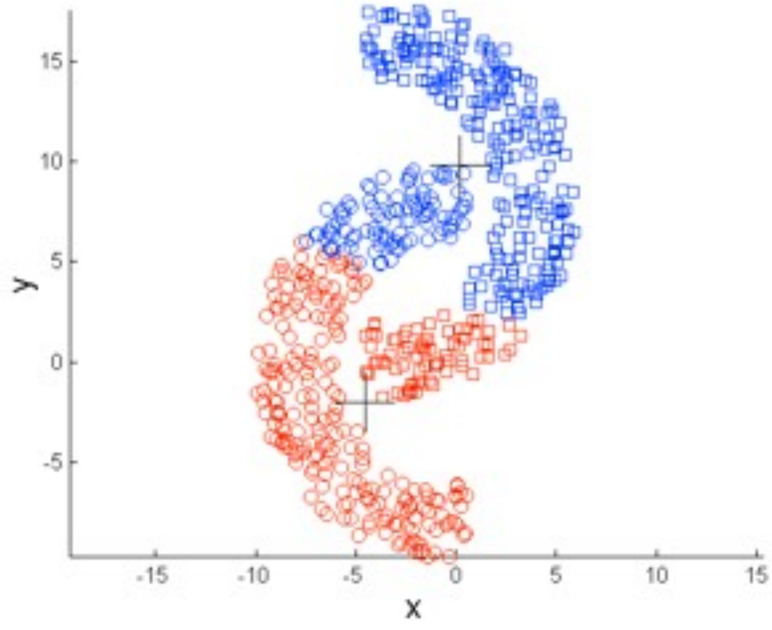
Original Points

K-means (2 Clusters)

Limitations of K-means: Non-globular Shapes

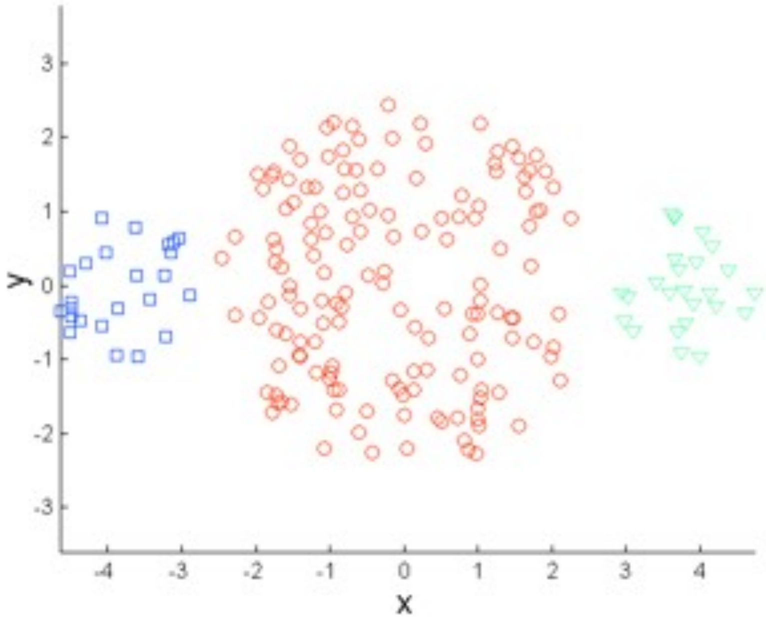


Original Points

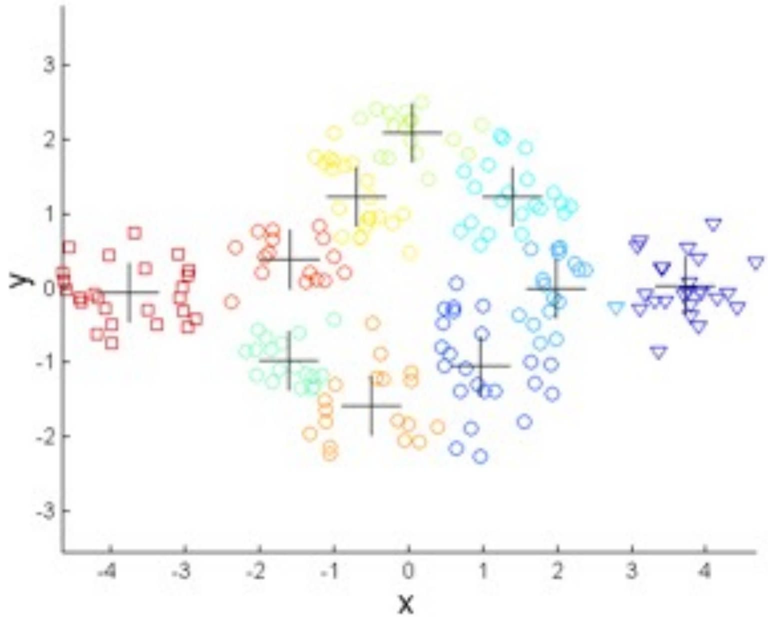


K-means (2 Clusters)

