Advanced Analytical Theory and Methods: Cluster Analysis

Prof. Dr. Shamim Akhter

Professor, Dept. of CSE

Ahsanullah University of Science and Technology

What is it?

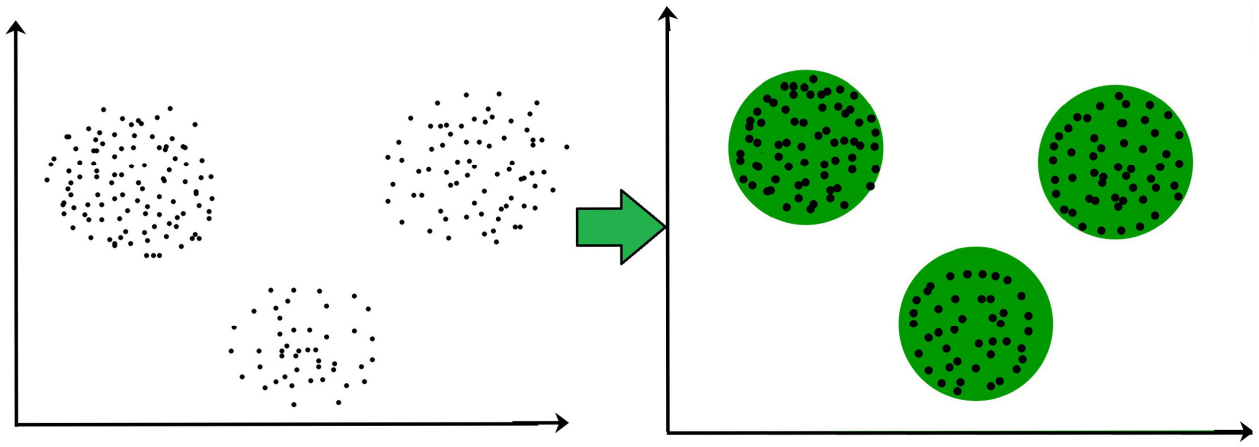
Cluster

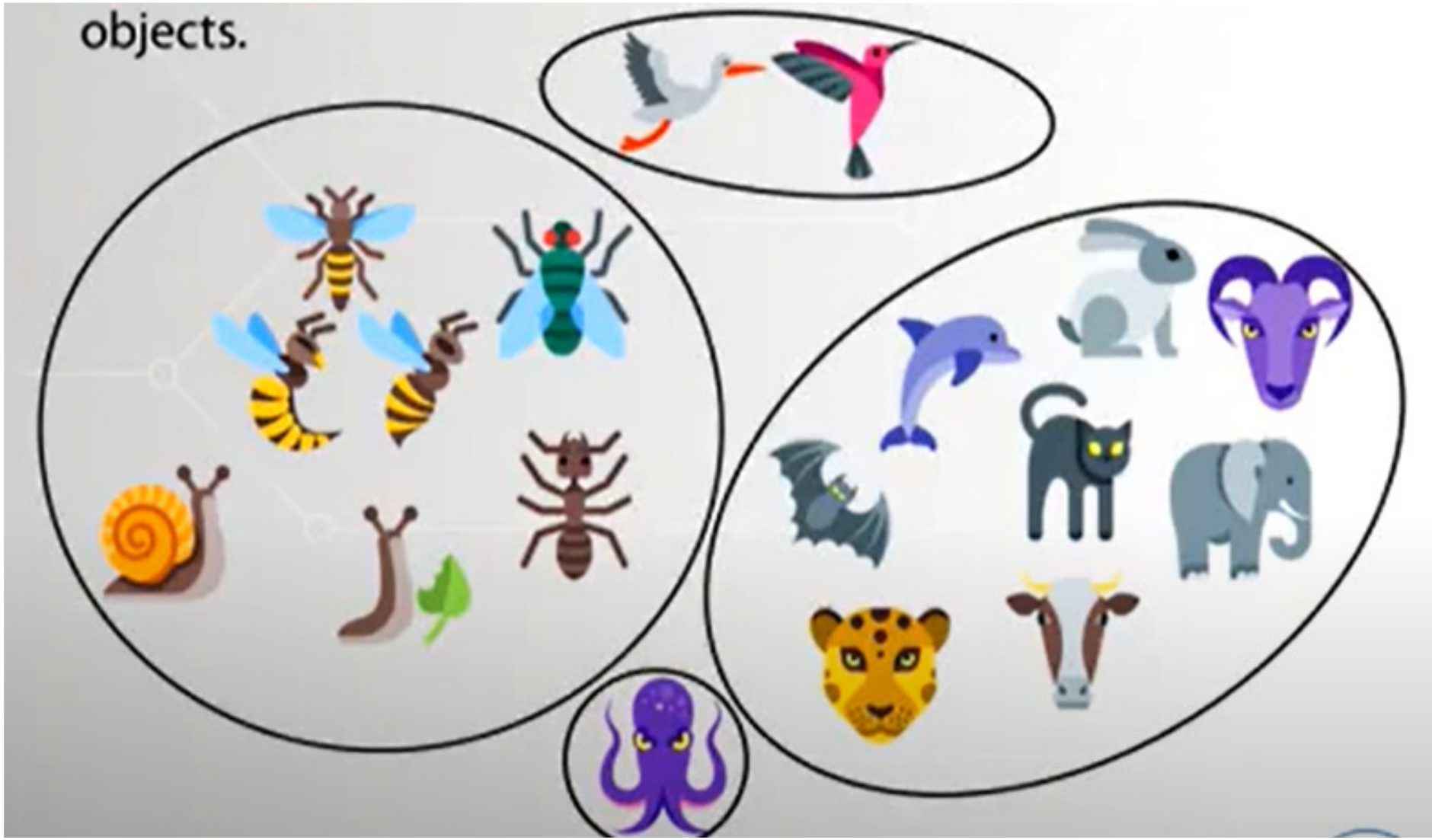
- A cluster is a collection of objects “similar” between them and “dissimilar” to the objects belonging to other clusters.

