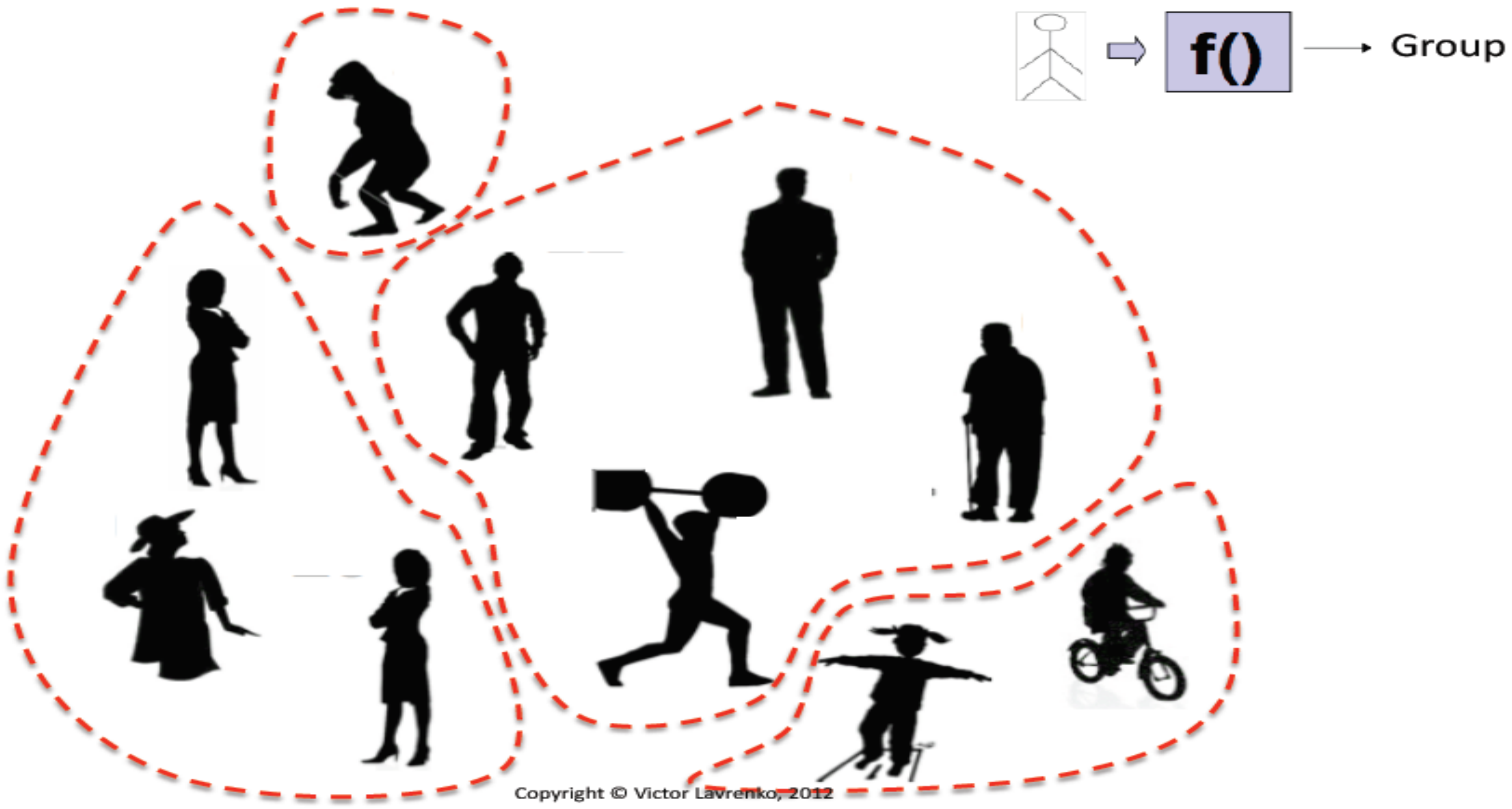
Clustering

आई आई टी हैदराबाद
IIT Hyderabad

ML Problems

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

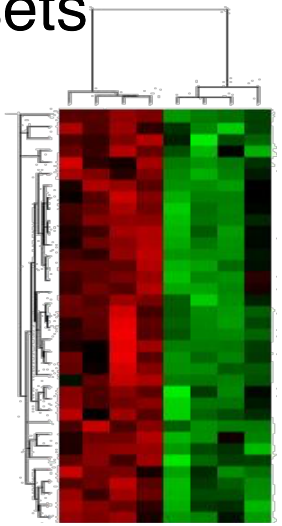
Clustering (Unsupervised Learning)