Overcoming K-means Limitations



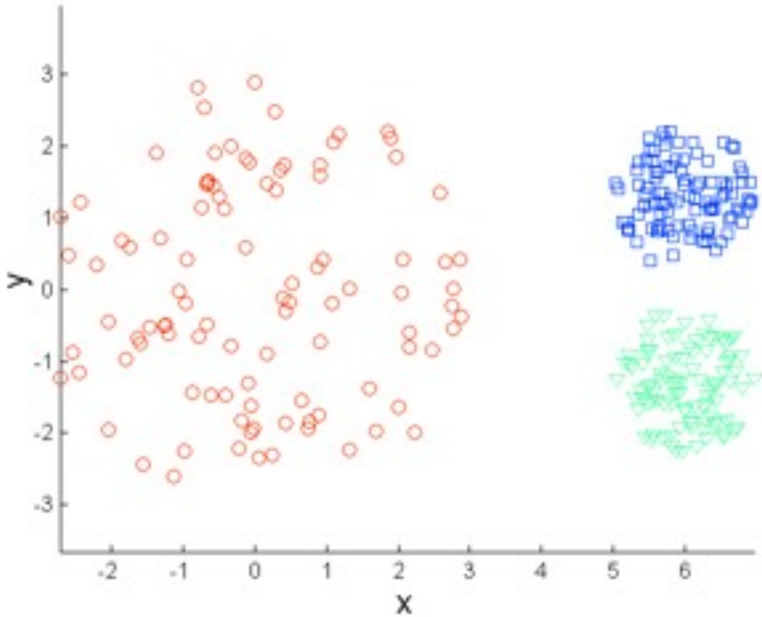
Original Points



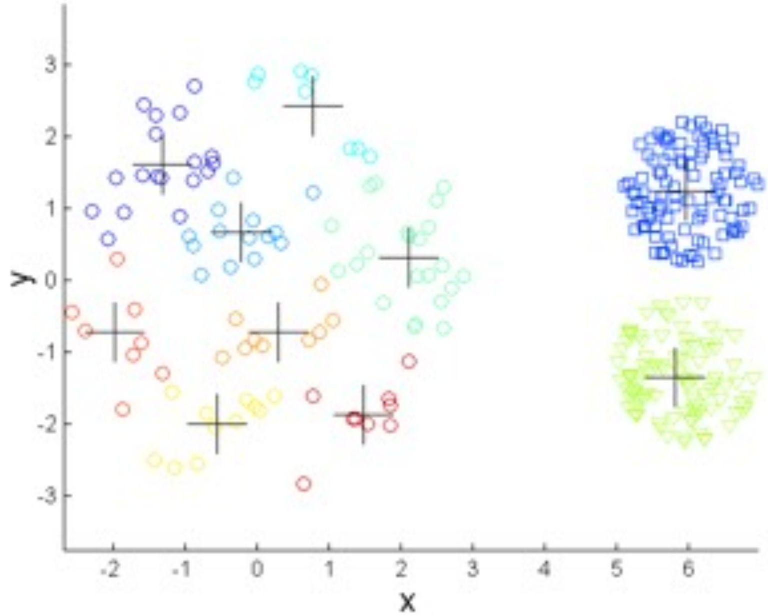
K-means Clusters

One solution is to use many clusters.
Find parts of clusters, but need to put together.

