Machine Learning
DSE 2254

Rohini R. Rao & Manjunath Hegde
Department of Data Science and Computer Applications
MIT, Manipal

March 2022

Slide - Set 5 – Clustering

Unsupervised Learning

- Data with no target attribute. Describe hidden structure from unlabeled data.
- Explore the data to find some intrinsic structures in them.

- Clustering: the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar to each other than to those in other clusters.
- Useful for
  - Automatically organizing data.
  - Understanding hidden structure in data.
  - Preprocessing for further analysis.

Applications

- Biology: classification of plants and animal kingdom given their features
- Marketing: Customer Segmentation based on a database of customer data containing their properties and past buying records
- Clustering weblog data to discover groups of similar access patterns.
- Recognize communities in social networks.

Aspects of Clustering

- A clustering algorithm such as
  - Partitional clustering eg, kmeans
  - Hierarchical clustering eg, AHC
  - Mixture of Gaussians
- A distance or similarity function
  - such as Euclidean, Minkowski, cosine
- Clustering quality
  - Inter-clusters distance ⇒ maximized
  - Intra-clusters distance ⇒ minimized

The quality of a clustering result depends on the algorithm, the distance function, and the application.

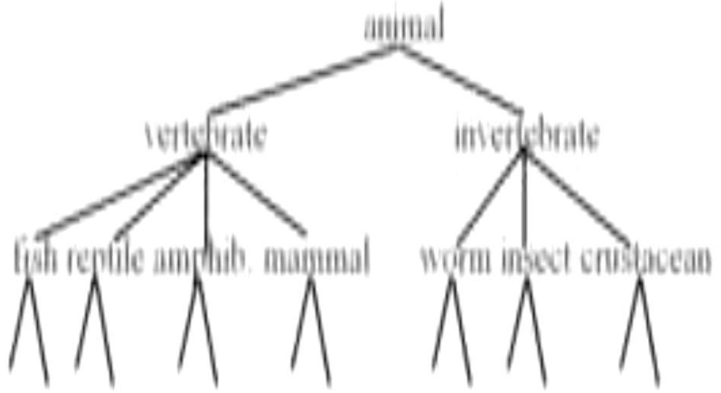
Major Clustering Approaches

- Partitioning: Construct various partitions and then evaluate them by some criterion
- Hierarchical: Create a hierarchical decomposition of the set of objects using some criterion
- Model-based: Hypothesize a model for each cluster and find best fit of models to data
- Density-based: Guided by connectivity and density functions
- Graph-Theoretic Clustering

Partitioning Algorithms

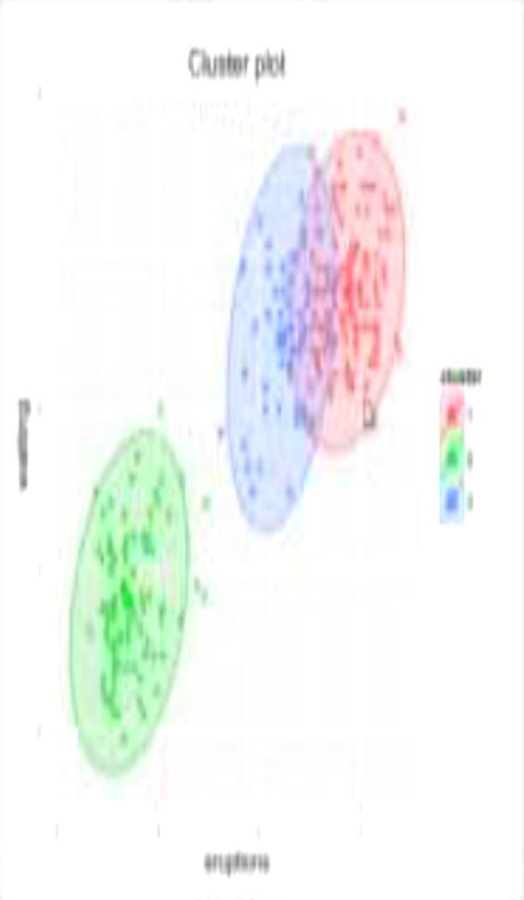
- <u>Partitioning method:</u> Construct a partition of a database *D* of *m* objects into a set of *k* clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic method: *k-means* (MacQueen, 1967)

Hierarchical Clustering



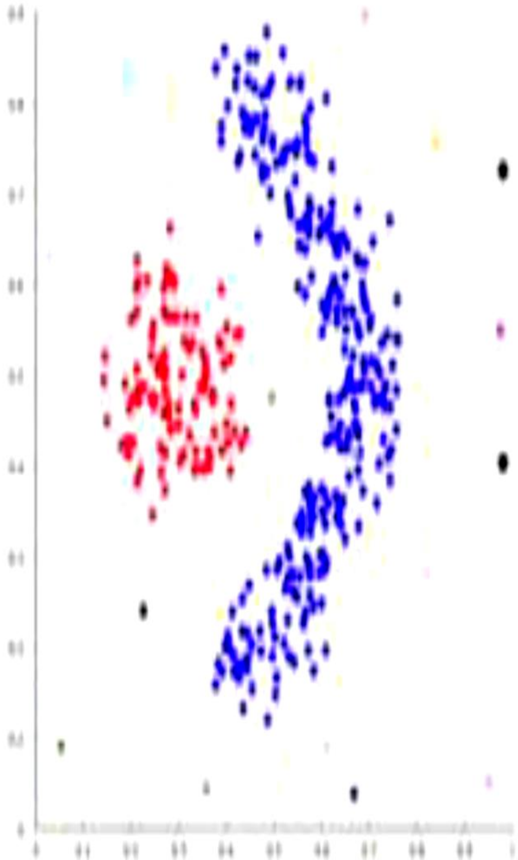
- Produce a nested sequence of clusters.
- One approach: recursive application of a partitional clustering algorithm.

Model Based Clustering



- A model is hypothesized
- e,g., Assume data is generated by a mixture of underlying probability distributions
- Fit the data to model

Density based Clustering



- Based on density connected points
- Locates regions of high density separated by regions of low density
- e.g., DBSCAN

Graph Theoretic Clustering

- Weights of edges between items (nodes) based on similarity
- E.g., look for minimum cut in a graph

(Dis) Similarity Measures

- Euclidean Distance
  - $X_1 = (x_{11}, x_{12}, \ldots x_{1n})$ and $X_2 = (x_{21}, x_{22} .. x_{2n})$ is
  - $\text{dist}(X_1, X_2) = \sqrt{\Sigma_{1=1}^{n} (x_{1i} - x_{2i})^2}$
- Manhattan or City Block Distance
  - $\text{dist}(i, j) = | x_{i1} - x_{j1} | + | x_{i2} - x_{j2} | + \ldots | x_{in} - x_{jn} |$
- Minkowski Distance
  - $\text{dist}(i,j) = (| x_{i1} - x_{j1} |^p + | x_{i2} - x_{j2} |^p + \ldots | x_{in} - x_{jn} |^p)^{1/p}$

- Correlation coefficients (scale-invariant)
- Mahalanobis distance

$$d(x_i, x_j) = \sqrt{(x_i - x_j)\Sigma^{-1}(x_i - x_j)}$$

- Pearson correlation

$$r(x_i, x_j) = \frac{Cov(x_i, x_j)}{\sigma_{x_i}\sigma_{x_j}}$$

Quality of Clusters

- Internal evaluation:
  - assign the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters, e.g., Davies-Bouldin index

$$DB = \frac{1}{n}\sum_{i=1}^{k} \max_{j \neq i} \frac{\sigma_i + \sigma_j}{d(c_i, c_j)}$$

- External evaluation:
  - evaluated based on data such as known class labels and external benchmarks, eg, Rand Index, Jaccard Index, f-measure

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN}$$

K-means Clustering

Given k

1. Randomly choose k data points (seeds) to be the initial cluster centres
2. Assign each data point to the closest cluster centre
3. Re-compute the cluster centres using the current cluster memberships.
4. If a convergence criterion is not met, go to 2.

