CS6510
Applied Machine Learning

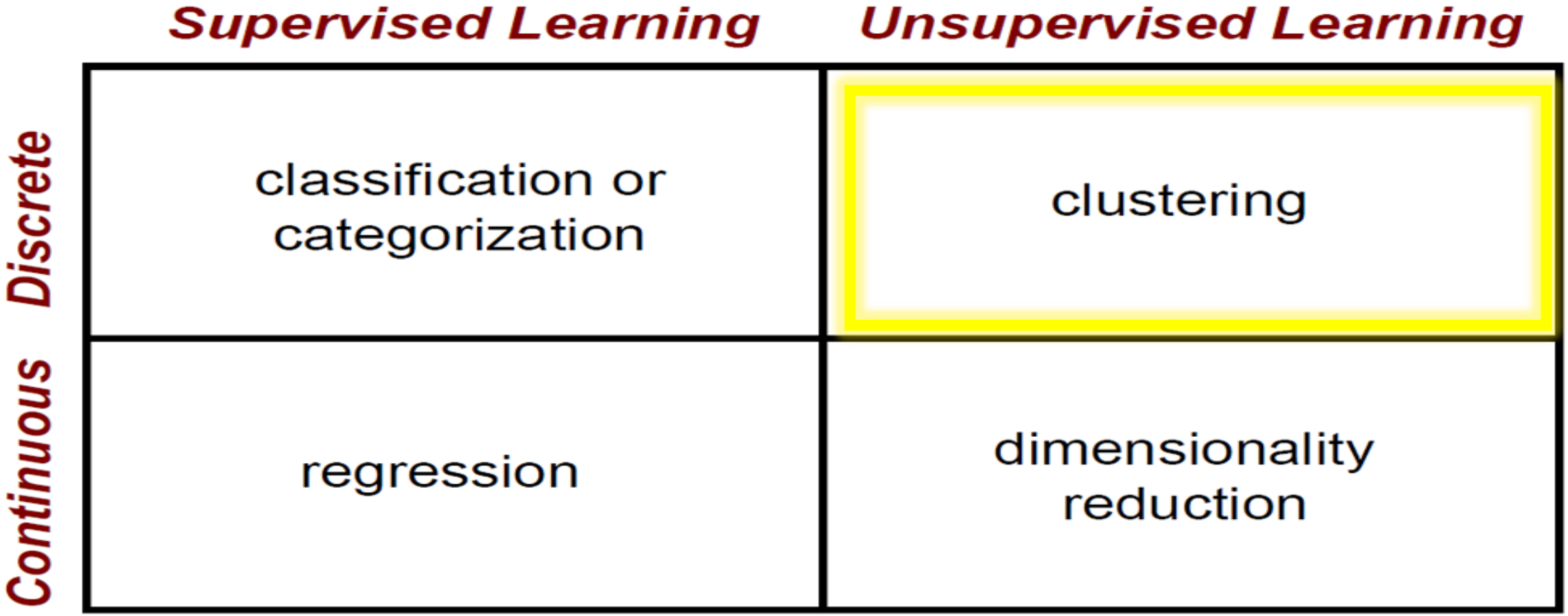
Clustering

8, 22 Oct 2016

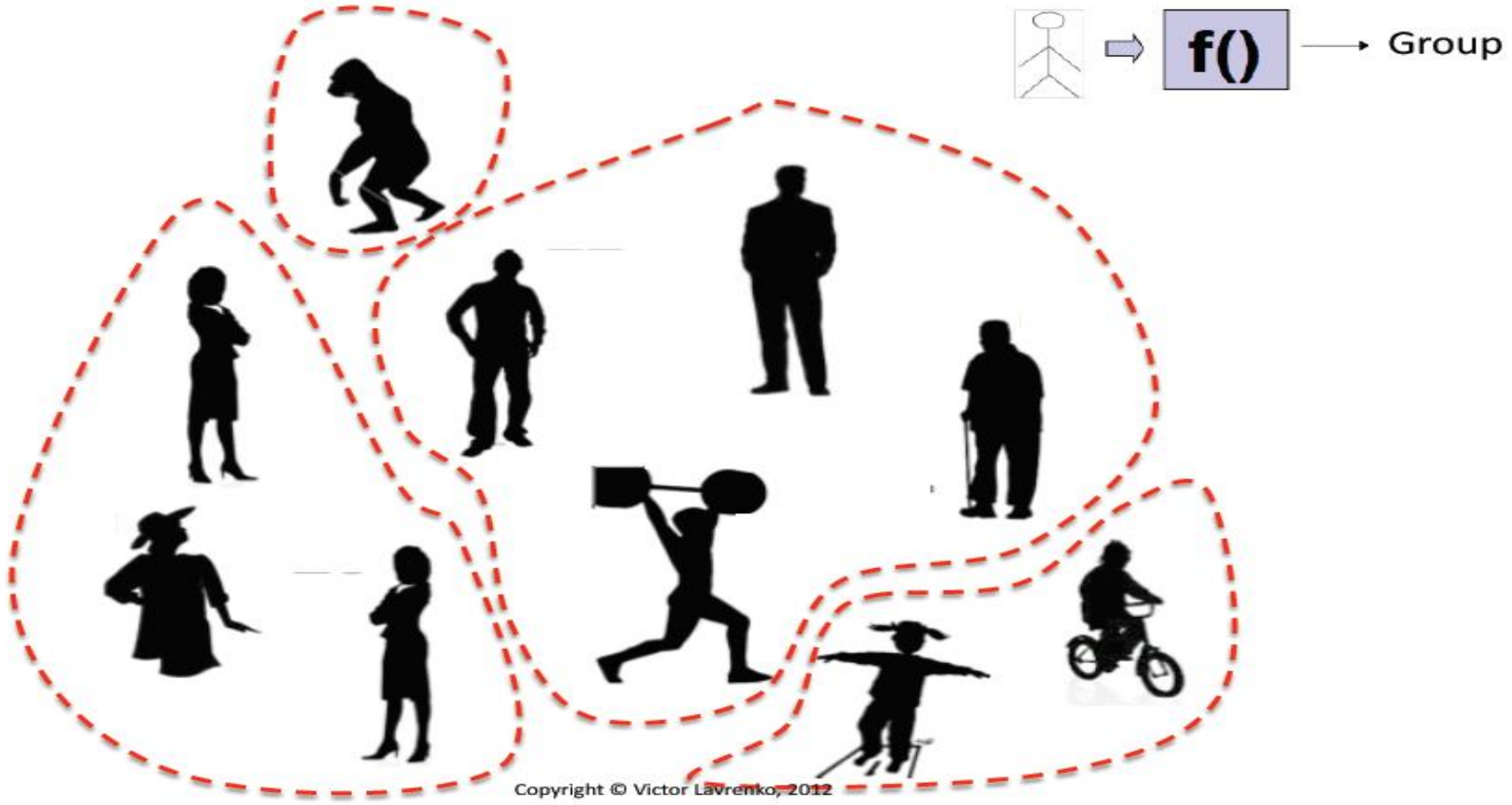
Vineeth N Balasubramanian

आई आई टी हैदराबाद
IIT Hyderabad

ML Problems


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

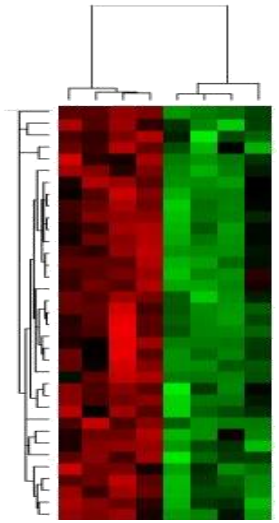
Clustering (Unsupervised Learning)

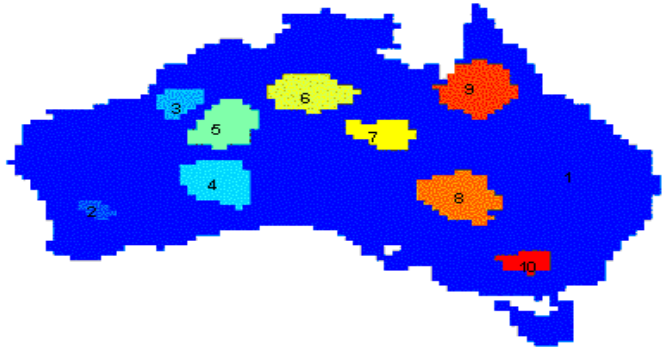


Where is Clustering used?