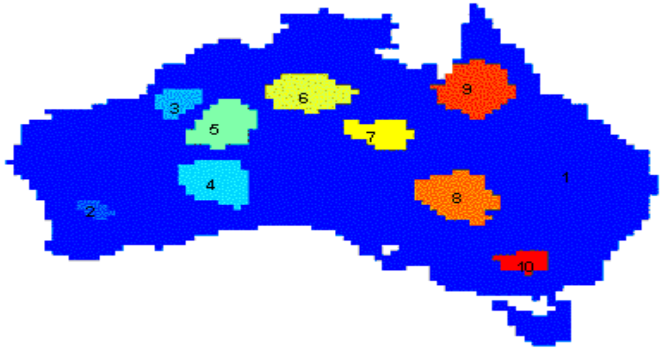


Where is Clustering used?

- **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?

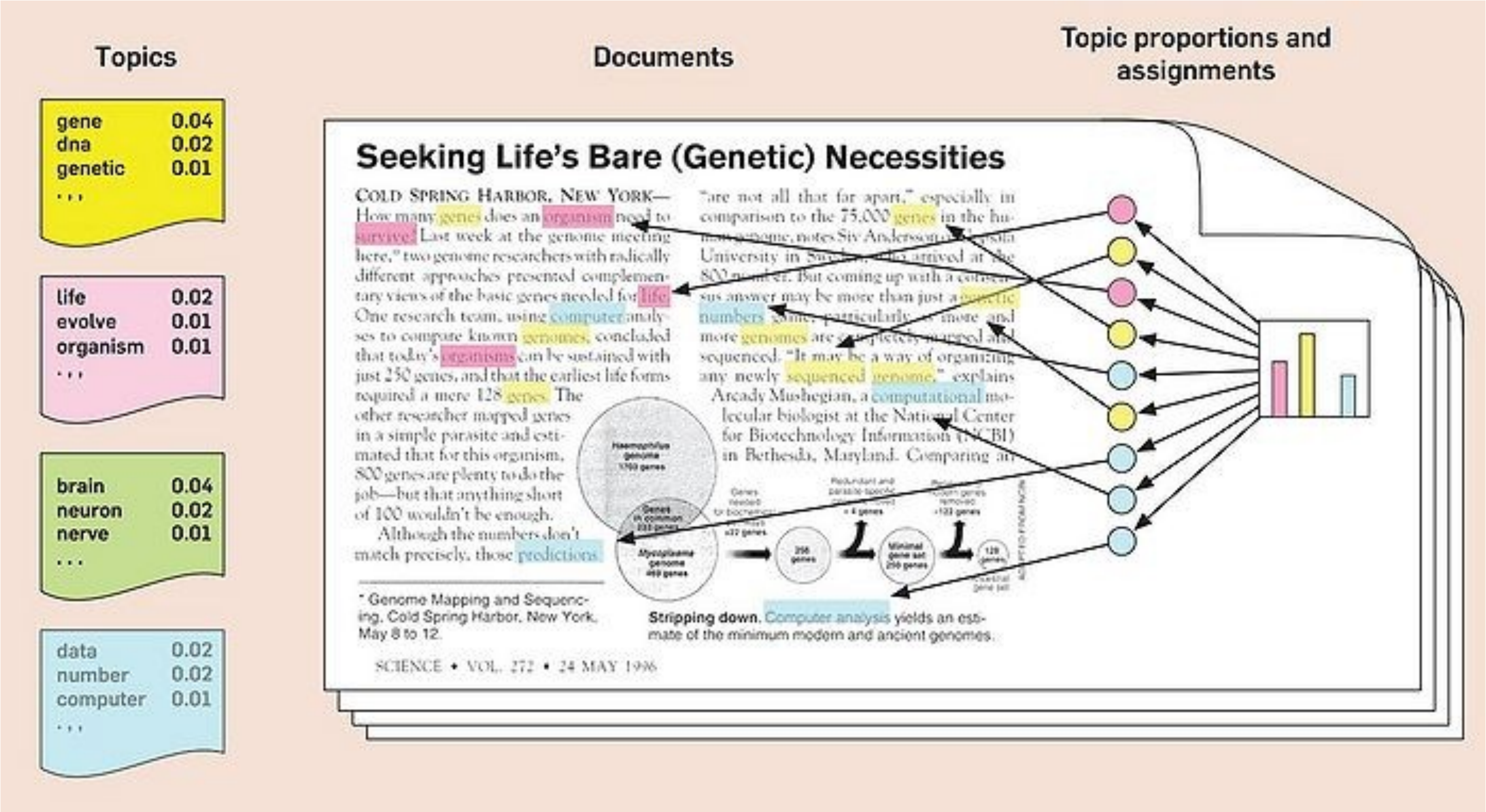




Clustering precipitation in Australia

Where is Clustering used?