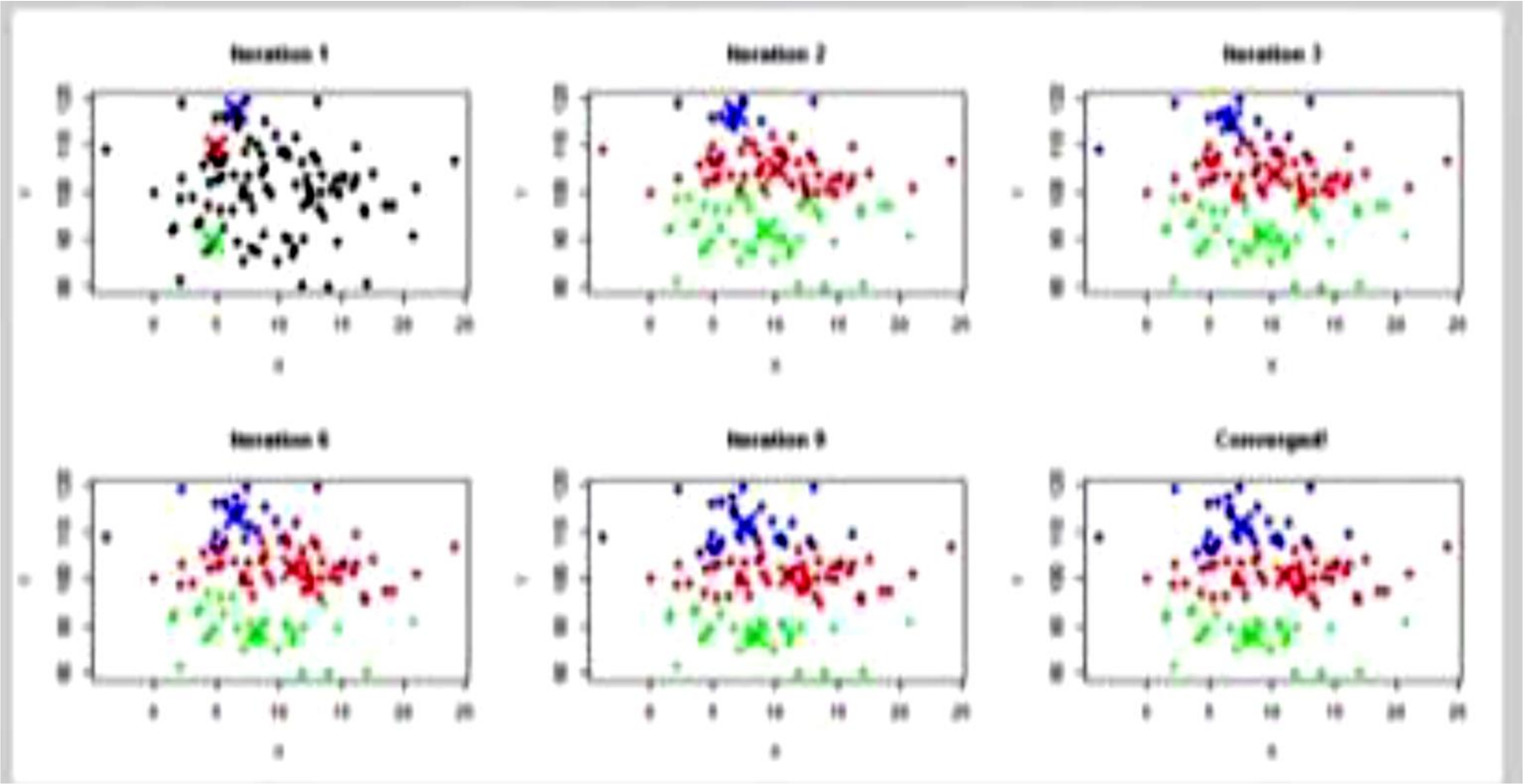
Stopping/convergence criterion

OR

1. no re-assignments of data points to different clusters
2. no (or minimum) change of centroids
3. minimum decrease in the *sum of squared error*

$$SSE = \sum_{i=1}^{k} \sum_{x \in S_i} \|x_i - \mu_i\|^2$$

K-means Illustrated



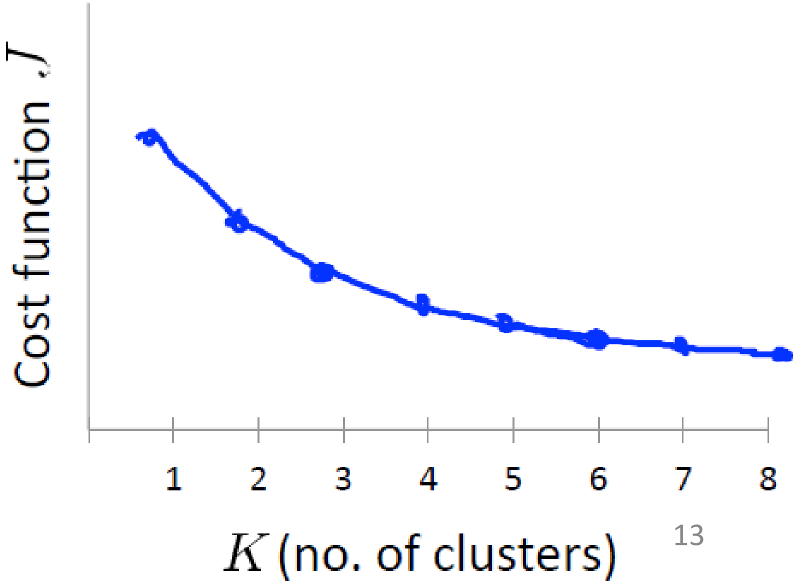
Optimization Objective

- Min $J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$

$c^{(i)}$ = index of cluster (1,2,…,K) to which example $x^{(i)}$ is currently assigned

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been

μ_k = cluster centroid k ($\mu_k \in \mathbb{R}^n$)



Advantages

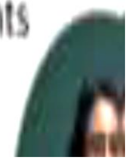
- Fast, robust easy to understand.
- Relatively efficient: O(tkmn)
- Gives best result when data set are distinct or well separated from each other.