Clustering

- Given a finite set of data(X), the problem of clustering in X is to find several cluster centers that are required to form a partition of X such that the degree of association is strong for data within blocks of the partition and weak for data in different blocks.

Machine learning defines data clustering as **unsupervised learning.**




objects.

Classical Clustering Approaches

Classical clustering algorithms find a **"hard partition"** of a given dataset based on certain criteria that evaluate the goodness of a partition.

- "hard partition" means that each datum belongs to exactly one partition cluster.

The concept of "hard partition" is defined as follows:

Definition 1 (Hard Partition): Let X be a set of data, and x_1 be an element of X. A partition $P = \{C_1, C_2, \ldots, C_k\}$ of X is "hard" if and only if the following two condition hold.

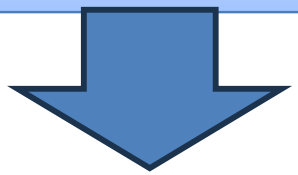
(i) for all $x_i \in X$ there exists a partition $C_i \in P$ such that $x_i \in C_j$.

(ii) for all $x_i \in X$, $x_i \in C_j \Rightarrow x_i \notin C_k$ where $k \neq j$, $C_k C_j \in P$.

The first condition in the definition assures that the partition covers all data points in X, whereas the second condition assures that all clusters in the partition are mutually exclusive.

Hard Partition Clustering

- A cluster is a collection of objects “similar” between them and “dissimilar” to the objects belonging to other clusters.



The similarity criterion is the distance

- Two or more objects belong to the same cluster if they are “close’ according to a given distance (geometrical distance)

- A simple Euclidean distance metric is sufficient to group similar data instances successfully.

Similarity Measures

- Nearest (Similar) Neighbor Technique
  - Nearest can be taken to mean the smallest **Euclidean distance**

$$d(x, y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

Features (m) with large dissimilarity emphasis more during squared.

  - City Block Distance/Manhattan metric/taxi cab distance: Absolute difference rather than squares

$$d(x, y) = \sum_{i=1}^{m} |x_i - y_i|$$

De-emphasize large feature and influence by more small ones.

Similarity Measures

– Maximum distance metric (Chebychev)

$$d(x,y) = \max_{i=1}^{m} |x_i - y_i|$$

Only consider most dissimilar pair of feature.

– Minkowski Distance

$$d(x,y) = \left(\sum_{i=1}^{m} |x_i - y_i|^r \right)^{1/r}$$

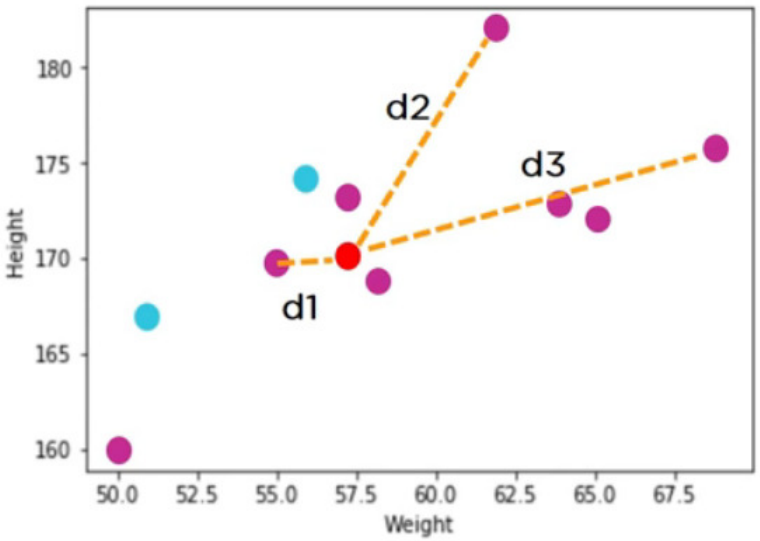
Where r is an adjustable parameter

– Cosine similarity

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

K-Nearest Neighbor (K-NN)