- **Understanding Documents**



Where is Clustering Used?

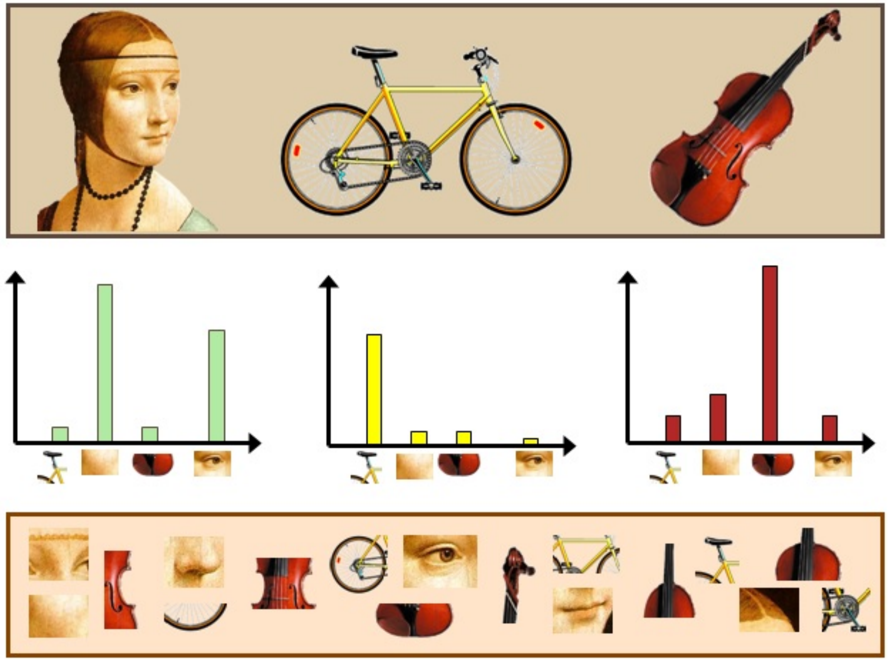
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Climate change: understanding earth climate, find patterns of atmospheric and ocean
- Finance: stock clustering analysis to uncover correlation among underlying shares
- Information retrieval/organization: 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