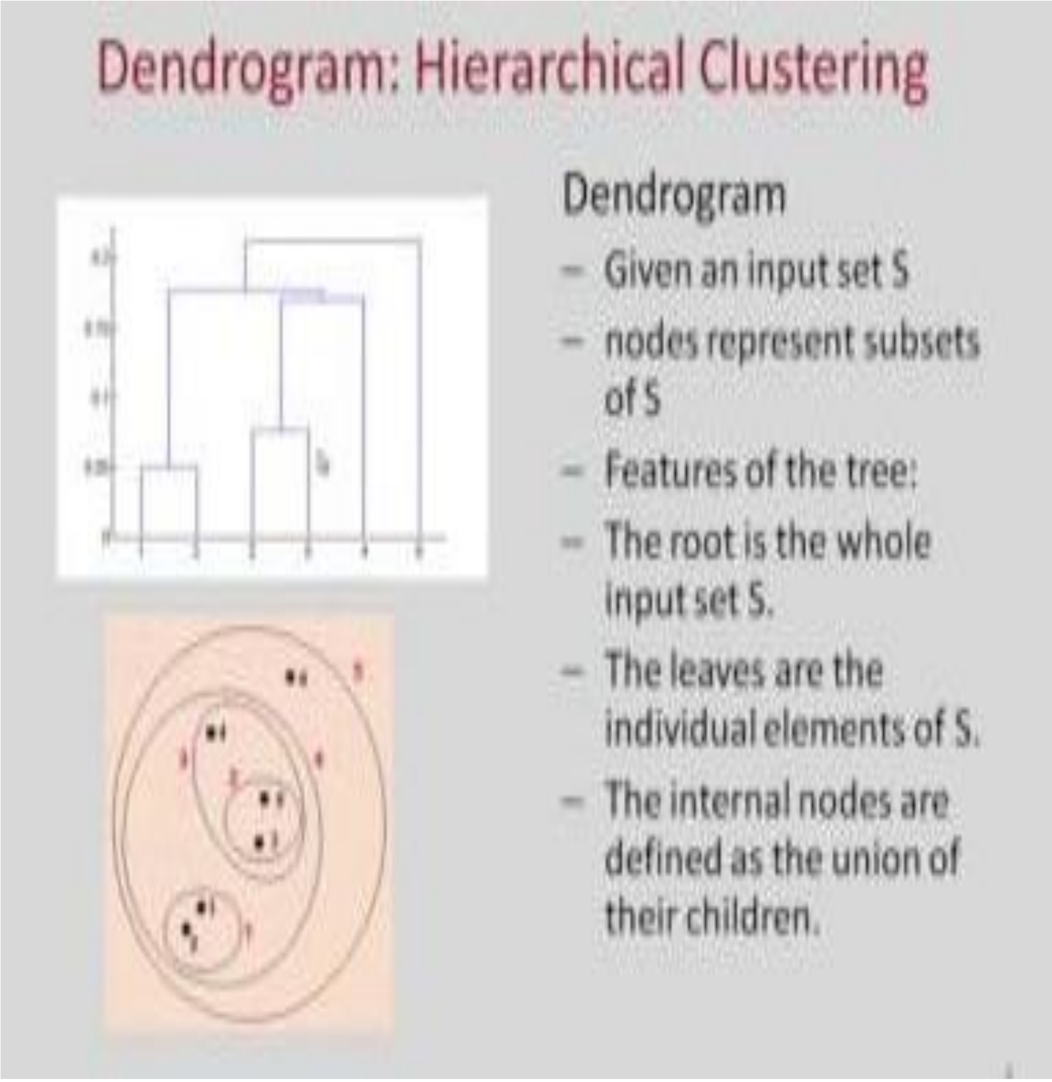
Disadvantages

- Requires apriori specification of the number of cluster centers.
- Hard assignment of data points to clusters
- Euclidean distance measures can unequally weight underlying factors.
- Applicable only when mean is defined i.e. fails for categorical data.
- Only local optima

Types of hierarchical clustering

- Agglomerative (bottom up) clustering: It builds the dendrogram (tree) from the bottom level, and
  - merges the most similar (or nearest) pair of clusters
  - stops when all the data points are merged into a single cluster (i.e., the root cluster).
- Divisive (top down) clustering: It starts with all data points in one cluster, the root.
  - Splits the root into a set of child clusters. Each child cluster is recursively divided further
  - stops when only singleton clusters of individual data points remain, i.e., each cluster with only a single point





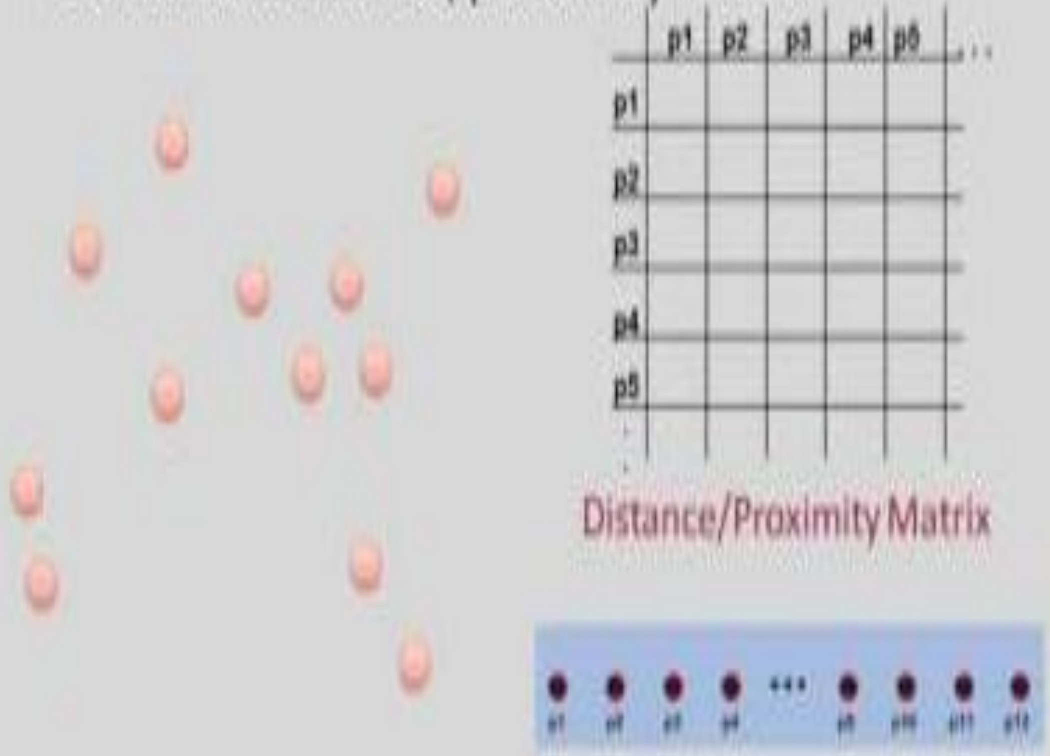
Hierrarchical Agglomerative clustering

- Initially each data point forms a cluster.
- Compute the distance matrix between the clusters.
- Repeat
  - Merge the two closest clusters
  - Update the distance matrix
- Until only a single cluster remains.

Different definitions of the distance leads to different algorithms.

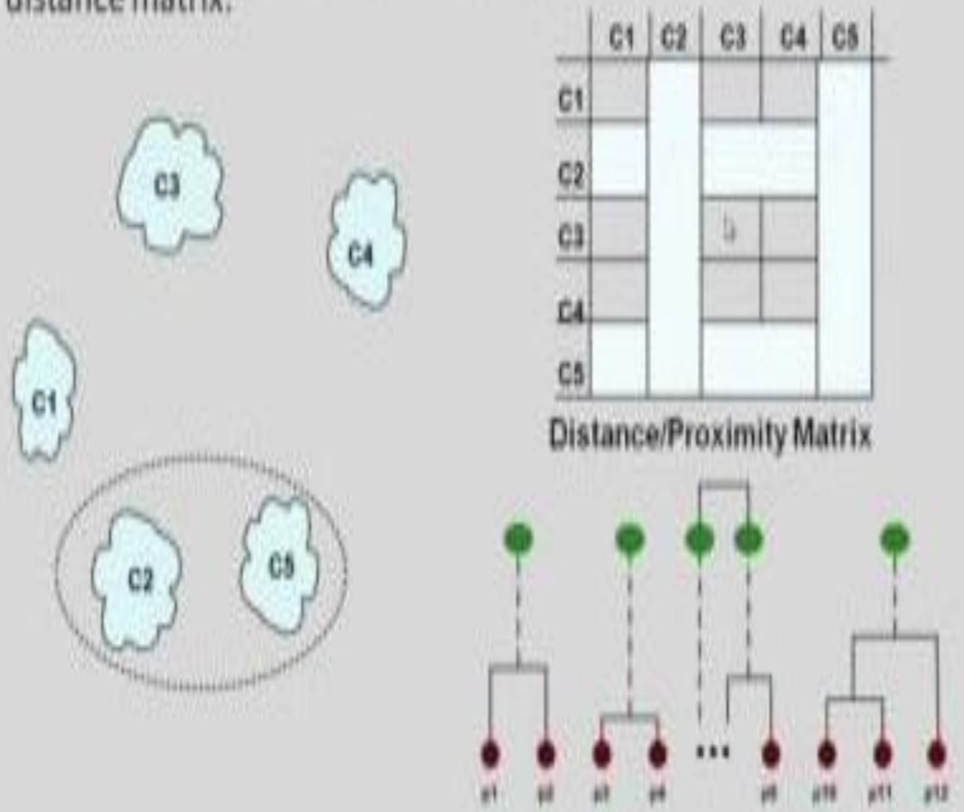
Initialization

- Each individual point is taken as a cluster
- Construct distance/proximity matrix