- **Understanding**
  - Group genes and proteins that have similar functionality, or group stocks with similar price fluctuations
- **Summarization**
  - Reduce the size of large data sets

More real-world applications?





Clustering precipitation in Australia

Where is Clustering Used?

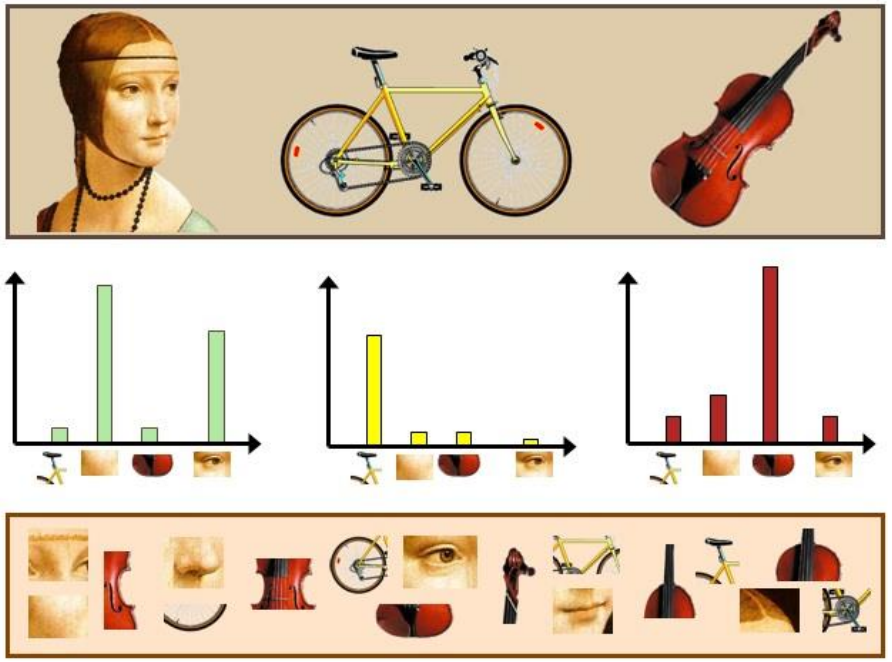
- Bank/Internet Security: fraud/spam pattern discovery
- Biology: taxonomy of living things such as kingdom, phylum, class, order, family, genus and species
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Climate change: understanding earth climate, find patterns of atmospheric and ocean
- Finance: stock clustering analysis to uncover correlation underlying shares
- Image Compression/segmentation: coherent pixels grouped
- Information retrieval/organisation: Google search, topic-based news
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Social network mining: special interest group automatic discovery

Clustering: Objectives

- Discover underlying structure of data
- What sub-populations exist in the data?
  - How many are there?
  - What are their sizes?
  - Do elements in a sub-population have any common properties?
  - Are sub-populations cohesive? Can they be further split?
  - Are there outliers?

Clustering as Preprocessing

- Popular application of clustering
- Estimated group labels h_j (soft) or b_j (hard) may be seen as the dimensions of a new k dimensional space, where we can then learn our discriminant or regressor
- E.g. Bag-of-words representation in images

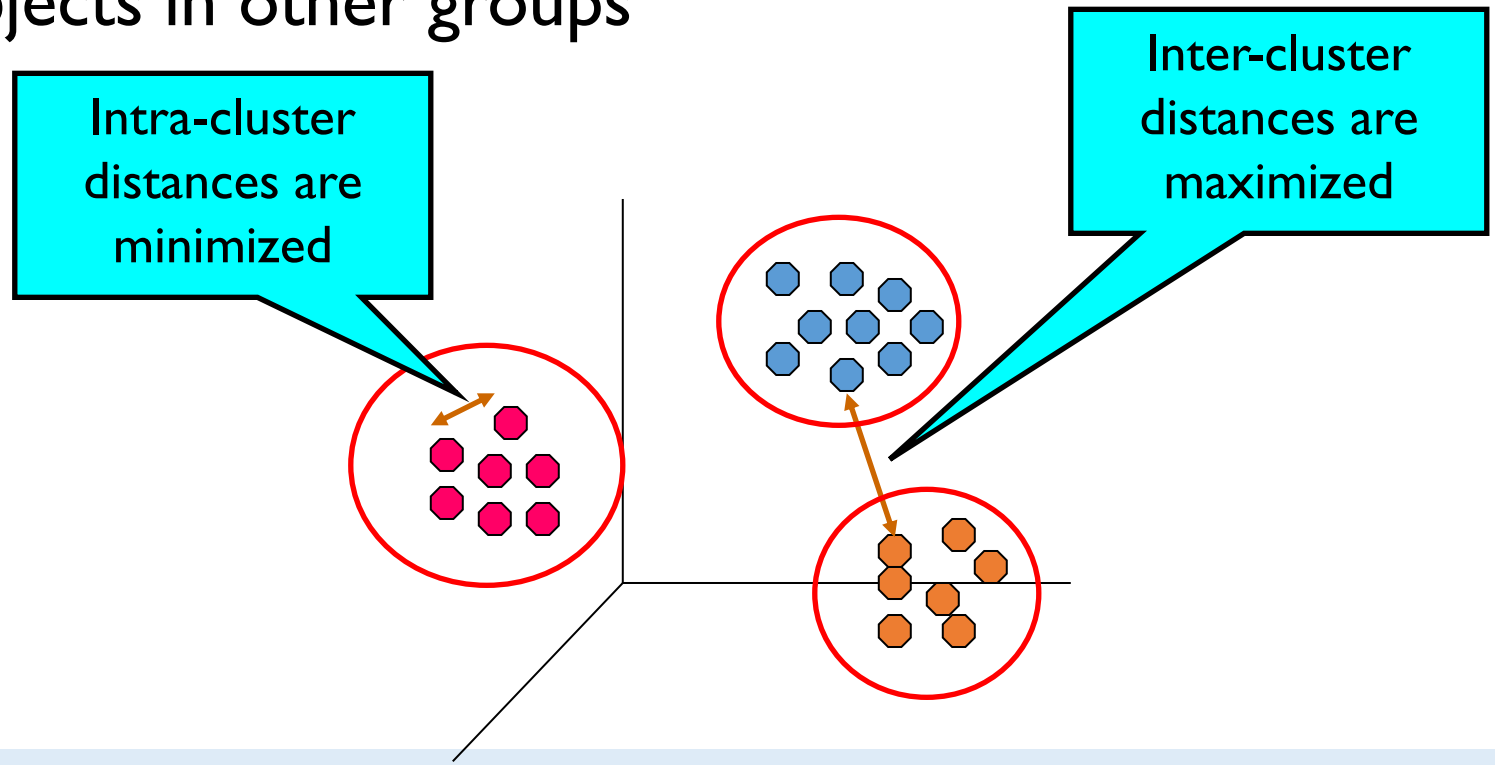


Types of Clustering Methods

- **In terms of objective:**
  - Monothetic: cluster members have some common property
    - E.g. All are males aged 20-35, or all have X% response to test B
  - Polythetic: cluster members are similar to each other
    - Distance between elements defines membership
- **In terms of overlap of clusters**
  - Hard clustering: clusters do not overlap
  - Soft clustering: clusters may overlap
    - "Strength of association" between element and cluster
- **In terms of methodology**
  - Flat/partitioning (vs) hierarchical: Set of groups (vs) taxonomy
  - Density-based (vs) Model-based: DBSCAN vs GMMs

Clustering Methods

- 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
- How?