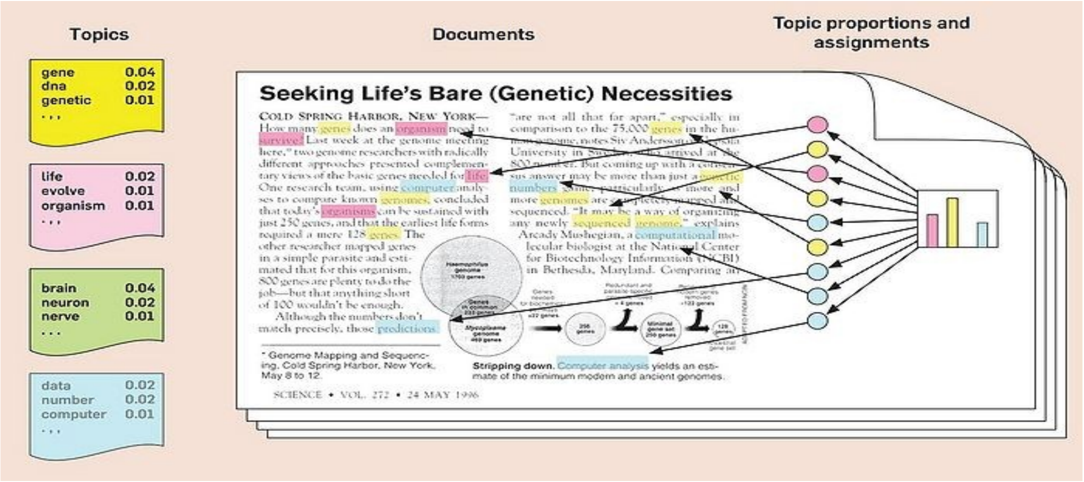
Clustering: Objectives

- 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



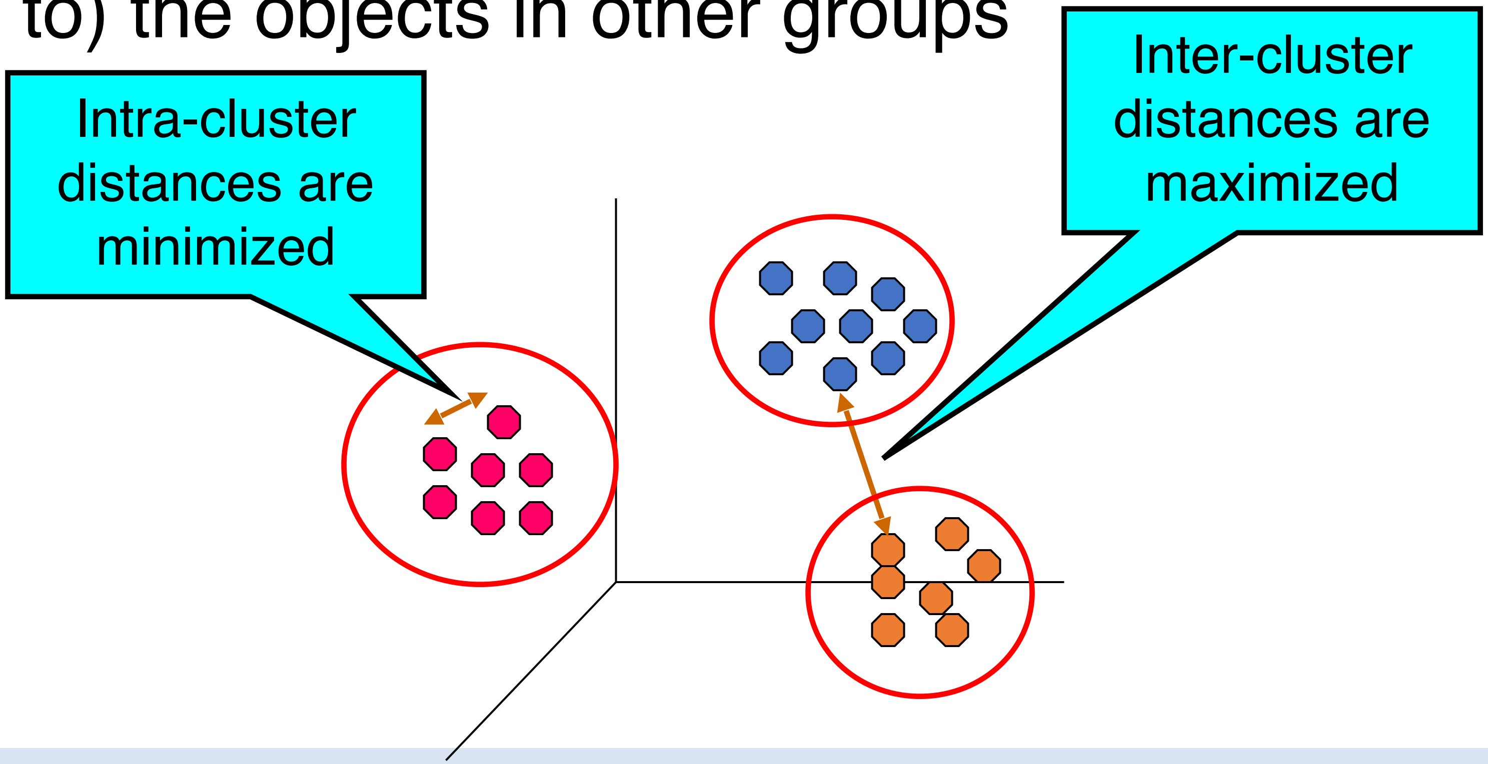


Types of Clustering Methods

- **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/Distribution-based: DBSCAN vs GMMs
  - Connectionist (vs) Centroid-based: k-means vs Hierarchical clustering

Clustering Methods

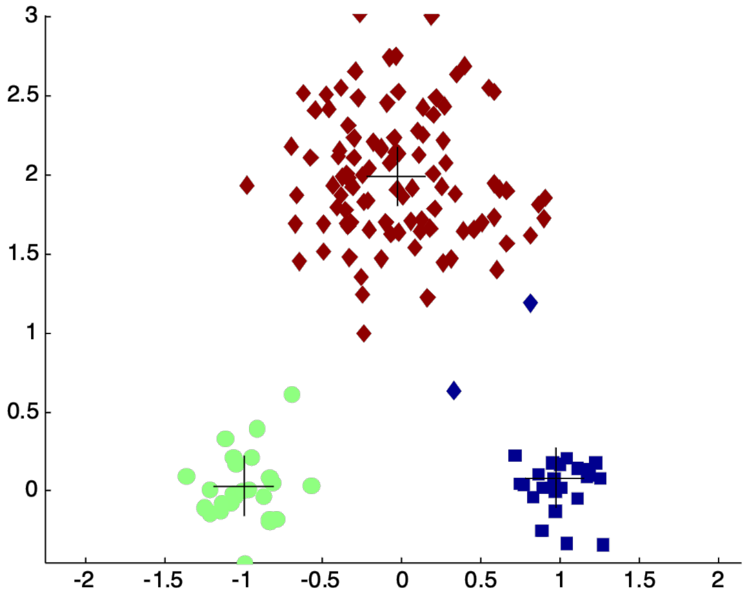
- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups
- How?