Intermediate State

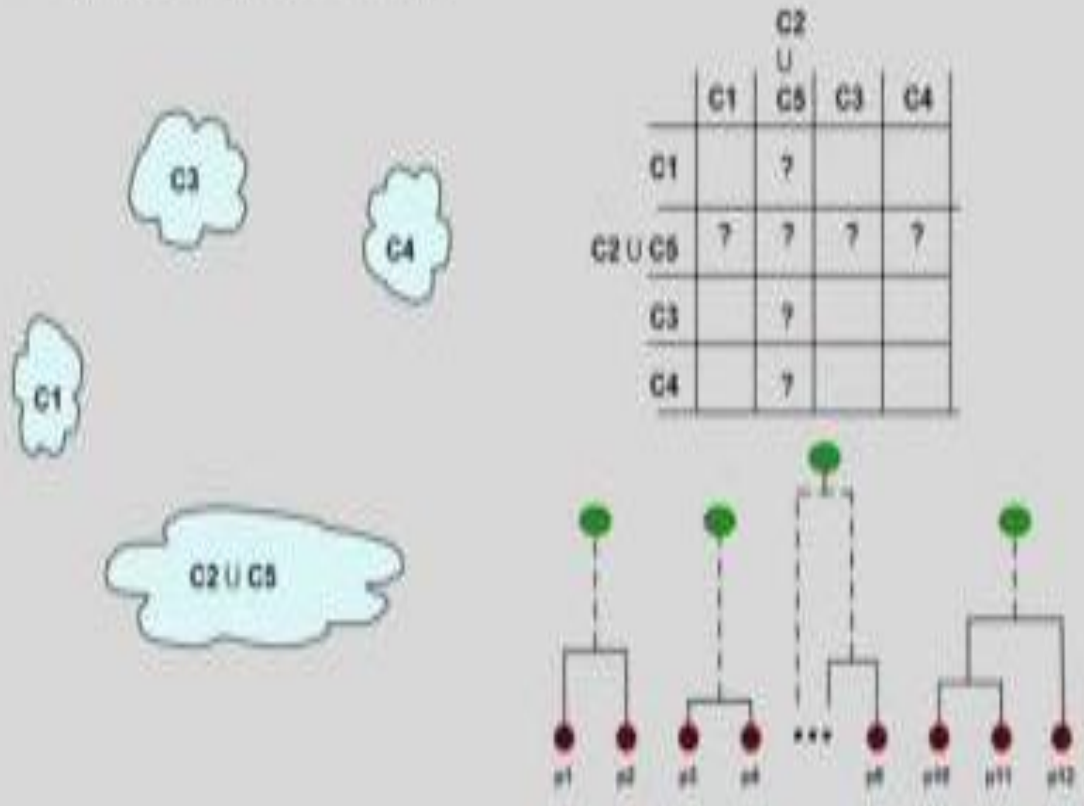
Merge the two closest clusters (C2 and C5) and update the distance matrix.



Distance/Proximity Matrix

After Merging

- Update the distance matrix



Closest Pair

- A few ways to measure distances of two clusters.
- **Single-link**
  - Similarity of the *most* similar (single-link)
- **Complete-link**
  - Similarity of the *least* similar points
- **Centroid**
  - Clusters whose centroids (centers of gravity) are the most similar
- **Average-link**
  - Average cosine between pairs of elements

Distance between two clusters

- Single-link distance between clusters C_i and C_j is the *minimum distance* between any object in C_i and any object in C_j

$$sim(C_i, C_j) = \max_{x \in C_i, y \in C_j} sim(x, y)$$

Complete link method

- The distance between two clusters is the distance of two furthest data points in the two clusters.

$$sim(c_i, c_j) = \min_{x \in c_i, y \in c_j} sim(x, y)$$

- Makes "tighter," spherical clusters that are typically preferable.
- It is sensitive to outliers because they are far away

Average Link Clustering

- Similarity of two clusters = average similarity between any object in Ci and any object in Cj

$$sim(c_i, c_j) = \frac{1}{|C_i||C_j|} \sum_{\vec{x} \in C_i} \sum_{\vec{y} \in C_j} sim(\vec{x}, \vec{y})$$

- Compromise between single and complete link. Less susceptible to noise and outliers.
- Two options:
  - Averaged across all ordered pairs in the merged cluster
  - Averaged over all pairs *between* the two original clusters

Example

Distance Matrix

	I1	I2	I3	I4	I5
I1	1.00	0.90	$\boxed{0.10}$	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	$\boxed{0.10}$	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00

Merged (I1, I3)

Iteration 1

$$\begin{array}{c|c}
 & \begin{array}{cccc} (I1,I3) & I2 & I4 & I5 \end{array} \\
\begin{array}{c} (I1,I3) \\ I2 \\ I4 \\ I5 \end{array} & \begin{bmatrix} 0 & 0.7 & 0.4 & \boxed{0.2} \\ 0.7 & 0 & 0.6 & 0.5 \\ 0.4 & 0.6 & 0 & 0.8 \\ \boxed{0.2} & 0.5 & 0.8 & 0 \end{bmatrix}
\end{array}$$

Merge

(I1, I3) & (I5)

Iteration 2

$$\begin{array}{c|c}
 & \begin{array}{ccc} (I1,I3,I5) & I2 & I4 \end{array} \\
\begin{array}{c} (I1,I3,I5) \\ I2 \\ I4 \end{array} & \begin{bmatrix} 0 & 0.5 & \boxed{0.4} \\ 0.5 & 0 & 0.6 \\ 0.4 & 0.6 & 0 \end{bmatrix}
\end{array}$$

Merge

(I1 I3, I5) & I4

Iteration 3

$$\begin{array}{c} \\ (I1, I3, I4, I5) \\ I2 \end{array} \begin{array}{c} \begin{array}{cc} (I1, I3, I4, I5) & I2 \end{array} \\ \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix} \end{array}$$