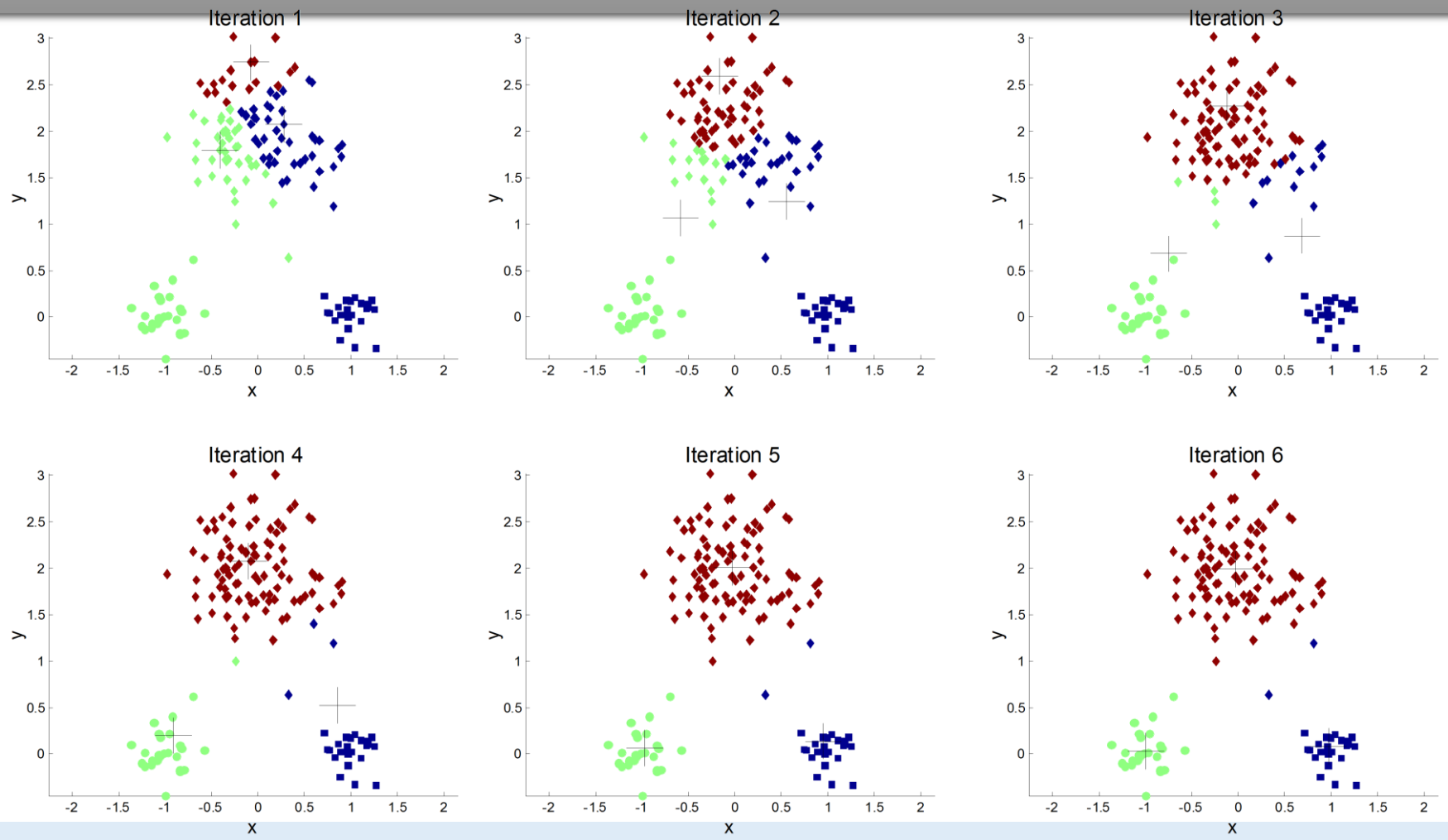


k-Means Clustering

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The basic algorithm is very simple

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

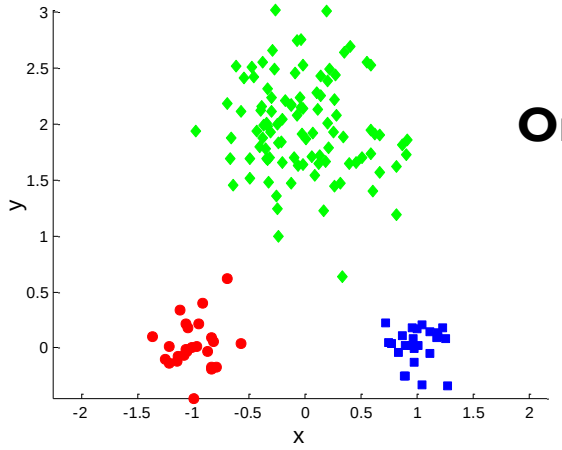
k-Means: Illustration



k-Means Clustering

- 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.
- K-means will converge for common similarity measures mentioned above (**local minimum** though)
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to 'Until relatively few points change clusters'
- Nearby points may not end up in the same cluster! Example?

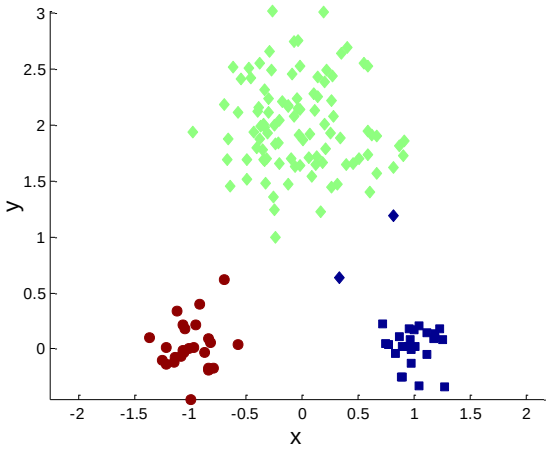
Two different k-Means clusterings

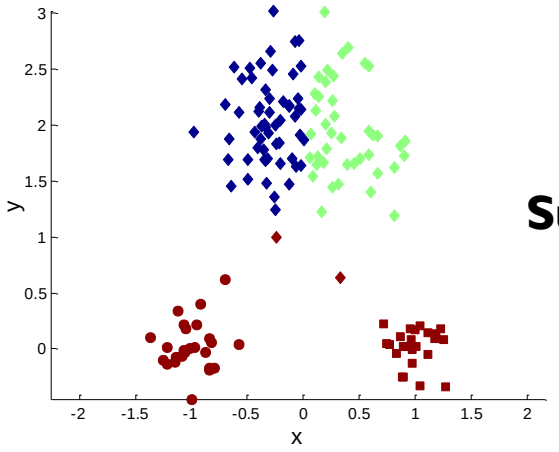


Original Points

What's the problem?

Optimal Clustering





Sub-optimal Clustering

Selecting Initial Centroids

- If there are K 'real' clusters then the chance of selecting one centroid from each cluster is small.
  - Chance is relatively small when K is large
  - If clusters are the same size, n, then

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

  - For example, if K = 10, then probability = 10!/1010 = 0.00036
  - Sometimes the initial centroids will readjust themselves in 'right' way, and sometimes they don't

Possible Solutions

- Multiple runs
  - Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
  - Select most widely separated
- Bisecting K-means
  - Not as susceptible to initialization issues

Evaluating k-Means Clusters

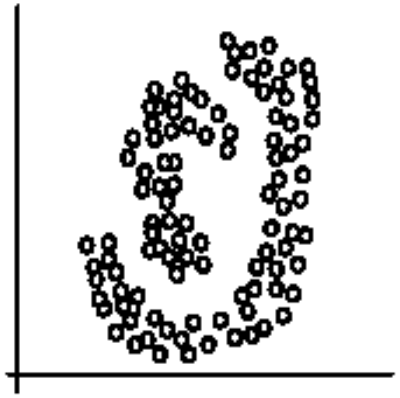
- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