Outline

- K-Means
- Hierarchical Clustering
- Model-based Clustering (GMM and Expectation Maximization)
- Evaluation of Clustering Algorithms

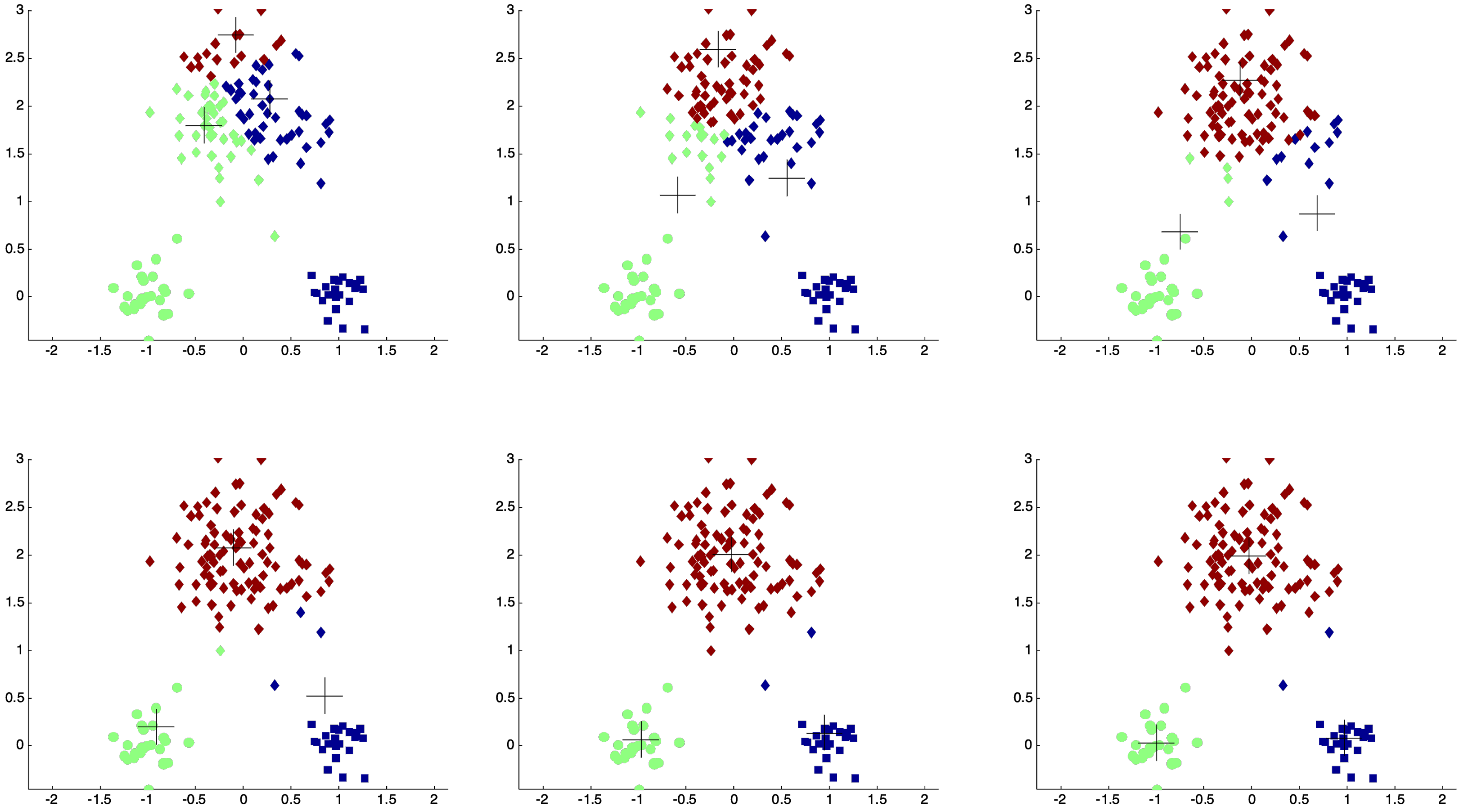
k-Means Clustering



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
- chicken-and-egg problem
- Number of clusters, K, must be specified
- The basic algorithm is very simple