Merge I2 $\Rightarrow$ 1 cluster

{I1, I2, I3, I4, I5}

Visualization

DENDROGRAM

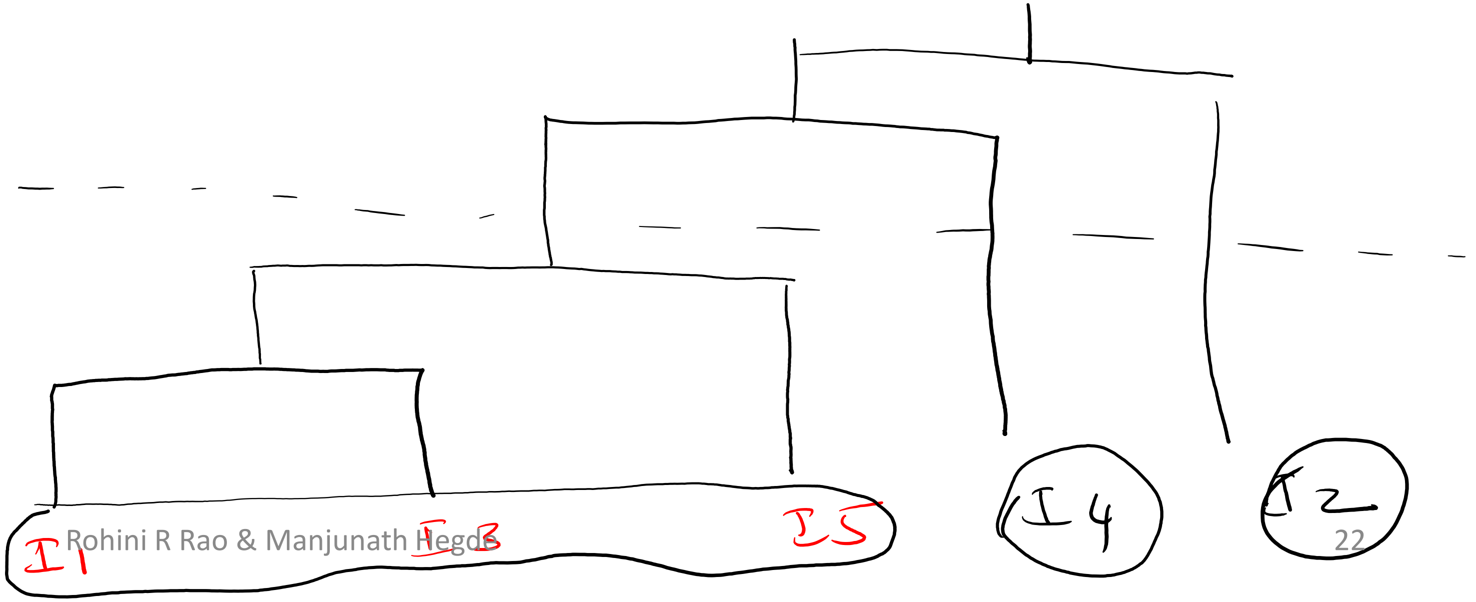
Step 4

Step 3

Step 2

Step 1

Step 0



392 cars – Agglomerative clustering Euclidean distance & average linkage

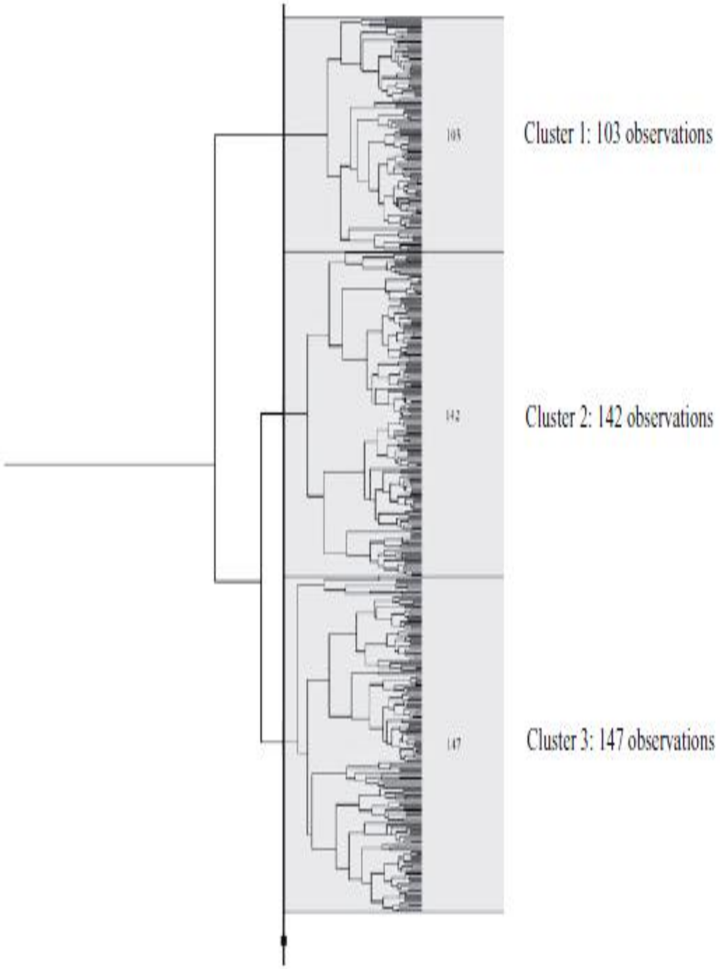
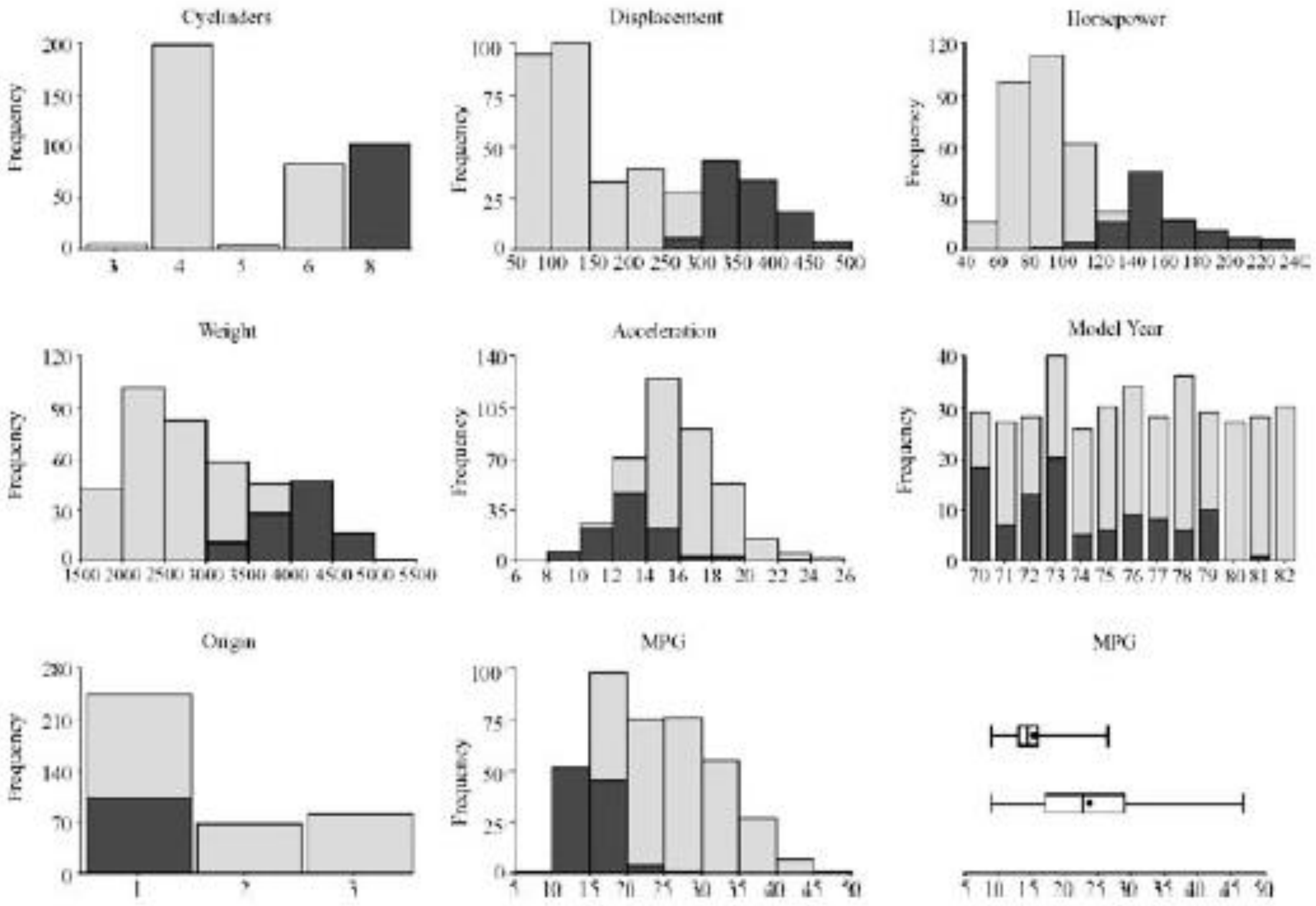