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

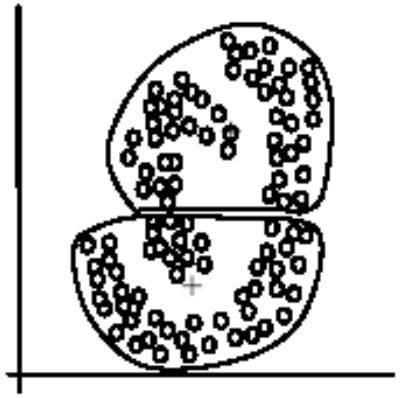
  - x is a data point in cluster Ci and m_i is the representative point for cluster Ci
  - Can show that m_i corresponds to the center (mean) of the cluster
- Given two clusters, we can choose the one with the smaller error
- One easy way to reduce SSE is to increase K, the number of clusters
- A good clustering with smaller K can have a lower SSE than a poor clustering with higher K
- Relatively faster than other clustering methods: O(# iterations * # clusters * # instances * # dimensions)

Limitations

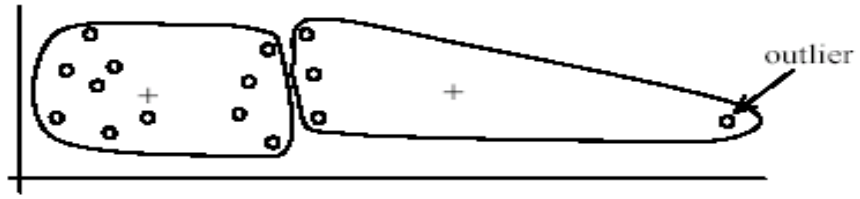
- k-Means has problems when clusters are of differing
  - Sizes, Densities, Non-globular shapes
- Sensitive to outliers
- The number of clusters (K) is difficult to determine



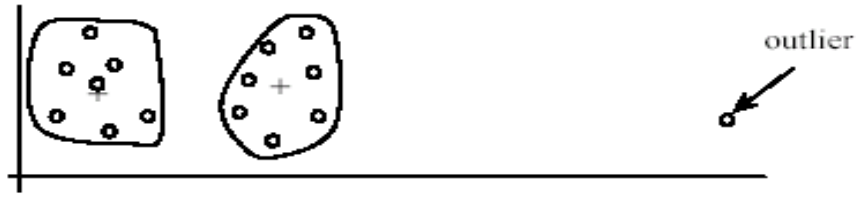
(A): Two natural clusters



(B): *k*-means clusters



(A): Undesirable clusters



(B): Ideal clusters

Extensions

- Use of various distance metrics
  - Euclidean distance

$$d(\mathbf{x},\mathbf{y}) = \sqrt{|x_1 - y_1|^2 + |x_2 - y_2|^2 \cdots + |x_n - y_n|^2}$$

  - Manhattan (city-block) distance

$$d(\mathbf{x},\mathbf{y}) = |x_1 - y_1| + |x_2 - y_2| \cdots + |x_n - y_n|$$

  - Cosine distance

$$\cos(\mathbf{x},\mathbf{y}) = \frac{x_1 y_1 + \cdots + x_n y_n}{\sqrt{x_1^2 + \cdots + x_n^2}\sqrt{y_1^2 + \cdots + y_n^2}}$$

$$d(\mathbf{x},\mathbf{y}) = 1 - \cos(\mathbf{x},\mathbf{y})$$

  - Chebyshev distance

$$dist(\mathbf{x}_i,\mathbf{x}_j) = \max(|x_{i1} - x_{j1}|, |x_{i2} - x_{j2}|, ..., |x_{ir} - x_{jr}|)$$

Extensions

- k-Medioids
- Bisecting k-Means
- k-Means+

Challenge

How many clusters?

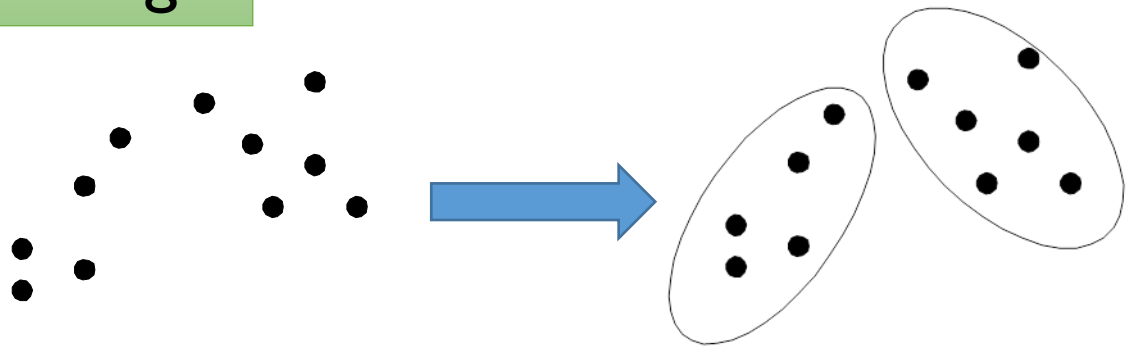
Six Clusters

Two Clusters

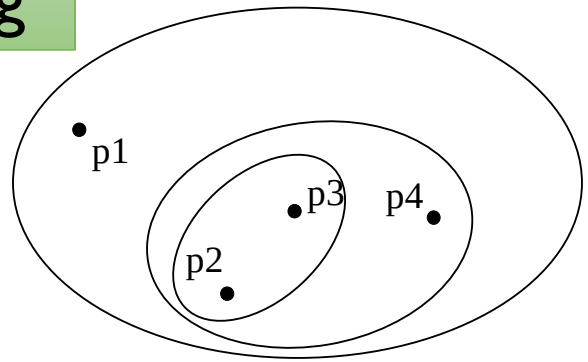
Four Clusters

Types of Clustering Methods

Partitional Clustering

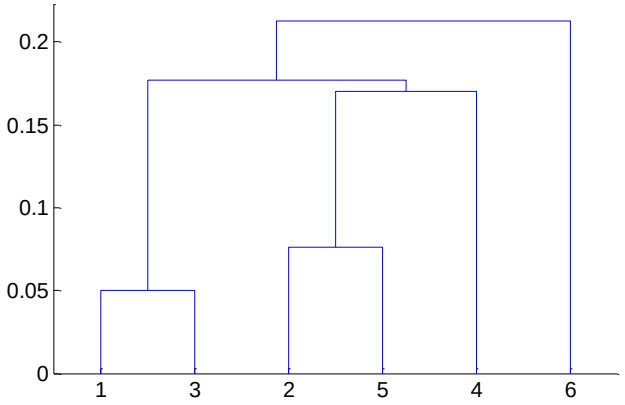


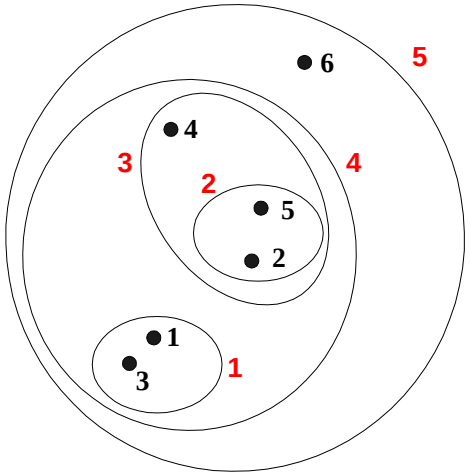
Hierarchical Clustering



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

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by 'cutting' the dendrogram at the proper level
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Hierarchical Clustering