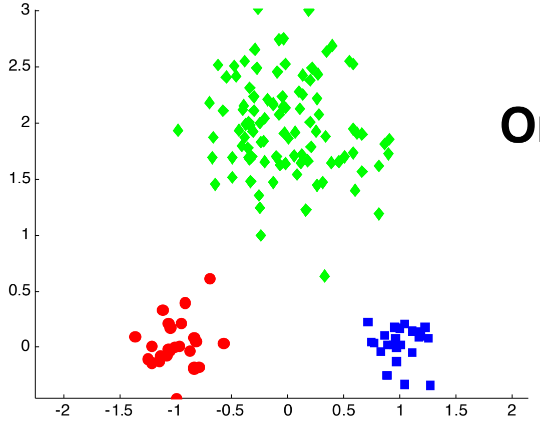
k-Means: Illustration



k-Means Clustering

- Initial centroids are often chosen randomly.
  - Clusters produced can 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 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?

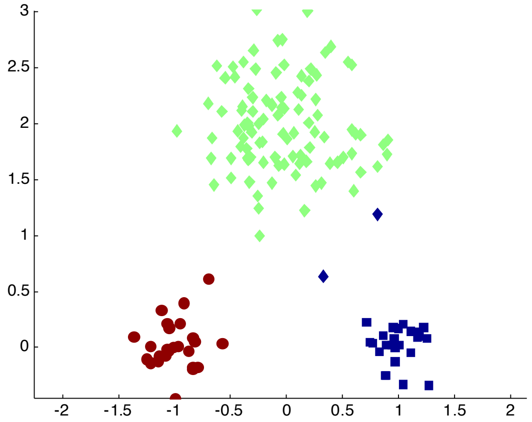
Two different k-Means clusterings

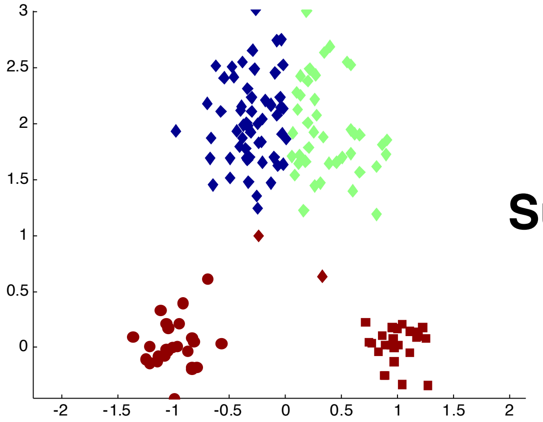


Original Points

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

Possible Solutions

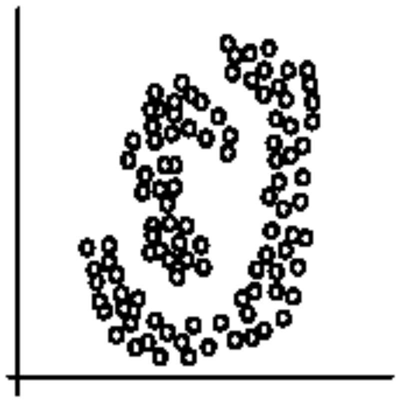
- Multiple runs
  - Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
  - Select most widely separated

Evaluating k-Means Clusters

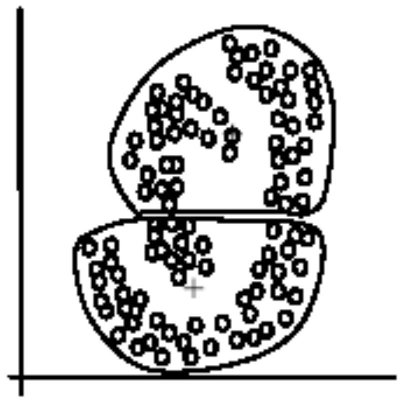
- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.
  - x is a data point in cluster Ci and m_i is the representative point for cluster Ci
  - Can show that m_i corresponds to the center (mean) of the cluster
- Given two clusterings, 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