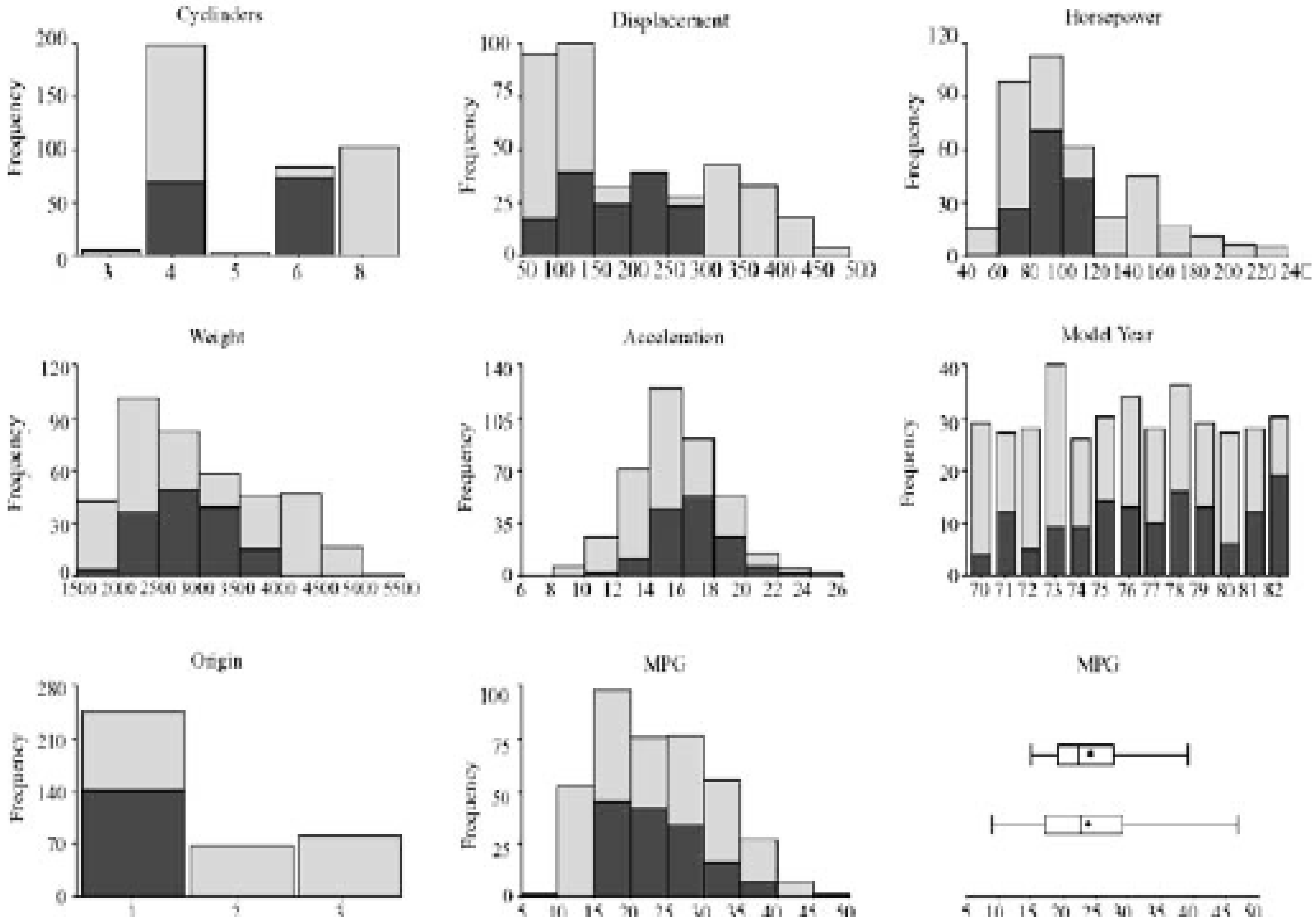
Table 6.8. Table of car observations

Names	Cylinders	Displace- ment	Horse power	Weight	Accele- ration	Model/ Year	Origin	MPG
Chevrolet Chevelle Malibu	8	307	130	3,504	12	1970	1	18
Buick Skylark 320	8	350	165	3,693	11.5	1970	1	15
Plymouth Satellite	8	318	150	3,436	11	1970	1	18
Amc Rebel SST	8	304	150	3,433	12	1970	1	16
Ford Torino	8	302	140	3,449	10.5	1970	1	17
Ford Galaxie 500	8	429	198	4,341	10	1970	1	15
Chevrolet Impala	8	454	220	4,354	9	1970	1	14
Plymouth Fury III	8	440	215	4,312	8.5	1970	1	14

Result - Summary of Cluster 1



Result - Summary of Cluster 2



Computational Complexity

- In the first iteration, all HAC methods need to compute similarity of all pairs of *N* initial instances, which is $O(N^2)$.
- In each of the subsequent *N*–2 merging iterations, compute the distance between the most recently created cluster and all other existing clusters.
- In order to maintain an overall $O(N^2)$ performance, computing similarity to each other cluster must be done in constant time.
  - Often $O(N^3)$ if done naively or $O(N^2 \log N)$ if done more cleverly

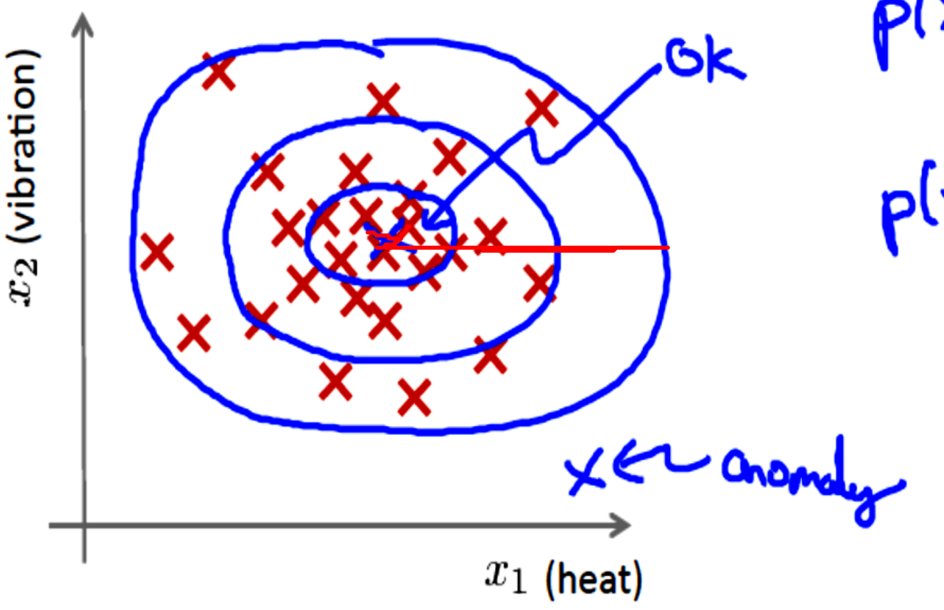
The complexity

- All the algorithms are at least $O(n^2)$. n is the number of data points.
- Single link can be done in $O(n^2)$.
- Complete and average links can be done in $O(n^2 \log n)$.
- Due the complexity, hard to use for large data sets.