- KNN is a supervised, non-parametric, and lazy learning algorithm.
  - no assumption on underlying data distribution, does not assume any specific form for the relationship between independent and dependent variables.
  - does not use the training data points to do any generalization.



dist(**d1**)= √(170-167)² + (57-51)² ~= 6.7

dist(**d2**)= √(170-182)² + (57-62)² ~= 13

dist(**d3**)= √(170-176)² + (57-69)² ~= 13.4

Similarly, we will calculate Euclidean distance of unknown data point from all the points in the dataset

kNN tends to work best on smaller data-sets that do not have many features.

https://www.youtube.com/watch?v=4HKqjENq9OU

K-Nearest Neighbor (K-NN)

Where (x1, y1) = (57, 170) whose class we have to classify

Weight(x2)	Height(y2)	Class	Euclidean Distance
51	167	Underweight	6.7
62	182	Normal	13
69	176	Normal	13.4
64	173	Normal	7.6
65	172	Normal	8.2
56	174	Underweight	4.1
58	169	Normal	1.4
57	173	Normal	3
55	170	Normal	2

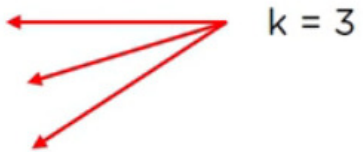
1. Compute a distance value between the item to be classified and every item in the training data-set

https://www.youtube.com/watch?v=4HKqjENq9OU

K-Nearest Neighbor (K-NN)

Now, lets calculate the nearest neighbor at k=3

Weight(x2)	Height(y2)	Class	Euclidean Distance
51	167	Underweight	6.7
62	182	Normal	13
69	176	Normal	13.4
64	173	Normal	7.6
65	172	Normal	8.2
56	174	Underweight	4.1
58	169	Normal	1.4
57	173	Normal	3
55	170	Normal	2



57 kg	170 cm	?

2. Pick the k closest data points (the items with the **k lowest distances**)

3. Conduct a **"majority vote"** among those data points — the dominating classification in that pool is decided as the final classification

https://www.youtube.com/watch?v=4HKqjENq9OU

Forgy's Algorithm

1. Initialize the cluster centroids to the seed points(# of clusters- k random samples)

2. For each sample, find the cluster centroid nearest it. Put the sample identified with this nearest cluster centroid.

3. If no samples changed clusters in step 2, stop.

4. Compute the centroids of the resulting clusters and go to step 2.

Example: Forgy's Algorithm

Samples (4,4), (8,4), (15,8), (24,4), (24,12)

- 1st Iteration

Centroids (4,4) and (8,4)

Complete clustering for all samples then decide centroid for the cluster.

Sample	Nearest Cluster Centroid
(4,4)	(4,4)
(8,4)	(8,4)
(15,8)	(8,4)
(24,4)	(8,4)
(24,12)	(8,4)

- 2nd Iteration

Centroids (4,4) and (17.75,7)

Sample	Nearest Cluster Centroid
(4,4)	(4,4)
(8,4)	(4,4)
(15,8)	(17.75,7)
(24,4)	(17.75,7)
(24,12)	(17.75,7)

Example: Forgy's Algorithm

Samples (4,4), (8,4), (15,8), (24,4), (24,12)

- 3rd Iteration

Centroids (6,4) and (21,8)

Sample	Nearest Cluster Centroid
(4,4)	(6,4)
(8,4)	(6,4)
(15,8)	(21,8)
(24,4)	(21,8)
(24,12)	(21,8)

The K-mean algorithm

1. **Begin with k clusters, each consisting of one of the first k samples.**
   - For each of the remaining n-k samples, find the centroid nearest it.
   - Put the sample in the cluster identified with this nearest centroid.
   - After each sample is assigned, recompute the centroid of the altered cluster.
2. **Go through the data a second time.**
   - For each sample, find the centroid nearest it.
   - Put the sample in the cluster identified with this nearest centroid.
   - Do not recompute any centroid here.