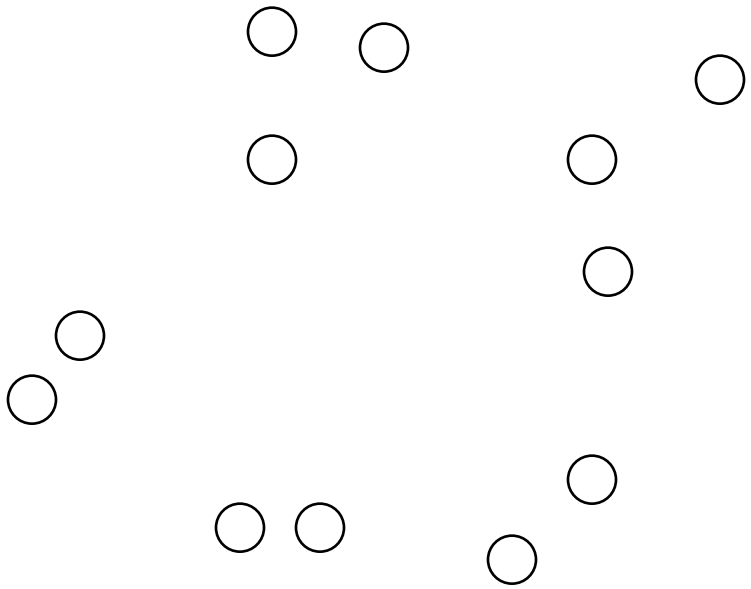
- Two main types of hierarchical clustering
  - **Agglomerative:**
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - **Divisive:**
    - Start with one, all-inclusive cluster
    - At each step, split 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

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. Repeat
     1. Merge the two closest clusters
     2. Update the proximity matrix
  4. Until only a single cluster remains
- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

Methodology

- Start with clusters of individual points and a proximity matrix

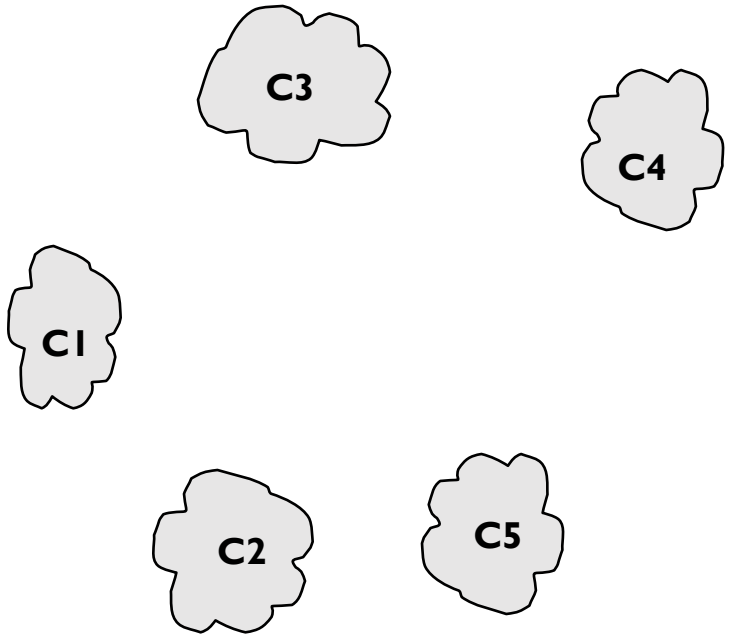


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

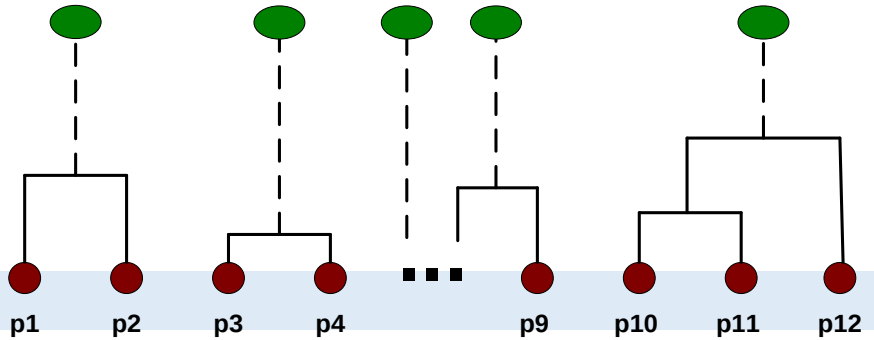
Methodology

- After some merging steps, we have some clusters



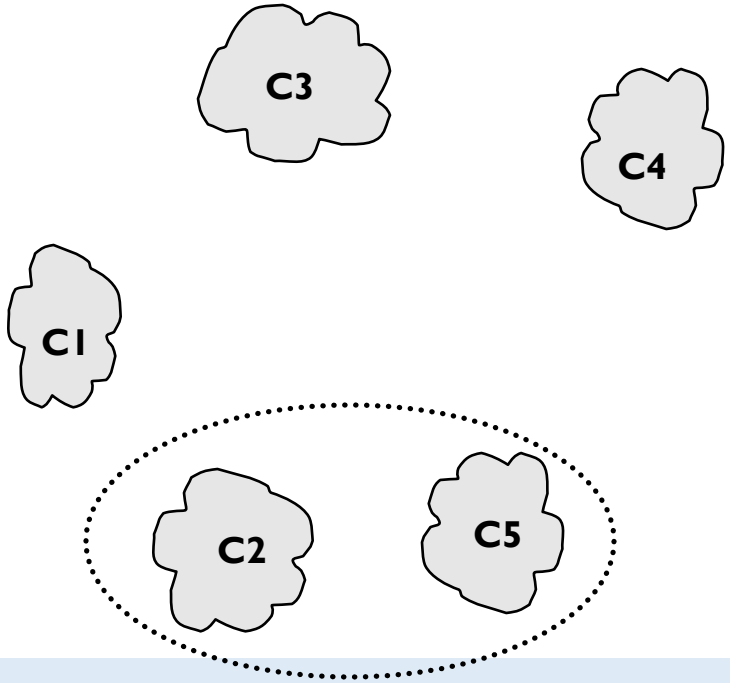
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



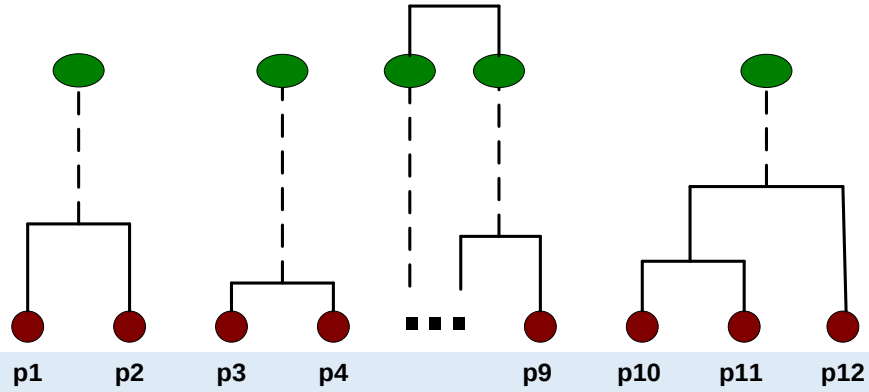
Methodology

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



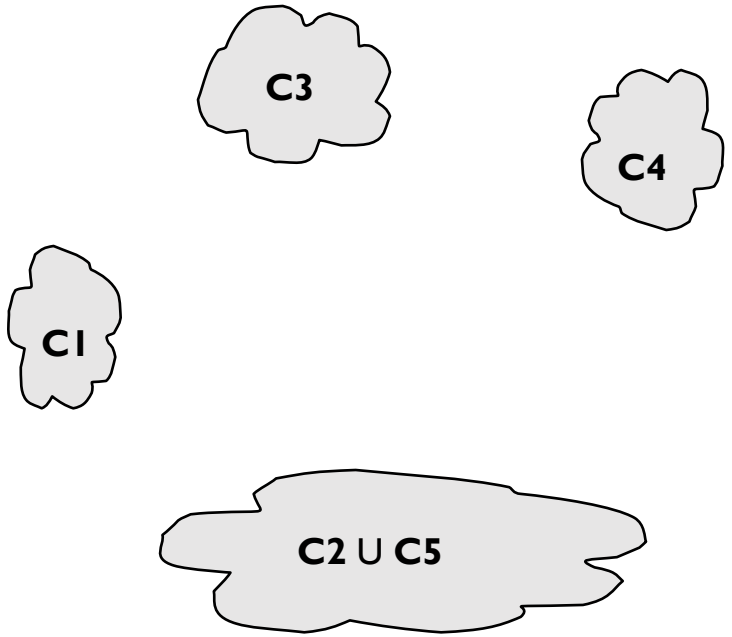
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