Overcoming K-means Limitations

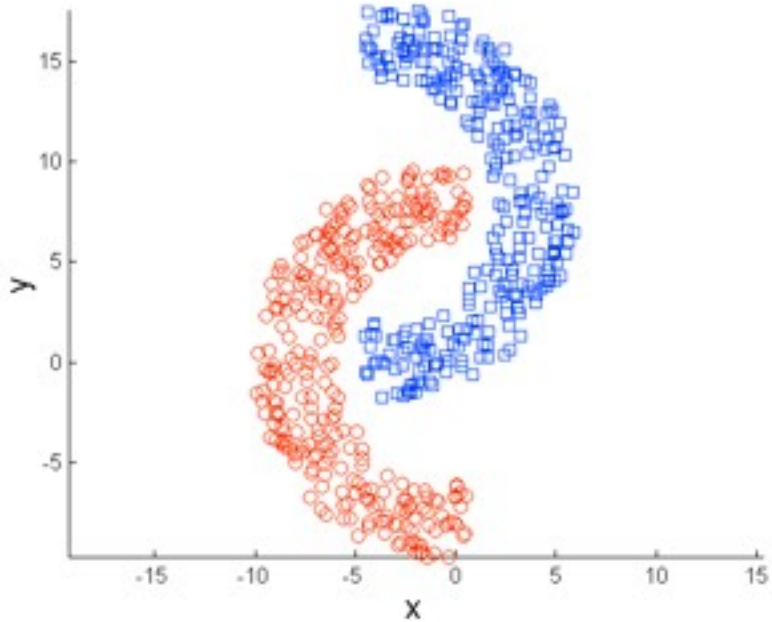


Original Points

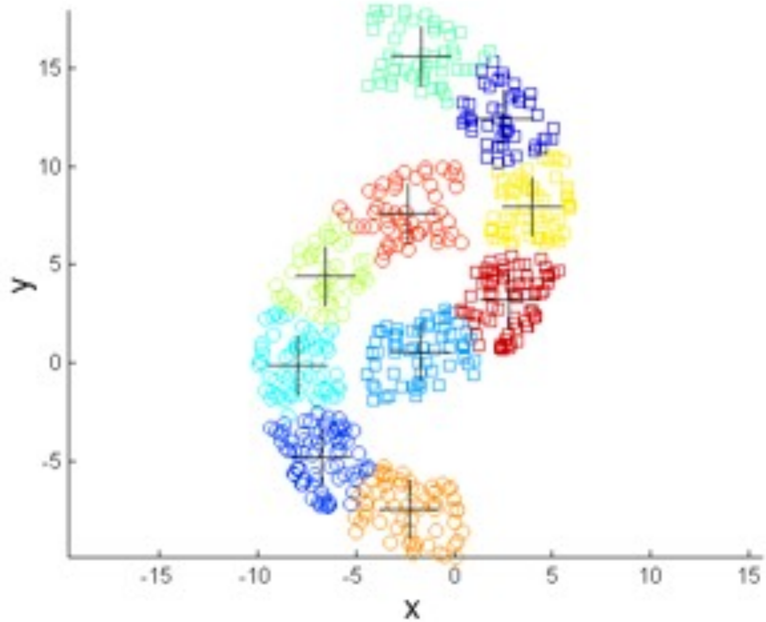


K-means Clusters

Overcoming K-means Limitations



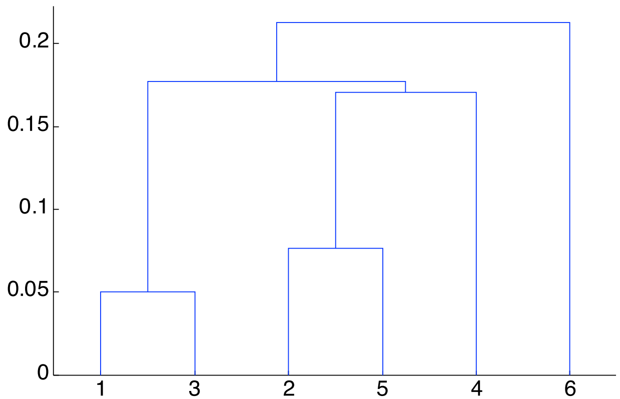
Original Points

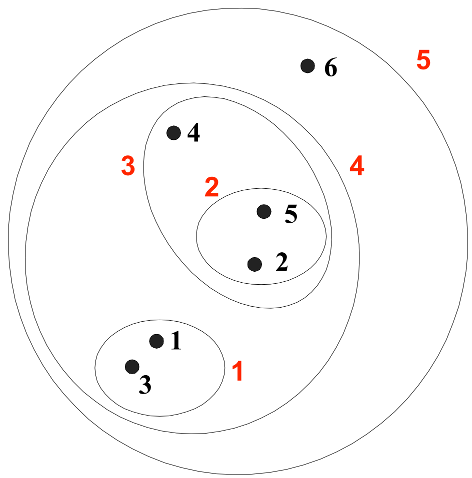


K-means Clusters

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits





Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by 'cutting' the dendogram at the proper level

- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Hierarchical Clustering

- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

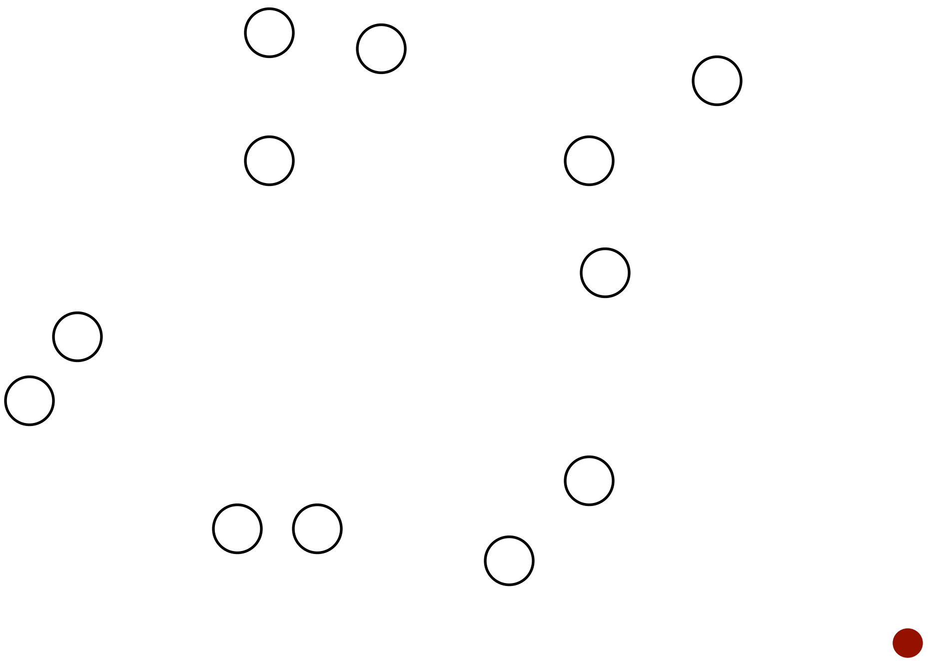
Agglomerative Clustering Algorithm

- Most popular hierarchical clustering technique

- Algorithm:
  1. Let each data point be a cluster
  1. Compute the distance matrix n x n
  2. Repeat
  3. Merge the two closest clusters
  4. Update distance matrix
  5. **Until** only a single cluster remains