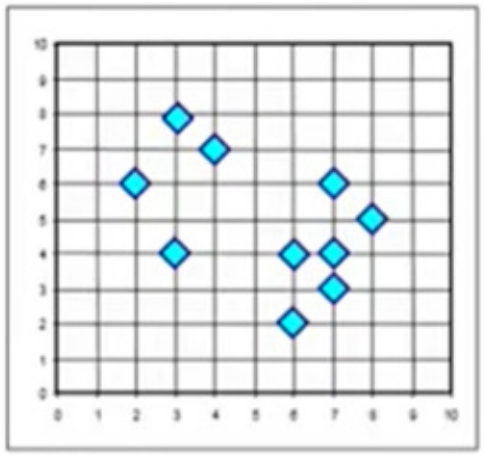
$$J(C_1, C_2, \ldots, C_m, \mu_1, \mu_2, \ldots \mu_k) = \frac{1}{m}\sum_{i=1}^{m} \|X_i - \mu_{C_i}\|^2$$

Sample		Distance to Centroid (9,5.3)	Distance to Centroid (24,8)
(8,4)	(8,4)	1.6	16.5
(24,4)	(24,4)	15.1	4.0
(15,8)	(8,4) -> (11.5,6)	6.6	9.0
(4,4)	(11.5,6)-> (9,5.3)	6.6	40.4
(24,12)	(24,4)-> (24,8)	16.4	4.0

Drawback of K-means

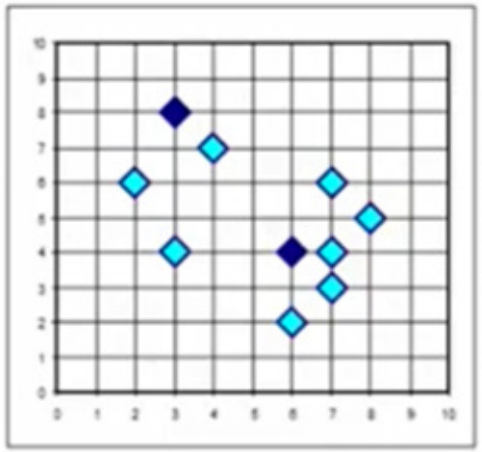
- K-means is sensitive to outliers
  - Such samples are far away from the majority of the data
  - Thus when assigned to a cluster, they can dramatically distort the mean value of the cluster.
- How can we modify the K-means to diminish such sensitivity to outliers?
  - Instead of mean values of objects, we could take actual objects to represent a cluster
  - K-medoid methods use the sum of absolute error

K-medoid Methods

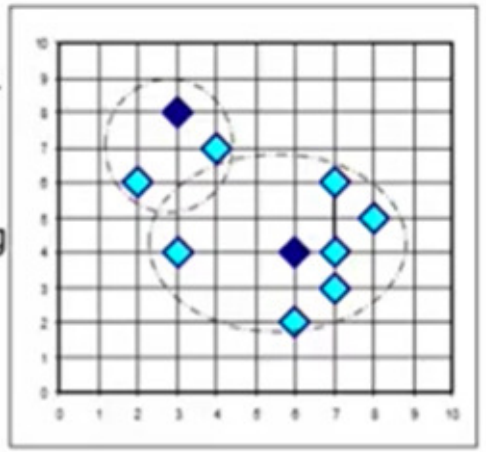


$K = 2$

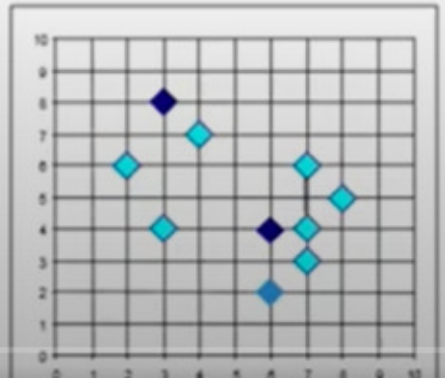
Arbitrary choose K object as initial medoids



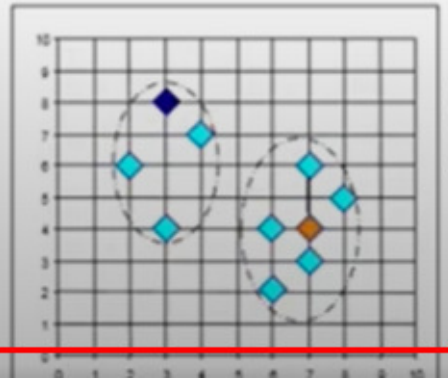
Assign each remaining object to nearest medoids