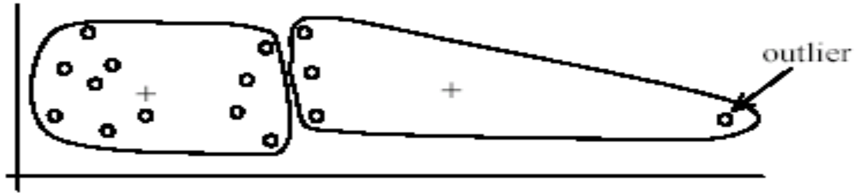
- k-Means has problems when clusters are of differing
  - Sizes, Densities, Non-globular shapes
- Sensitive to outliers
- The number of clusters (K) is difficult to determine



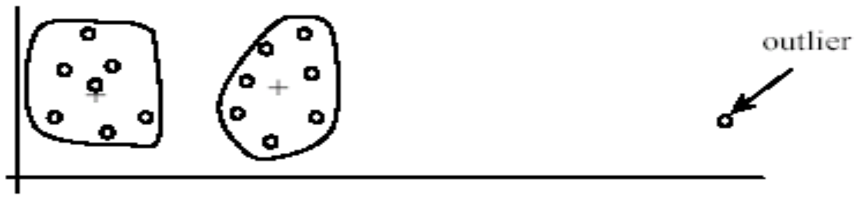
(A): Two natural clusters



(B): *k*-means clusters



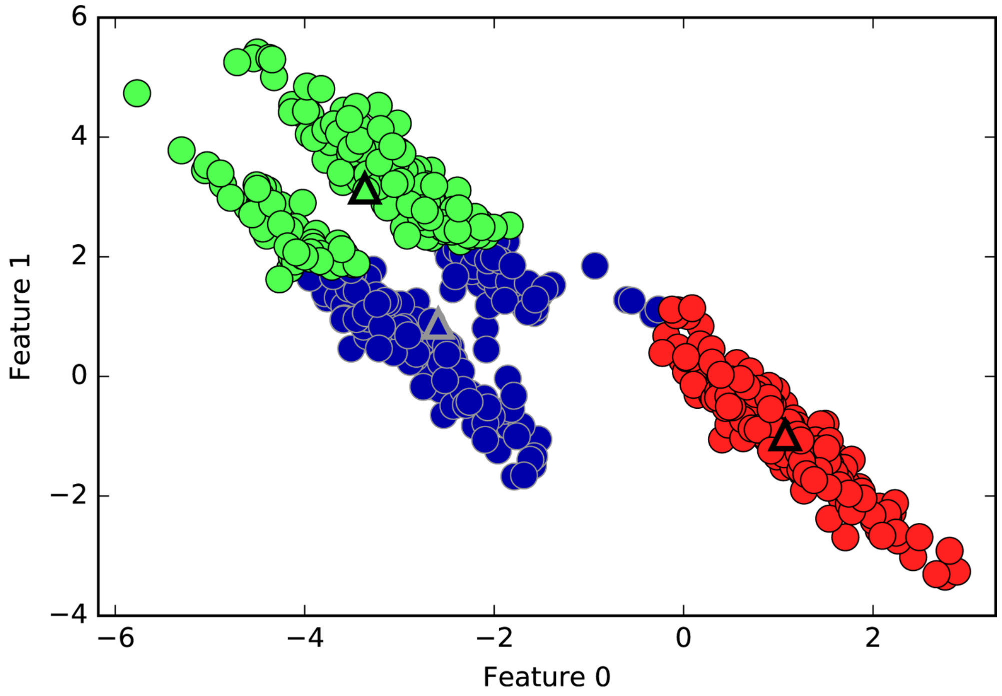
(A): Undesirable clusters

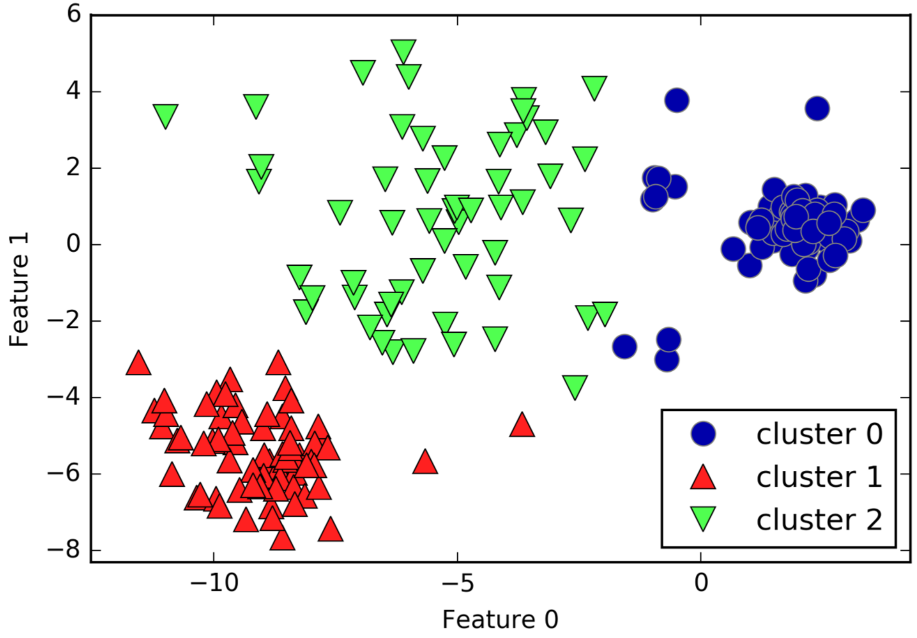


(B): Ideal clusters

Limitations

- k-Means has problems when clusters are of differing
  - Sizes, Densities, Non-globular shapes





Extensions

- Use of various distance metrics
  - Euclidean distance
  - Manhattan (city-block) distance
  - Cosine distance
  - Chebyshev distance

K-Means as optimization problem

- minimizes the sum of squared distances from the mean to every point in the data.

$$\mathcal{L}(z, \mu; \mathbf{D}) = \sum_n \left|\left| x_n - \mu_{z_n} \right|\right|^2 = \sum_k \sum_{n: z_n = k} ||x_n - \mu_k||^2$$

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

Algorithm 35 K-MEANS($\mathbf{D}$, K)

```
for k = 1 to K do
   μ_k ← some random location          // randomly initialize mean for kth cluster
end for
repeat
   for n = 1 to N do
      z_n ← argmin_k ||μ_k − x_n||      // assign example n to closest center
   end for
   for k = 1 to K do
      μ_k ← MEAN({ x_n : z_n = k })     // re-estimate mean of cluster k