Starting Situation

- Start with clusters of individual points and a distance matrix n x n

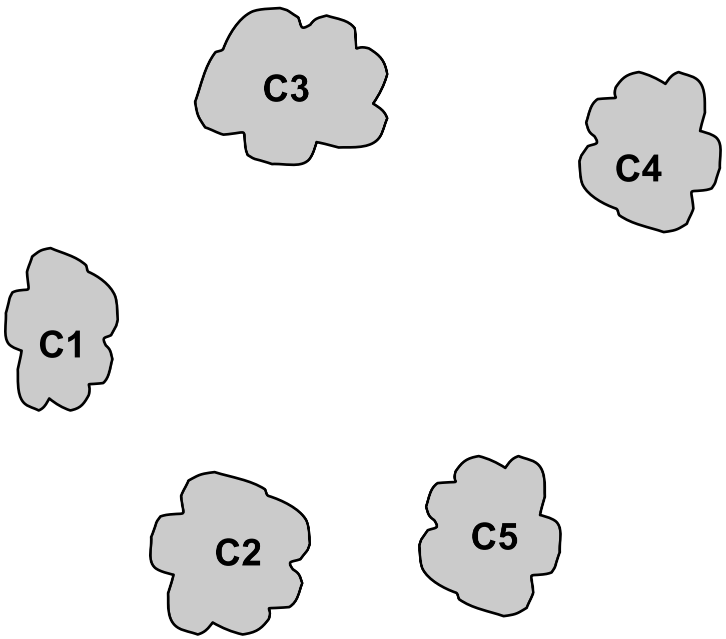


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix

Intermediate Situation

- After some merging steps, we have some clusters

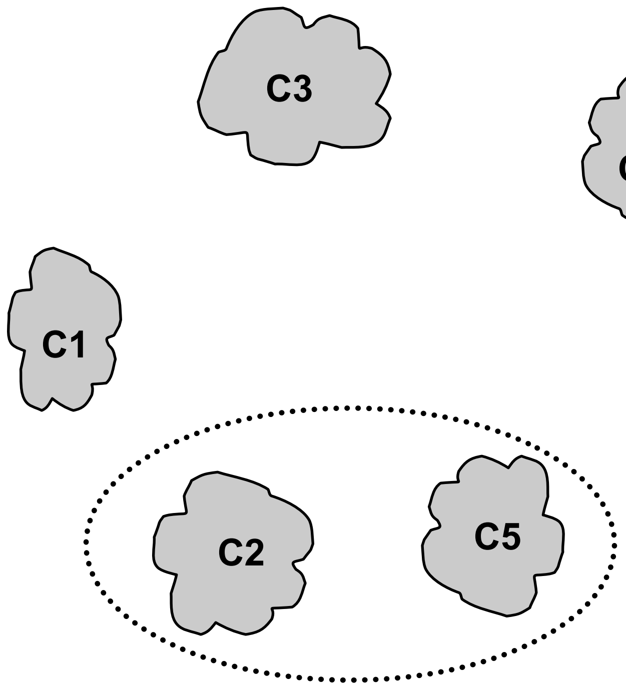


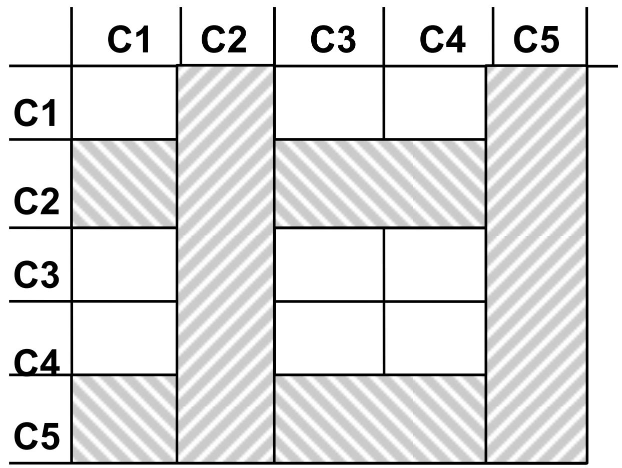
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance Matrix

Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the distance matrix.

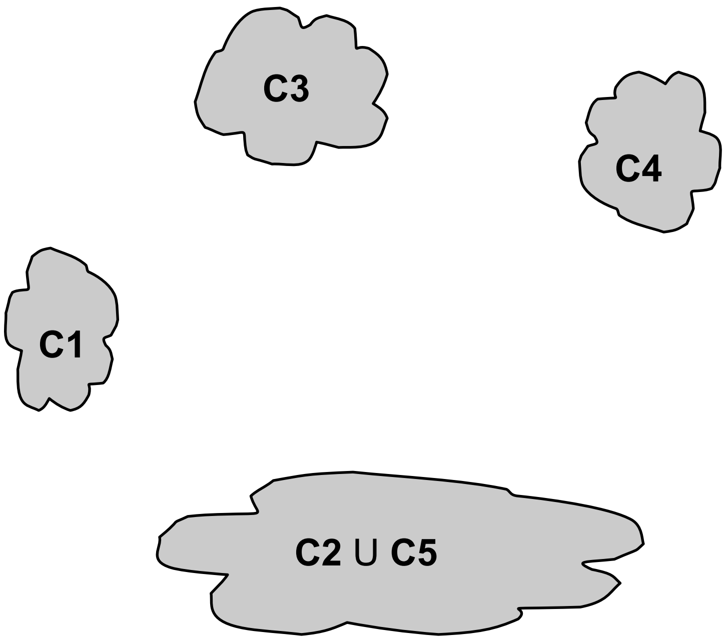




Distance Matrix

After Merging

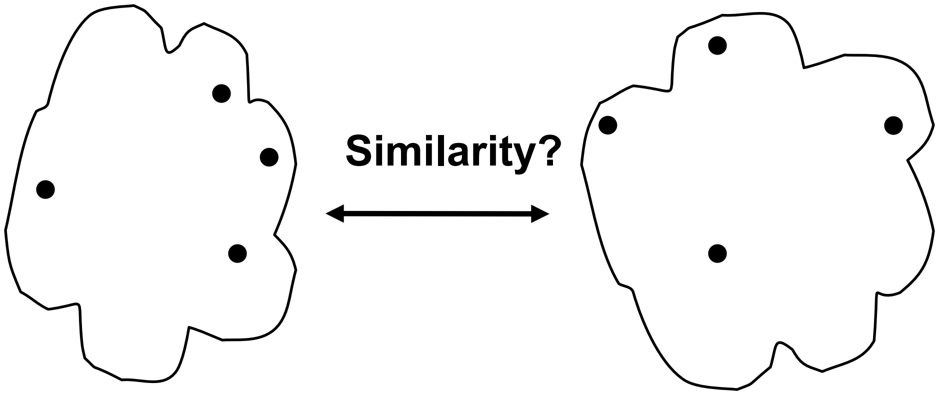
- The question is “How do we update the distance matrix?”