Randomly select a non-medoid object, O_{ramdom}



Compute total cost of swapping



Swapping O and O_{ramdom}

If quality is improved

$S = Past_{cost} - New_{cost} < 0$

Select initial K *medoids* randomly

Repeat

Object re-assignment

Swap medoid m with o_i if it improves the clustering quality

Until convergence criterion is satisfied

$$\text{COST} = \sum_{i=1}^{k} \sum_{p \in C_i} \text{dist}(p, o_i)$$

Soft Partition Clustering: Fuzzy Clustering

- In many real-world clustering problems, however, some data points partially belong to multiple clusters, rather than to a single cluster exclusively.
  - A Magnetic Resonance Image (MRI) pixel may correspond to a mixture of two different types of tissues.
  - A particular customer may be a "borderline case" between two groups.
  - "fuzzy clustering" algorithm.

Definition 2 (Soft Partition): Let X be a set of data, and x_i be an element of X. A partition $P = \{C_1, C_2, \ldots, C_k\}$ of X is "soft" if and only if the following two conditions hold.

(i) for all $x_i \in X$ and for all $C_j \in P$, $0 \le \mu_{cj}(x_i) \le 1$ such that $x_i \in C_j$.

(ii) for all $x_i \in X$ there exists $C_j \in P$ such that $\mu_{cj}(x_i) > 0$.

where $\mu_{cj}(x_i)$ denote clustering of special interest to which x_i belongs to cluster C_j.

A type of fuzzy clustering of special interest is one that ensures the membership degree of a point x in all clusters adding up to one, i.e.,

$$\sum_j \mu_{cj}(x_i) = 1 \qquad \forall x_i \in X$$

Definition 3 (Fuzzy Pseudo Partition or Fuzzy c-Partition): Let $x = \{x_1, x_2, \ldots, x_n\}$ be a set of given data. A fuzzy pseudo partition or fuzzy c-partition of X is a family of fuzzy subsets of X, denoted by $P\{A_1, A_2, \ldots, A_n\}$ which satisfies

$$\sum_{i=1}^{c} A_1(x_k) = 1$$

Fuzzy membership value

for all $k \in N_n$ and

$$0 < \sum_{k=1}^{n} A_i(x_k) < n$$

for all $i \in N_c$, where c is a positive integer.

Example 1: Let $X = \{x_1, x_2, x_3\}$ and

$$A_1 = 0.6/x_1 + 1/x_2 + 0.1/x_3$$

$$A_2 = 0.4/x_1 + 0/x_2 + 0.9/x_3$$

Then $\{A_1, A_2\}$ is a pseudo partition or fuzzy 2-partition of X. Fuzzy quantizations (or granulations) of variables in fuzzy systems are also examples of fuzzy pseudo partition.

Example: Fuzzy Membership

Fig. 4: Data points and their positions.

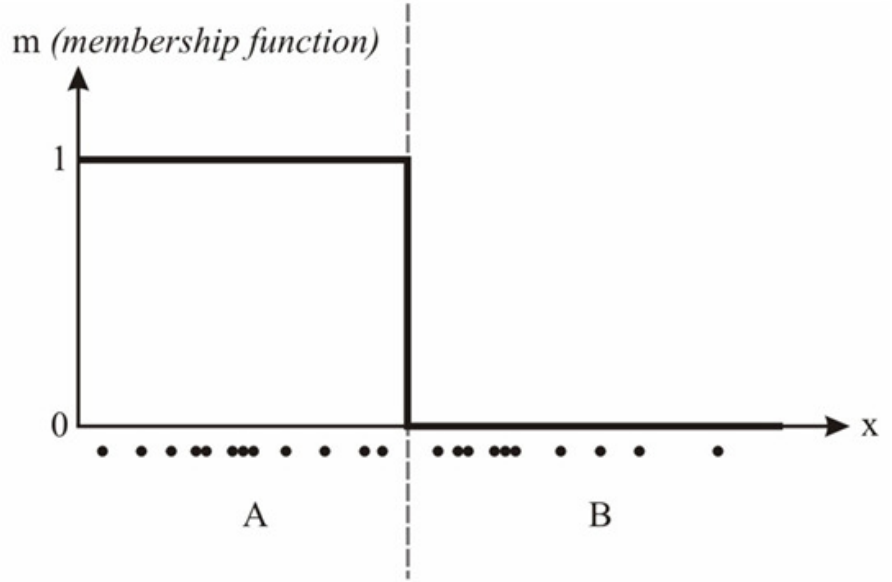


Fig. 5: Membership function for partitional hard clustering of data.