Anomaly Detection

- Dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$
- Is x_{test} anomalous?

Model $p(x)$.



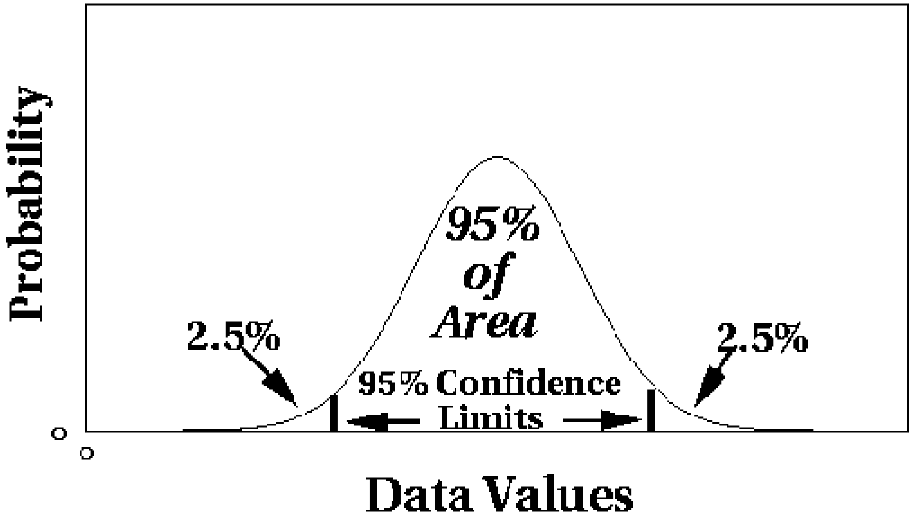
$p(x_{test}) < \varepsilon \rightarrow$ flag anomaly

$p(x_{test}) \geq \varepsilon \rightarrow$ Ok

What Is
Outlier Discovery or Anomaly Detection?

- What are outliers?
  - The set of objects are considerably dissimilar from the remainder of the data
  - Can be measurement or execution error.
- Problem
  - Given a set of n data points , Find top k outlier points that are considerably dissimilar with respect to remaining data
- Applications:
  - Credit card fraud detection
  - Telecom fraud detection
  - Customer segmentation
  - Medical analysis
- Type of outliers
  - Global Outliers
  - Contextual Outliers
  - Collective Outliers

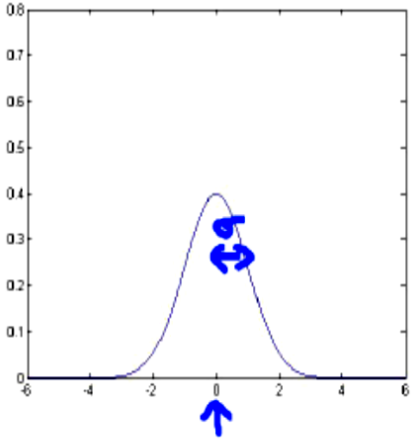
Anomaly Detection: Statistical Approaches



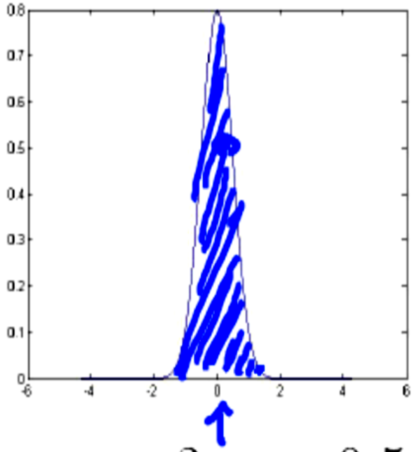
- Assume a model underlying distribution that generates data set (e.g. normal distribution)
- Use discordancy tests depending on
  - data distribution
  - Examines two hypotheses :
    - Working hypothesis - H : Oi belongs to F
    - Alternative hypothesis - H : Oi belongs to G
  - 2 procedures to detect outliers
    - Block procedure – all suspect objects are treated as outliers
    - Sequential procedure – Object least likely is tested first. If it is an outlier all extreme values are also outliers
  - distribution parameter (e.g., mean, variance) for the distribution must be detected
  - number of expected outliers
- Drawbacks
  - most tests are for single attribute
  - In many cases, data distribution may not be known

Gaussian distribution example

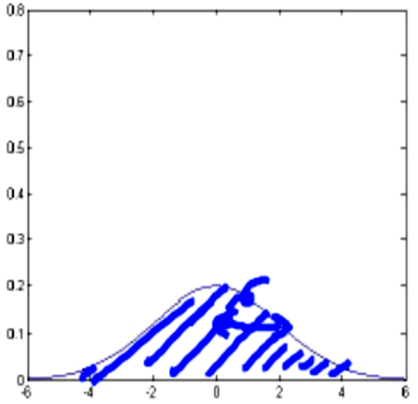
$\mu = 0, \sigma = 1$



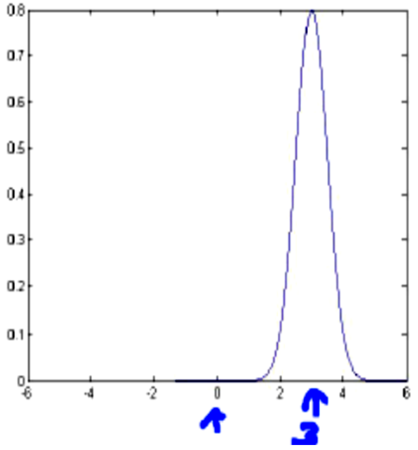
$\mu = 0, \sigma = 0.5$



$\mu = 0, \sigma = 2$



$\mu = 3, \sigma = 0.5$



Anomaly Detection Algorithm

1 Choose features Xi that are indicative of anomalous examples

2. Fit parameters $\mu_1, \ldots, \mu_n, \sigma_1^2, \ldots, \sigma_n^2$

$$\boxed{\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}}$$

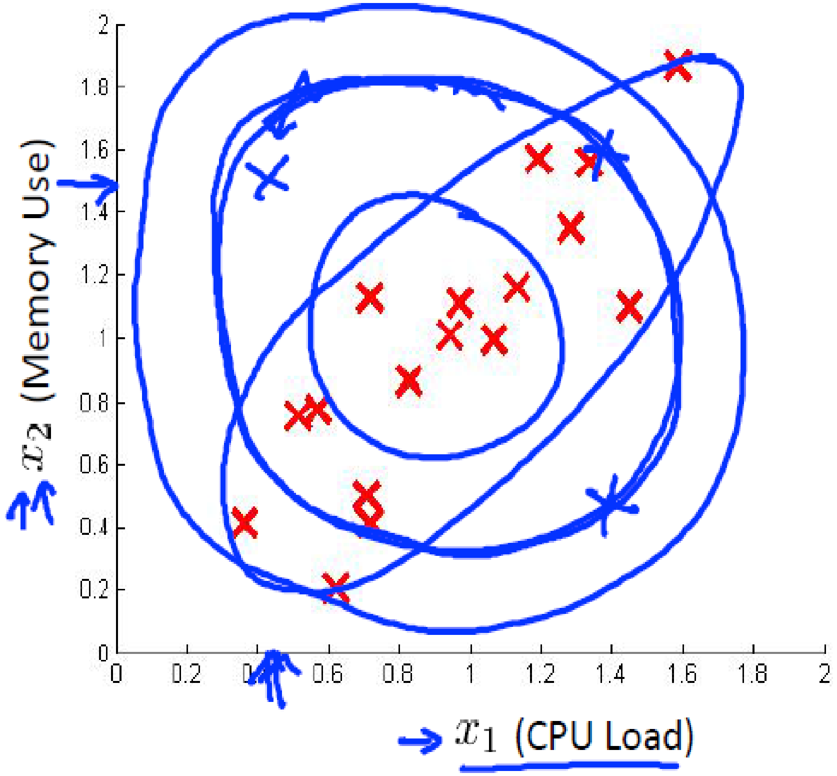
$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(x_j^{(i)} - \mu_j)^2$$