   end for
until converged
return z                                // return cluster assignments
```

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

$$\frac{dJ}{d\mu_k} = 2 \sum_{n=1}^{N} r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k) = 0$$

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}.$$

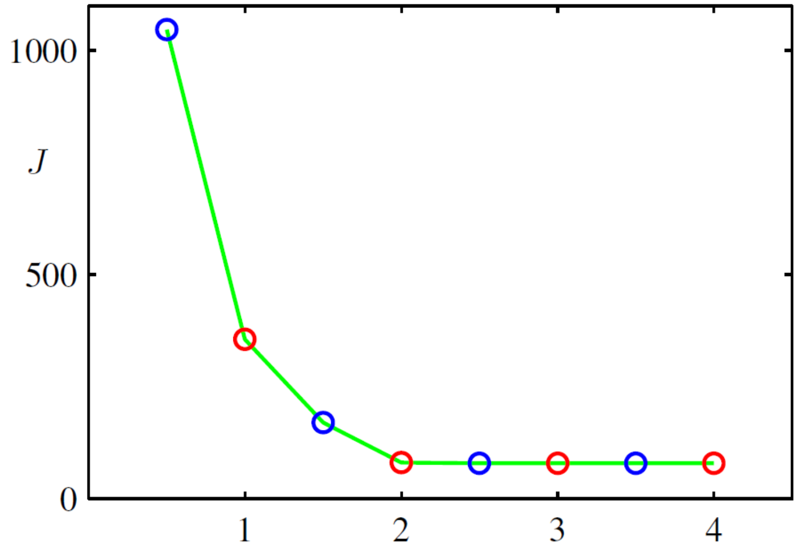
K-Means as optimization problem

- minimizes the sum of squared distances from the mean to every point in the data.

$$\mathcal{L}(z, \boldsymbol{\mu}; \mathbf{D}) = \sum_{n} \left|\left| \boldsymbol{x}_n - \boldsymbol{\mu}_{z_n} \right|\right|^2 = \sum_{k} \sum_{n:z_n=k} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2 \qquad J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

Algorithm 35 K-MEANS($\mathbf{D}$, K)

for $k = 1$ **to** K **do**
  $\boldsymbol{\mu}_k \leftarrow$ some random location — // randomly initialize mean for kth cluster
end for
repeat
  for $n = 1$ **to** N **do**
    $z_n \leftarrow \text{argmin}_k ||\boldsymbol{\mu}_k - \boldsymbol{x}_n||$ — // assign example n to closest center
  end for
  for $k = 1$ **to** K **do**
    $\boldsymbol{\mu}_k \leftarrow \text{MEAN}(\{ \boldsymbol{x}_n : z_n = k \})$ — // re-estimate mean of cluster k
  end for
until converged
return z — // return cluster assignments



Choosing number of clusters : Elbow plot in Clustering