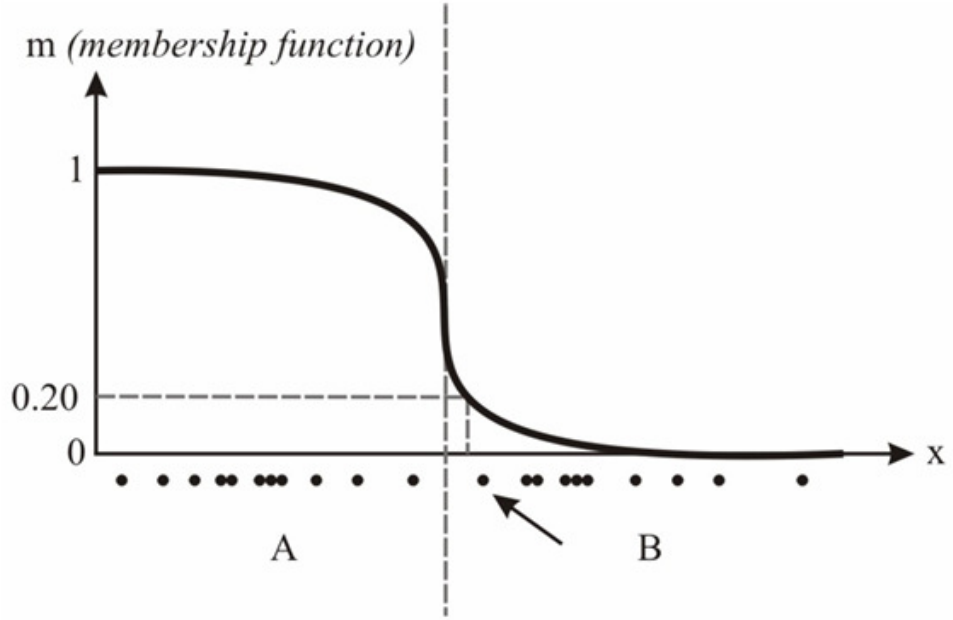
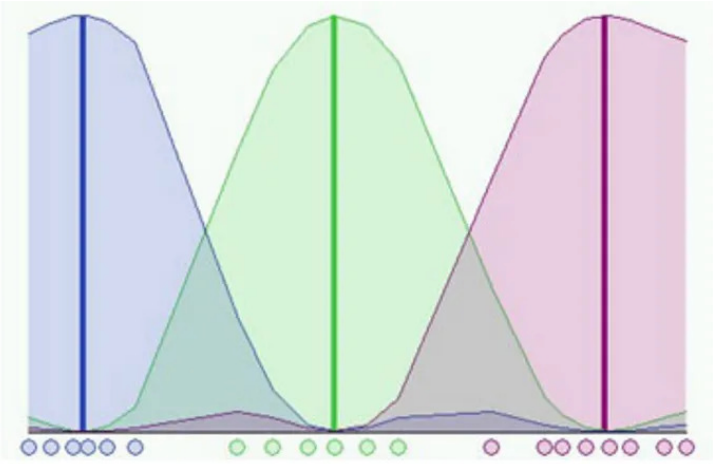


Fig. 6: Fuzzy membership function for cluster 'A'.

A matrix U = [A_i (x_k)]

$$U_{n\times c} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ \cdots & \cdots \\ 0 & 1 \end{bmatrix} \qquad U_{n\times c} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ \cdots & \cdots \\ 0.1 & 0.1 \end{bmatrix}$$

Fig. 7: Data points and their positions.



Fuzzy C-Means Clustering

- The most frequently used fuzzy clustering algorithm is the Fuzzy C-Means (FCM) which is a fuzzification of the k-means algorithm.

- FCM is a method of clustering that allows one piece of data to belong to two or more clusters.

Fuzzy c-means Algorithm

Input: Desired number of clusters c, a real number $m \in (1, \infty)$ and a small positive number ε, serving as a stopping criterion.

Output: A fuzzy pseudo partition and the associated cluster centers.

Fuzzy C-Means Clustering

Steps:

1. Let $t = 0$. Select an initial fuzzy pseudo partition $P^{(0)}$.

2. Calculate the c cluster centers $v_1^{(t)}, v_2^{(t)}, \ldots, v_c^{(t)}$, by using the equation

$$v_i = \frac{\sum_{k=1}^{n} [A_i(x_k)]^m x_k}{\sum_{k=1}^{n} [A_i(x_k)]^m}$$

X_k is the data point

Cluster	{1,3}	{2,5}	{4,8}	{7,9}
1	0.8	0.7	0.2	0.1
2	0.2	0.3	0.8	0.9

For $p^{(t)}$ and the chosen value of m.