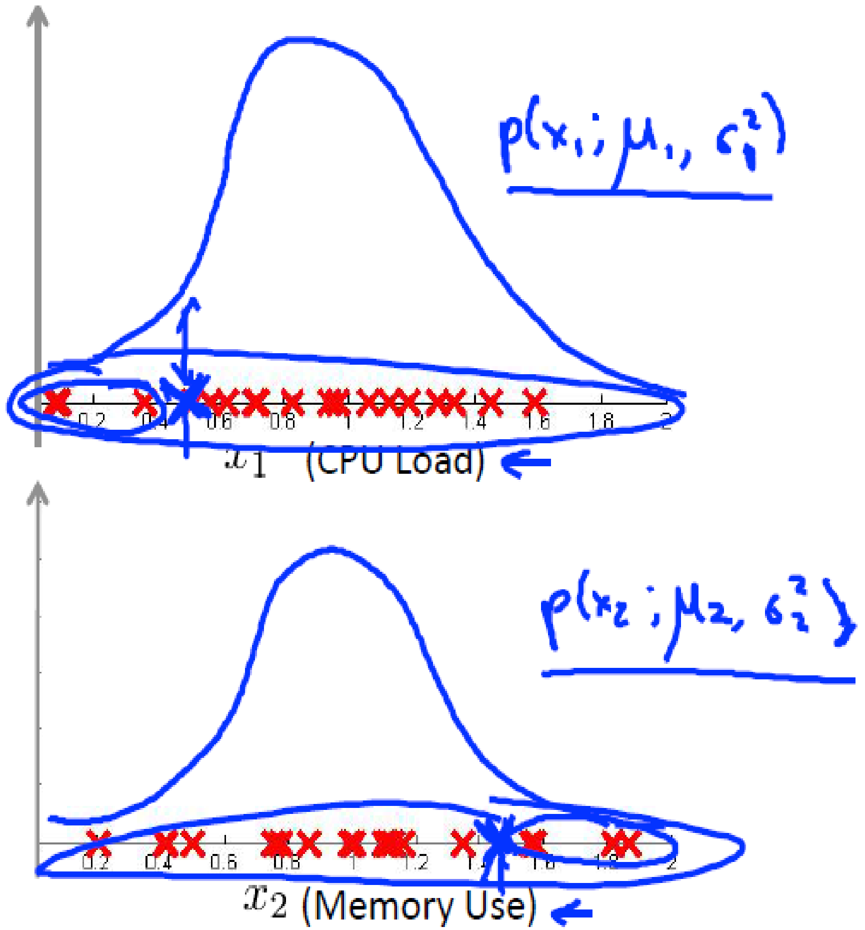
3. Given new example x, compute $p(x)$:

$$p(x) = \prod_{j=1}^{n} p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right)$$

4. Anomaly if $p(x) < \varepsilon$

Multivariate Anomaly Detection

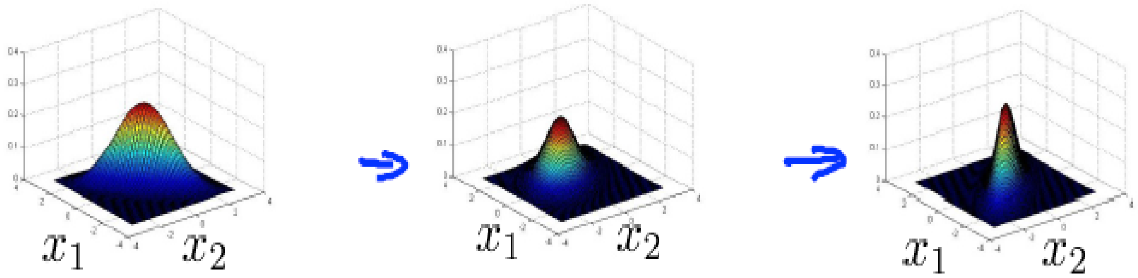




Multivariate Gaussian (Normal) distribution

Parameters μ, Σ $\quad \mu \in \mathbb{R}^n \quad \Sigma \in \mathbb{R}^{n \times n}$

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$



Parameter fitting:

Given training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$ $\quad x \in \mathbb{R}^n$

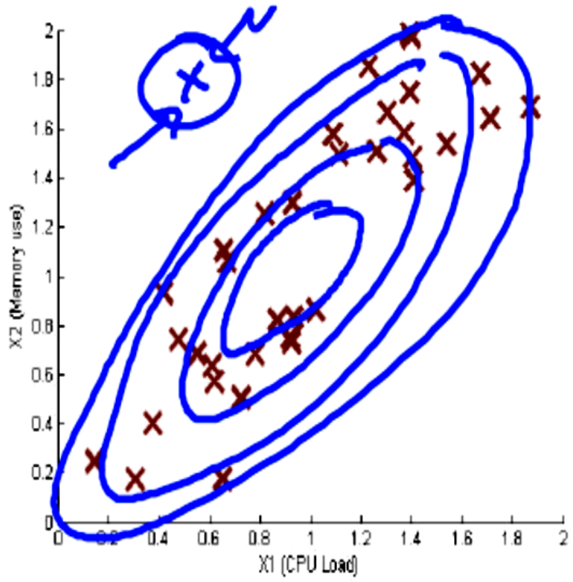
$$\boxed{\mu} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} \qquad \boxed{\Sigma} = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)}-\mu)(x^{(i)}-\mu)^T$$

Anomaly detection with the multivariate Gaussian

1. Fit model $p(x)$ by setting

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu)(x^{(i)} - \mu)^T$$



2. Given a new example x, compute

$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

Flag an anomaly if $p(x) < \varepsilon$

Outlier Discovery: Distance-Based Approach

- Introduced to counter the main limitations imposed by statistical methods
  - We need multi-dimensional analysis without knowing data distribution.
- Distance-based outlier: A DB(p, D)-outlier is an object O in a dataset T such that at least a fraction p of the objects in T lies at a distance greater than D from O
- For ex : Objects that lie 3 or more std. deviations from the mean.
- Algorithms for mining distance-based outliers
  - Index-based algorithm
    - Uses multidimensional indexing structures such as R trees to search for neighbors of each object o within radius dmin around the object.
    - Let M be the maximum number of objects within the dmin neighborhood.
    - Once M+1 neighbors are found , o is not an outlier.
    - Searches for neighbors of each object O within radius dmin around the object.
    - Time complexity – $O(n^2k)$.

Outlier Discovery: Density based Local Outlier Detection

- Previous methods depend on overall or global distribution
- Local outliers if the data point is outlying relative to its local neighborhood with respect to the density of the neighborhood.
- Outliers are not binary properties ,(Linear Outlier Factor) an assessment of the degree to which an object is an outlier
- K-distance of an object p is the maximal distance that p gets from its k-nearest neighbors. K can be minimum points in the neighborhood.

Outlier Discovery: Deviation-Based Approach

- Identifies outliers by examining the main characteristics of objects in a group
- Objects that “deviate” from this description are considered outliers
- sequential exception technique
  - simulates the way in which humans can distinguish unusual objects from among a series of supposedly like objects
  - Derives exception set – set of deviations or outliers – smallest subset of objects whose removal results in the greatest reduction of dissimilarity in the residual set.
  - Dissimilarity functions can be used instead of similarity
  - Ex :Variance
- OLAP data cube technique
  - uses data cubes to identify regions of anomalies in large multidimensional data