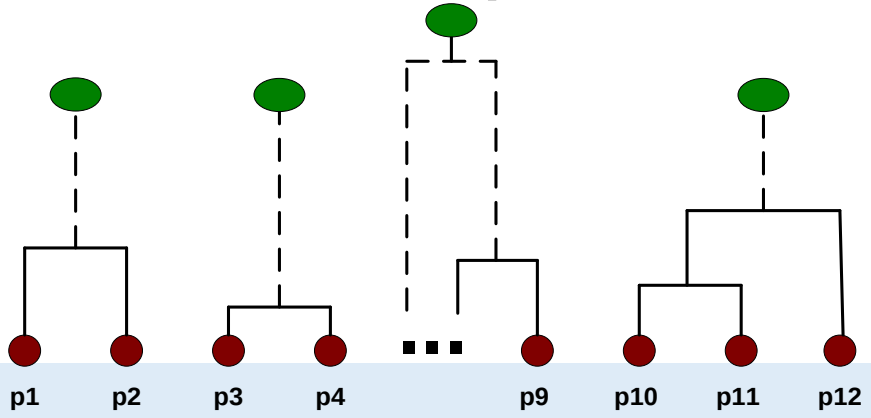
Methodology

- The question is "How do we update the proximity matrix?"

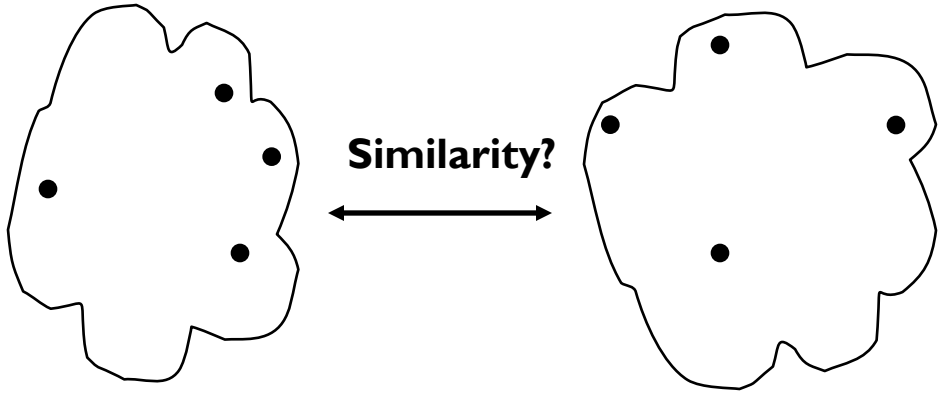


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



Defining Inter-cluster Similarity

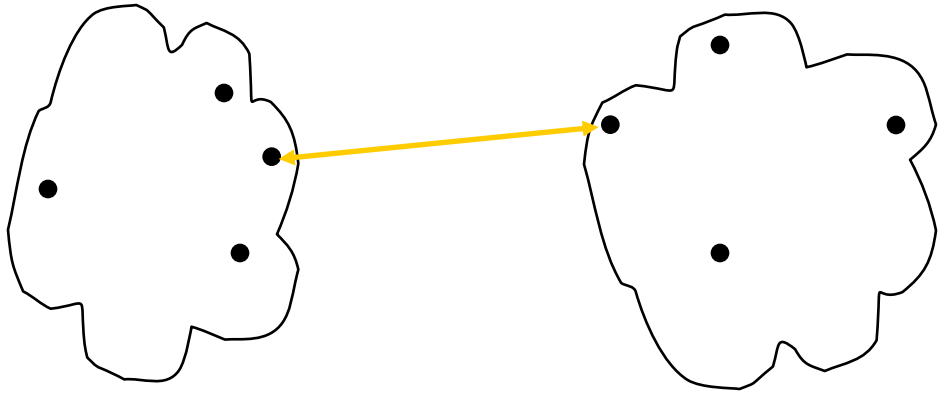


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

Defining Inter-cluster Similarity

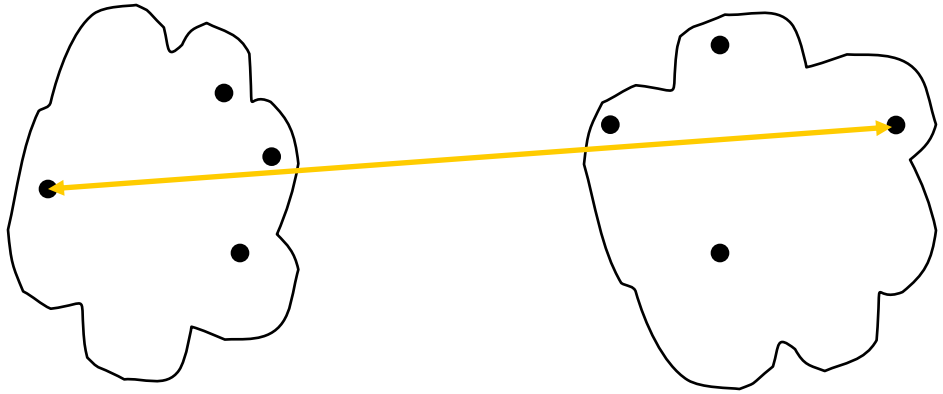


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- **MIN**
- MAX
- Group Average
- Distance Between Centroids

Defining Inter-cluster Similarity

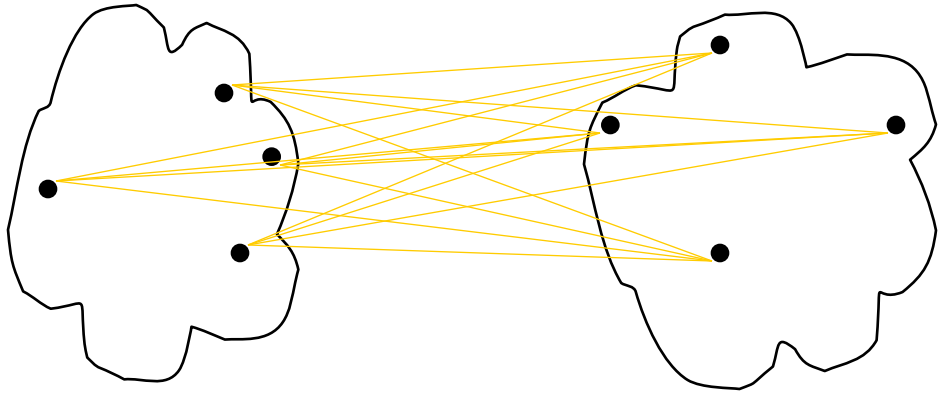


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- **MAX**
- Group Average
- Distance Between Centroids

Defining Inter-cluster Similarity

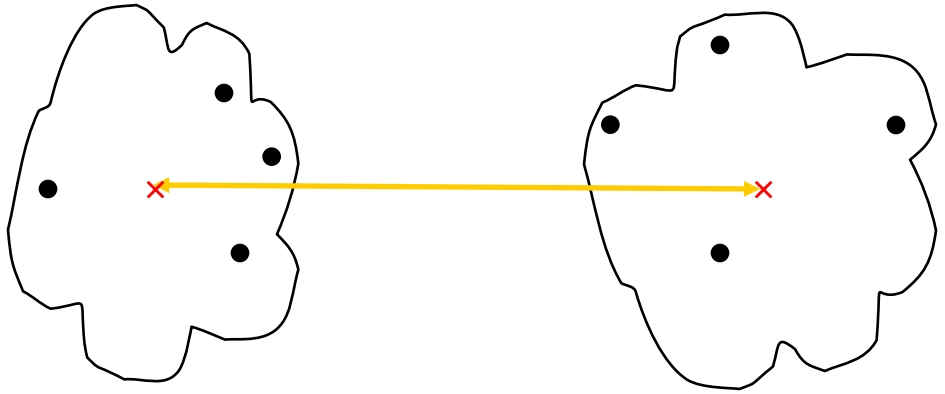


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- **Group Average**
- Distance Between Centroids