m=fuzziness parameters, usually 2

$V11=(0.8^2*1+0.7^2*2+0.2^2*4+0.1^2*7)/(0.8^2+0.7^2+0.2^2+0.1^2)=1.568$

$V12=(0.8^2*3+0.7^2*5+0.2^2*8+0.1^2*9)/(0.8^2+0.7^2+0.2^2+0.1^2)=4.051$

$V21=(0.2^2*1+0.3^2*2+0.8^2*4+0.9^2*7)/(0.2^2+0.3^2+0.8^2+0.9^2)=5.35$

$V22=(0.2^2*3+0.3^2*5+0.8^2*8+0.9^2*9)/(0.2^2+0.3^2+0.8^2+0.9^2)=8.215$

Centroids are {1.568, 4.051} and {5.35, 8.215}

$D11=\text{sqrt}((1-1.568)^2+ (3-4.051)^2)=1.2$, D12=6.79

D21=1.04, D22=4.54

D31=4.63, D32=1.36

D41=7.34, D42=1.82

Cluster	{1,3}	{2,5}	{4,8}	{7,9}
1	0.8	0.7	0.2	0.1
2	0.2	0.3	0.8	0.9
	1	1	2	2

Fuzzy C-Means Clustering

3. Update $p^{(*)}$ by the following procedure: For each $x_k \in X$, if $\left\| x_k - v_i^{(t)} \right\|^2 > 0$ for all $i \in N_c$, then define

$$A_i^{(t+1)}(x_k) = \left[\sum_{j=1}^{c} \left(\frac{\left\| x_k - v_i^{(t)} \right\|^2}{\left\| x_k - v_j^{(t)} \right\|^2} \right)^{\frac{1}{m-1}} \right]^{-1}$$

$A11=[(d11^2/d11^2)^{1/2-1}+(d11^2/d12^2)^{1/2-1}]^{-1}=0.97$
$A12=[(d12^2/d11^2)^{1/2-1}+(d12^2/d12^2)^{1/2-1}]^{-1}=0.03$
$A21=[(d21^2/d21^2)^{1/2-1}+(d21^2/d22^2)^{1/2-1}]^{-1}=0.95$
$A22=[(d22^2/d21^2)^{1/2-1}+(d22^2/d22^2)^{1/2-1}]^{-1}=0.05$

If $\left\| x_k - v_i^{(t)} \right\|^2 = 0$ for some $i \in I \subseteq N_c$, then define $A_i^{(t+1)}(x_k)$ for $i \in I$ by any nonnegative real number satisfying

$$\sum_{i \in I} A_i^{(t+1)}(x_k) = 1$$

And define $A_i^{(t+1)}(x_k) = 0$ for $i \in N_c - I$

$A31=[(d31^2/d31^2)^{1/2-1}+(d31^2/d32^2)^{1/2-1}]^{-1}=0.08$
$A32=[(d32^2/d11^2)^{1/2-1}+(d12^2/d12^2)^{1/2-1}]^{-1}=0.92$
$A41=[(d21^2/d21^2)^{1/2-1}+(d21^2/d22^2)^{1/2-1}]^{-1}=0.06$
$A42=[(d22^2/d21^2)^{1/2-1}+(d22^2/d22^2)^{1/2-1}]^{-1}=0.94$

Cluster	{1,3}	{2,5}	{4,8}	{7,9}
1	0.97	0.95	0.08	0.06
2	0.03	0.05	0.92	0.94

Fuzzy C-Means Clustering

4. Compare $P^{(t)}$ and $P^{(t+1)}$. That is, compute the distance $| P^{(t+1)} - P^{(t)} |$ between $P^{(t)}$ and $P^{(t+1)}$ in the space $R^{n \times c}$. If $| P^{(t+1)} - P^{(t)} | \leq \varepsilon$, then stop; otherwise, increase t by one and return to step 2.

In the fuzzy c-means algorithm, the parameter m is selected according to the problem under consideration. When $m \to 1$, the fuzzy c-means converges to a "generalized" classical c means. When $m \to \infty$, all clusters tend towards the centroid of the data set X i.e., is the partition becomes fuzzier with increasing m. Currently, there is no theoretical basis for an optimal choice for the value of m. However, it is established that the algorithm converges for any $m \in (1, \infty)$.

Problem with Euclidean distance

Euclidean distance can sometimes be misleading (not scale invariant).