	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Distance Matrix

How to Define Inter-Cluster Similarity

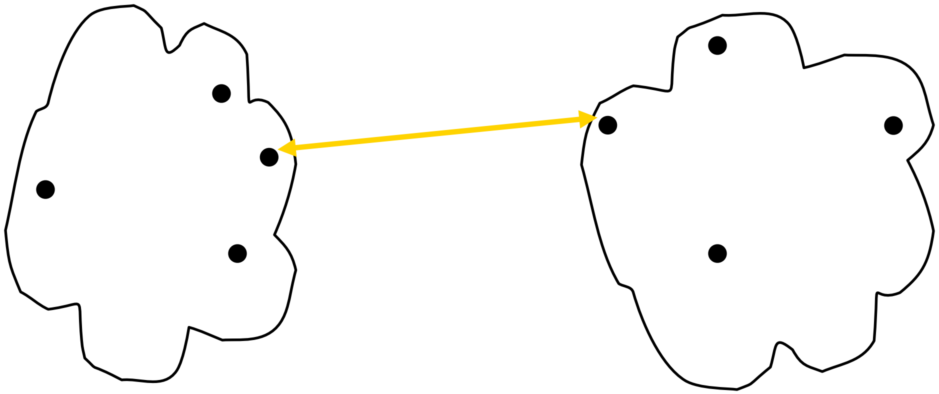


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

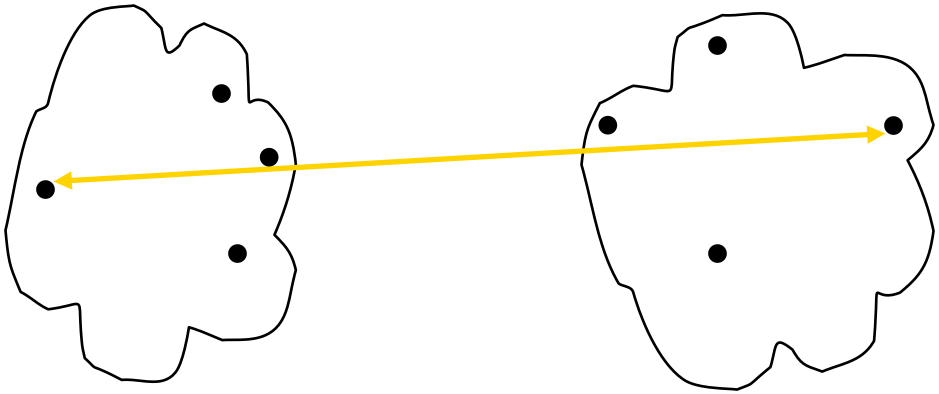


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

How to Define Inter-Cluster Similarity

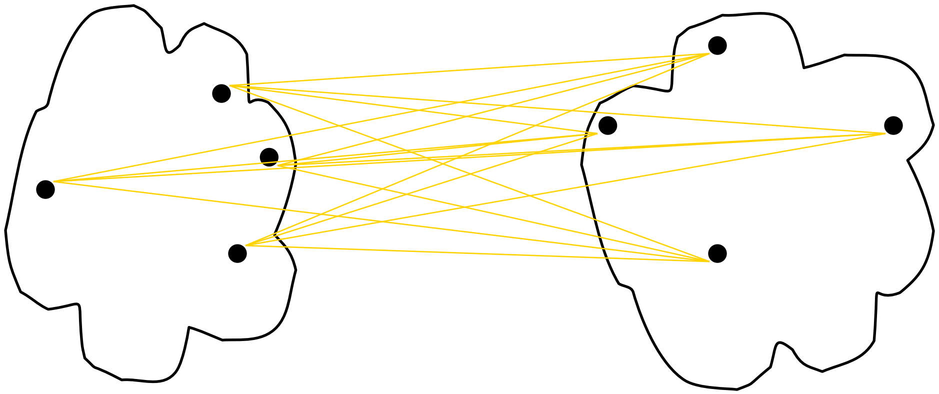


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

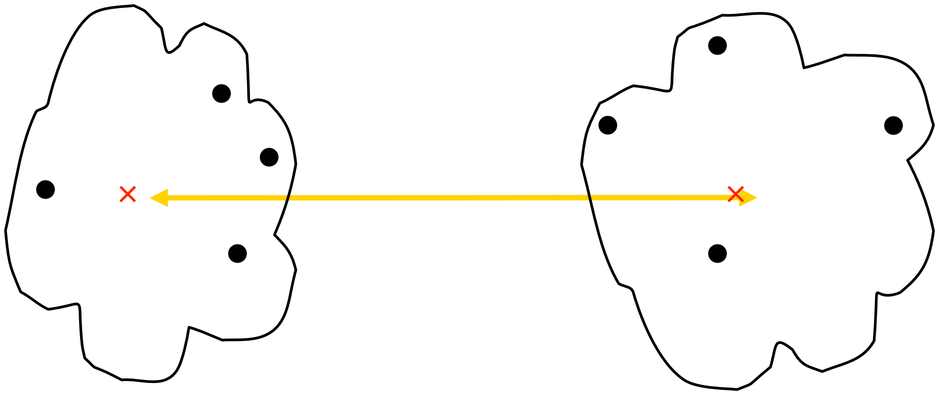


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  – Ward's Method uses squared error

How to Define Inter-Cluster Similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

Hierarchical Clustering: Problems and Limitations

- Once a decision is made to combine two clusters, it cannot be undone

- No objective function is directly minimized

- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters

Cluster Validity