Defining Inter-cluster Similarity



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

Proximity Matrix

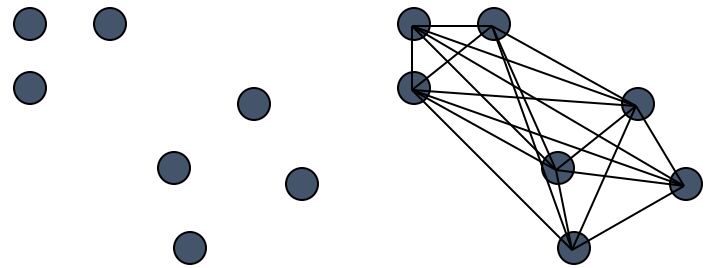
- MIN
- MAX
- Group Average
- **Distance Between Centroids**

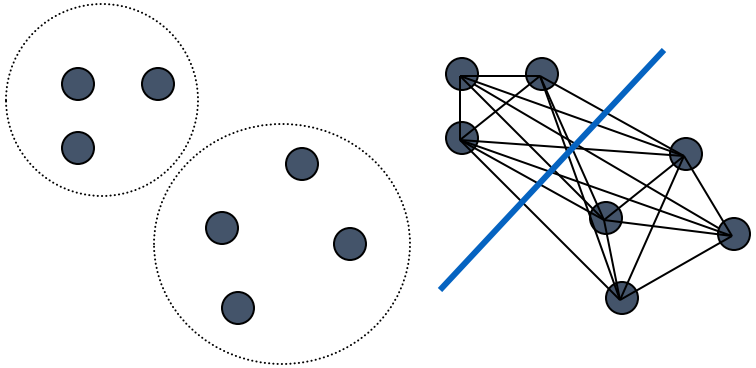
Hierarchical Clustering: Limitations

- 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 (MIN)
  - Difficulty handling different sized clusters and non-convex shapes (Group average, MAX)
  - Breaking large clusters (MAX)

Graph-Based/Spectral Clustering

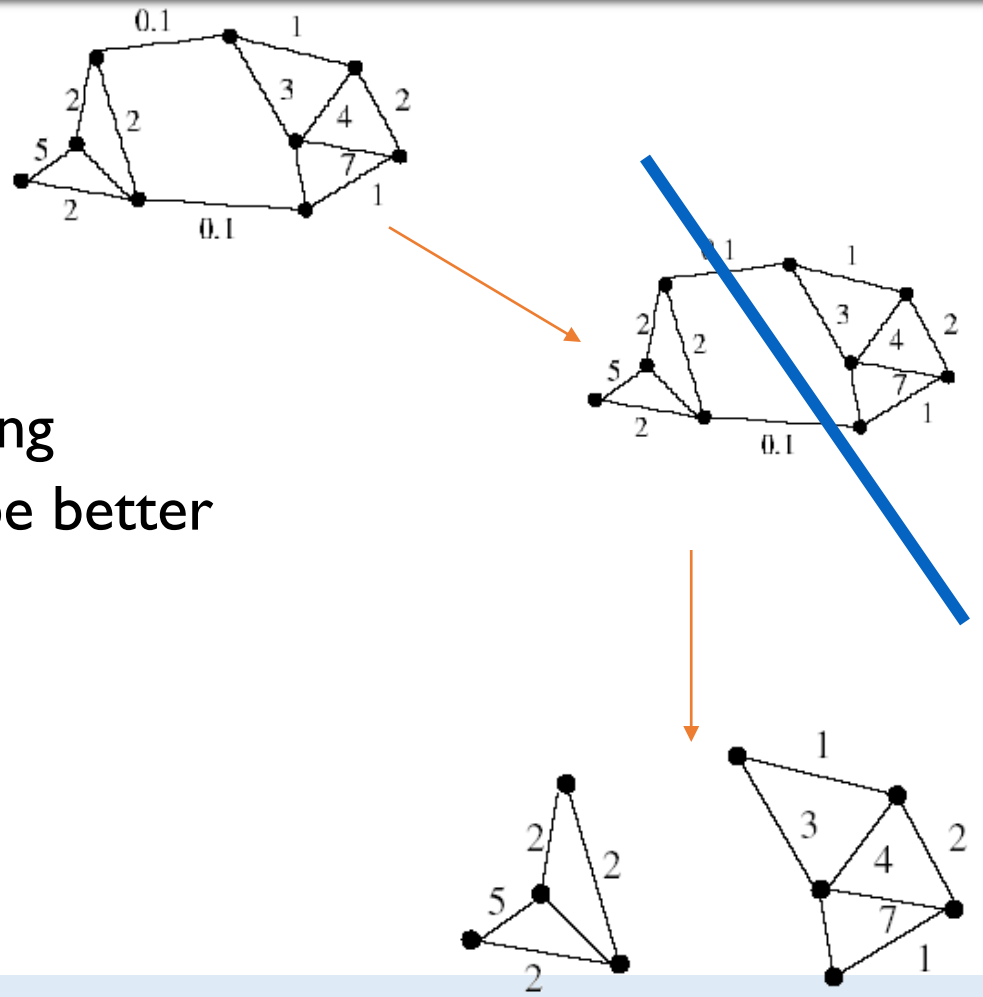
- Associate each data item with a vertex in a weighted graph
  - weights on the edges between elements are large if the elements are similar and small if they are not.
- Cut the graph into connected components with relatively large interior weights by cutting edges with relatively low weights.
- Clustering becomes a graph cut problem.





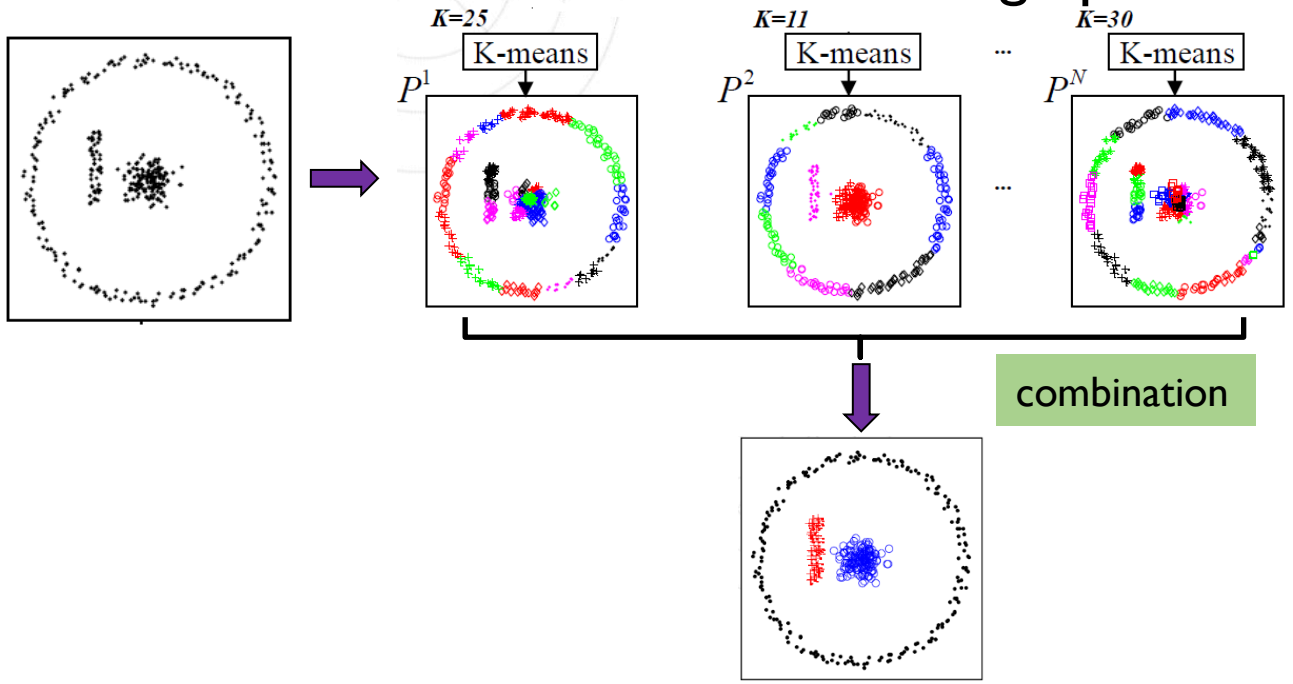
Graph-based/Spectral Clustering

- Method #1
  - Partition into two clusters
  - Use procedure recursively
- Method #2
  - Directly compute k-way partitioning
  - Experimentally has been seen to be better
- Examples
  - Minimum Cut
  - Normalized Cut