- For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?
- But “clusters are in the eye of the beholder”!
- Then why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters

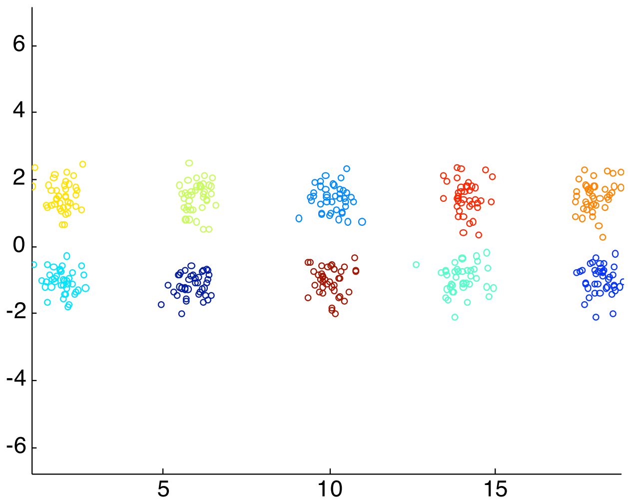
Measures of Cluster Validity

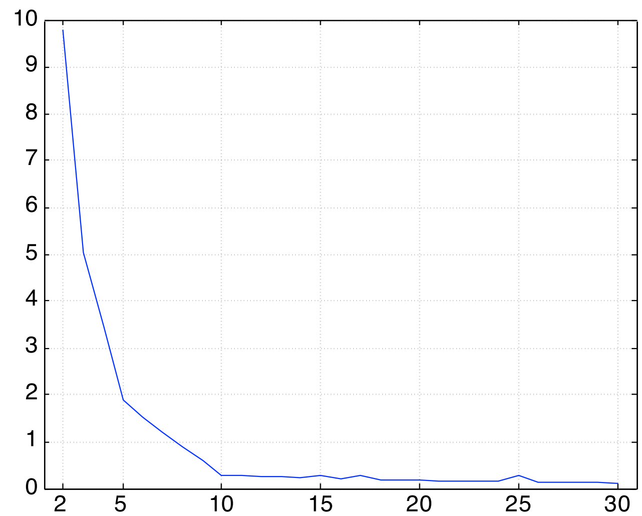
- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
  - Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information.
    - Sum of Squared Error (SSE)
  - Relative Index: To compare two different clusterings or clusters.
    - An external or internal index is used for this function, e.g., SSE or entropy

- Sometimes these are referred to as criteria instead of indices

Internal Measures: SSE

- Clusters in more complicated figures aren't well separated
- Internal Index: Used to measure the goodness of a clustering structure without respect to external information
  - SSE
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to estimate the number of clusters





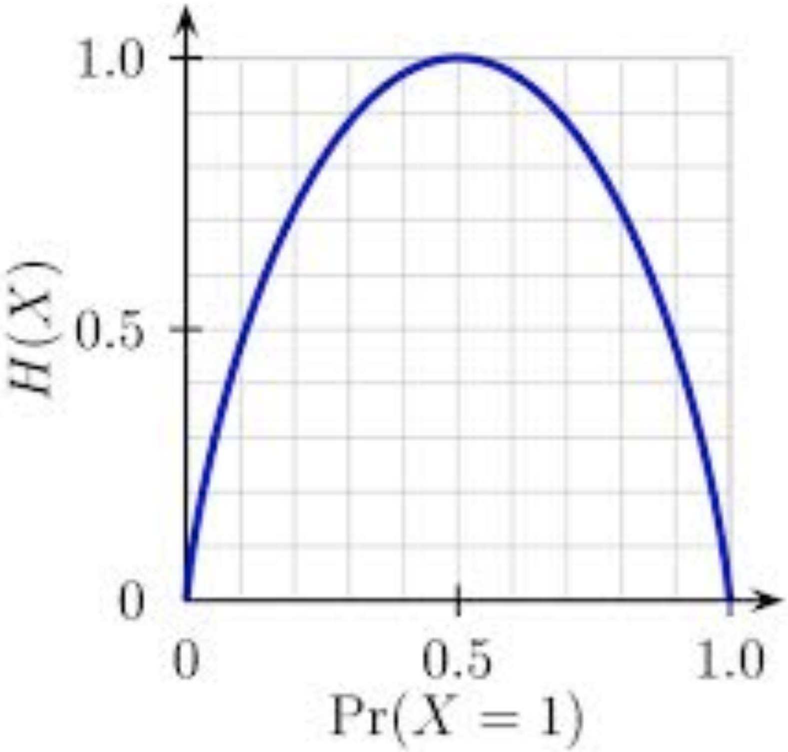
Entropy: definition

- Given a discrete random variable X with possible value {1,..,n} entropy is defined as

$$H(X) = -\sum_{i=1}^{n} P(X = i) \log_2 P(X = i)$$

- Entropy measure how **uncertain** is an event, the larger the entropy the more uncertain is the event

Entropy: intuition



Entropy of a binary variable.

Examples:

1. entropy of unbiased coin vs biased coin?

2. entropy of a dice roll?

3. Probability distribution: $P(X=c_i)$ = probability of finding character c_i in a text document. Easier to compress a document when entropy is high or low?

External Measures of Cluster Validity: Entropy

Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

Topics={Entertainment, Financial, Metro,...}={1,2,3,...k}

p_{ij}= Probability that an element of cluster j belongs to topic i.

E.g. p_{13}=1/685

For a cluster j better to have higher or lower entropy?

External Measures of Cluster Validity: Entropy

Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

m_j= size of cluster j, m=number of docs.

Entropy and purity of a cluster

$$e_j = -\sum_{i=1}^{K} p_{ij} \log p_{ij} \qquad \text{purity}_j = \max_i p_{ij}$$

Entropy and purity of a clustering:

$$\sum_j \frac{m_j}{m} e_j \qquad \sum_j \frac{m_j}{m} \text{purity}_j$$

Final Comment on Cluster Validity

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

Algorithms for Clustering Data, Jain and Dubes

k-means++

Algorithm 1 k-means++(k) initialization.

1: $\mathcal{C} \leftarrow$ sample a point uniformly at random from X
2: **while** $|\mathcal{C}| < k$ **do**
3: Sample $x \in X$ with probability $\frac{d^2(x,\mathcal{C})}{\phi_X(\mathcal{C})}$
4: $\mathcal{C} \leftarrow \mathcal{C} \cup \{x\}$
5: **end while**

where:

$$d(x,\mathcal{C}) = \min_{c \in \mathcal{C}} d(x,c), \quad \Phi_X(\mathcal{C}) = \sum_{x \in X} d^2(x,\mathcal{C})$$

$d^2(x,C)$ measures how "good" is the clustering for point x.
Points that are *relatively* far away from "their" centroids will be selected with higher probability.

K-means ++

- K-means++:
  - Initialize the centroids as in Algorithm 1
  - Run K-means algorithm to improve the clustering.

- **Theorem:** Let C_KM++ be the clustering produced by the K-means++ algorithm, let C_opt be an optimal clustering (with minimum SSE among all possible clusterings). Then, **SSE(C_KM++) <= 8*(ln k + 2)*SSE(C_opt)**, on expectation (average).

- **Exercise:** give an example where k-means computes an approximation ``worse'' than k-means++.

Algorithms

- K-means:
  - no guarantees on the quality of the solution
  - it always terminates
  - running time could be exponential but it is OK in practice

- K-means++
  - it always terminates
  - O(log k)-approximation on the quality of the solution.
  - In practice the advantage is noticeable for large k