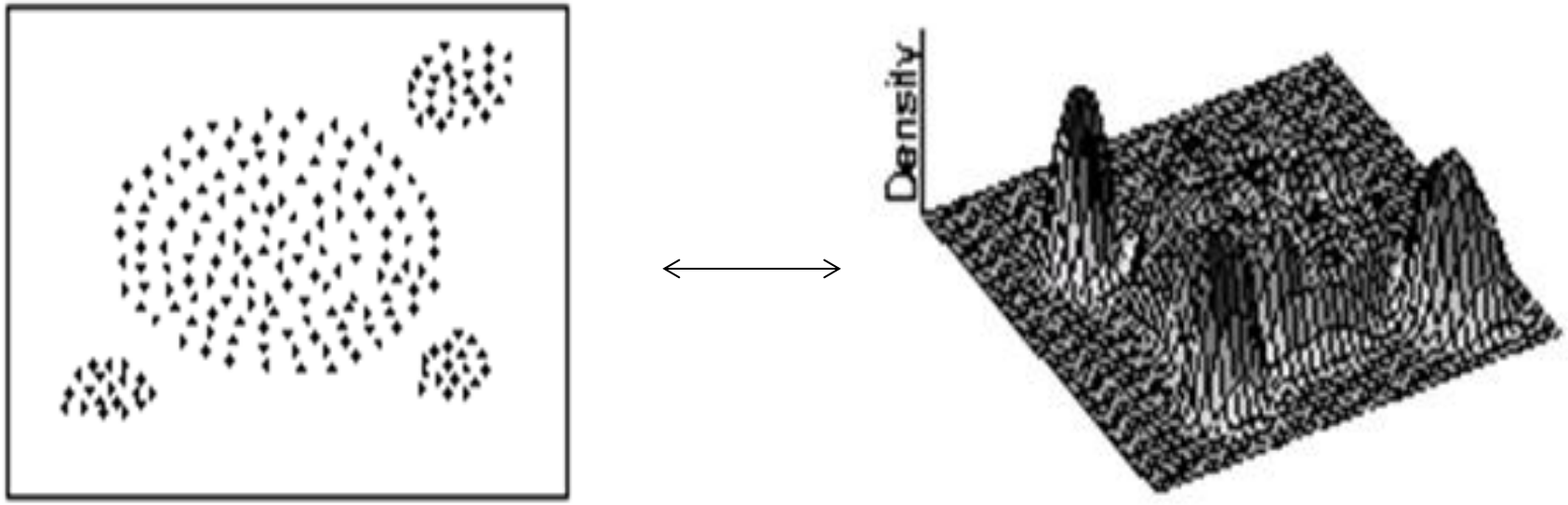
Clustering Ensembles

- Clustering ensemble approach
  - Combine multiple clustering results (different partitions)
  - Typical methods: Evidence-accumulation based, graph-based ……



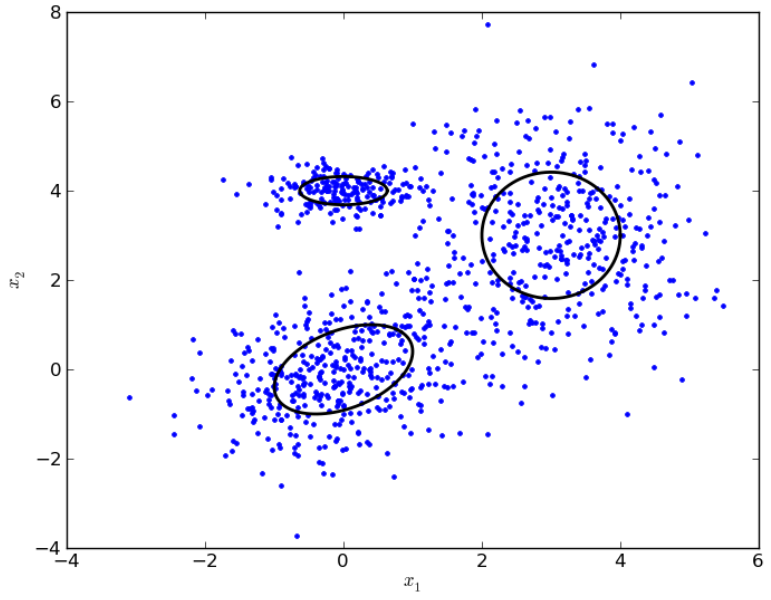
Other Clustering Methods

- Density-based methods: E.g. DBSCAN



Other Clustering Methods

- Model-based methods:
  - A generative model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - E.g. Gaussian Mixture Model (GMM) – we will see this later in this course



Cluster Validity

- 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: Used to compare two different clusterings or clusters.
  - Often an external or internal index is used for this function, e.g., SSE or entropy

Internal Measures

- Cluster Cohesion: Measures how closely related are objects in a cluster
  - Example: SSE
- Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters
- Example: Squared Error
  - Cohesion is measured by the within cluster sum of squares (SSE)

$$\boldsymbol{WSS} = \sum_{\boldsymbol{i}} \sum_{\boldsymbol{x} \in \boldsymbol{C_i}} (\boldsymbol{x} - \boldsymbol{m_i})^2$$

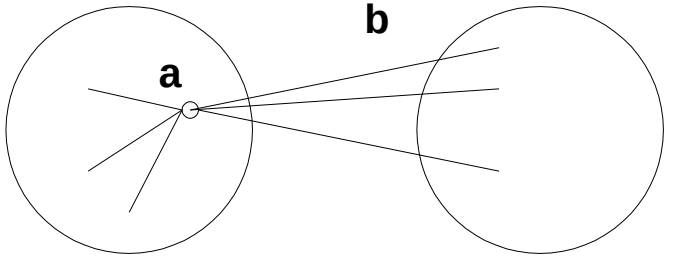
  - Separation is measured by the between cluster sum of squares

$$BSS = \sum_{i} |C_i| (m - m_i)^2$$

    - Where $|C_i|$ is the size of cluster i

Internal Measures: Silhouette Coefficient

- Combines ideas of both cohesion and separation, but for individual points as well as clusters
- For an individual point *i*
  - Calculate ***a*** = average distance of *i* to the points in its cluster
  - Calculate ***b*** = min (average distance of *i* to points in another cluster)
  - The silhouette coefficient for a point is then given by
    $s = 1 - a/b$ if $a < b$, (or $s = b/a - 1$ if $a \geq b$, not the usual case)
  - Typically between 0 and 1.
  - The closer to 1 the better.



External Measures

Table 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

entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the 'probability' that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^{K} \frac{m_i}{m} e_j$, where m_j is the size of cluster j, K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^{K} \frac{m_i}{m} purity_j$.

Open Issues with Clustering Methods

- Finding the number of "natural" clusters with arbitrary shapes
- Dealing with mixed types of features
- Handling massive amount of data – Big Data
- Coping with data of high dimensionality
- Performance evaluation (especially when no ground-truth available)
- "Holes" in the dataset – how to find?
  - E.g. in a disease database, we may find that:
    - certain symptoms and/or test values do not occur together, or
    - when a certain medicine is used, some test values never go beyond certain ranges
  - Discovery of such information can be important
    - Could mean the discovery of a cure to a disease or some biological laws