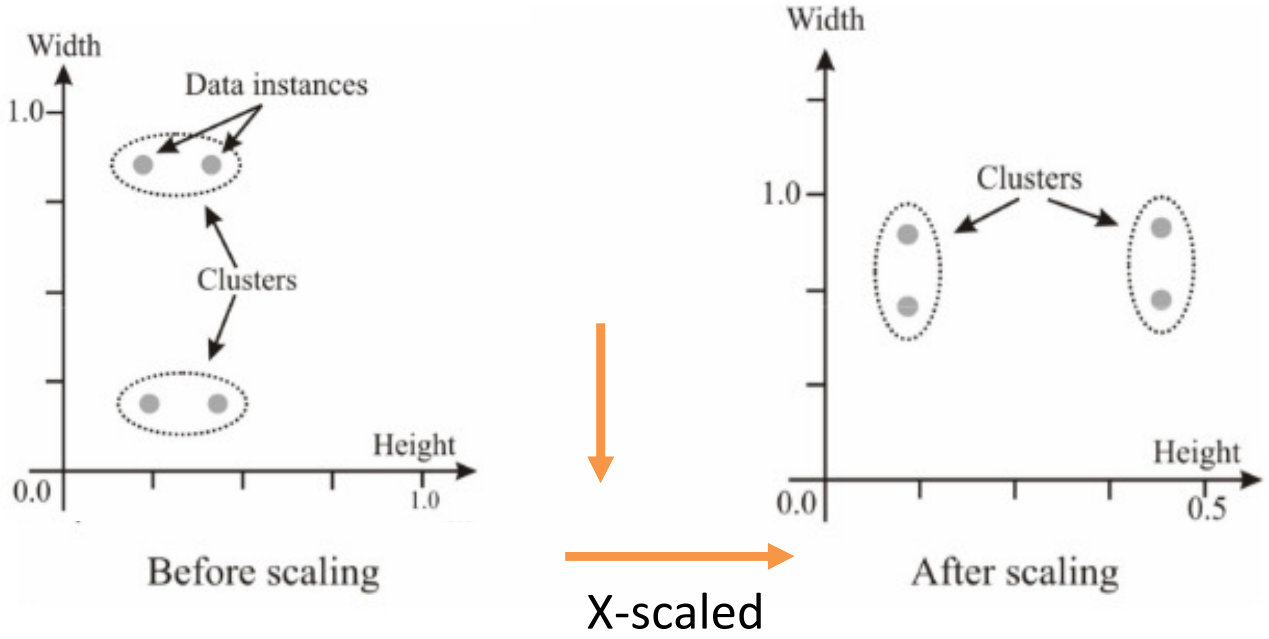


Fig. 2: Clusters before scaling and after scaling.

That means if you work with feet and inches, or pounds and kilograms, you must have them on the same scale.

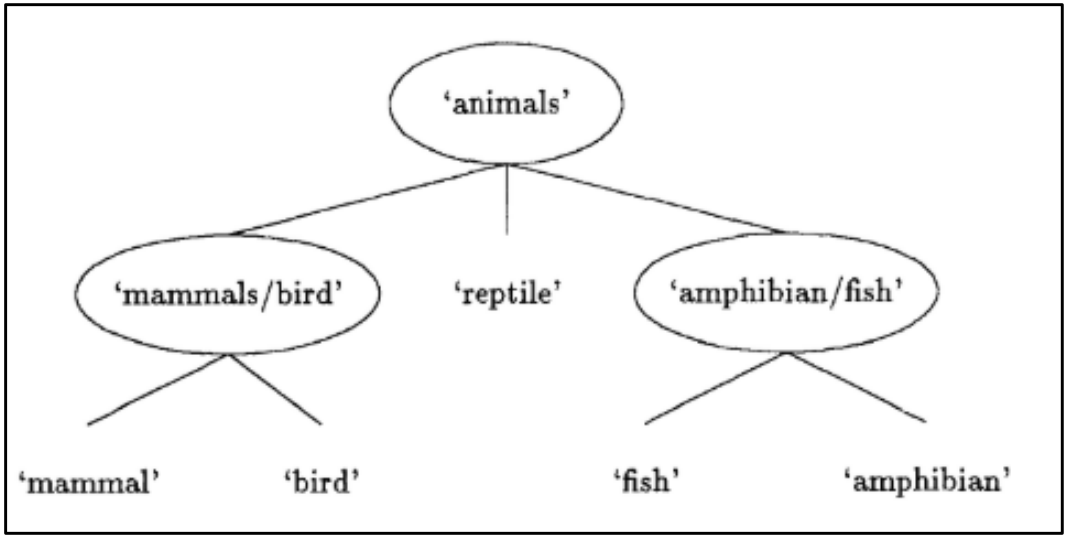
Well, there is a straightforward fix to this and that is to **standardize/normalize** your data.

Domain Knowledge is required. **How about Text Data?**

Conceptual Based Clustering

Table 1. Animal descriptions.

Name	*BodyCover*	*HeartChamber*	*BodyTemp*	*Fertilization*
'mammal'	hair	four	regulated	internal
'bird'	feathers	four	regulated	internal
'reptile'	cornified-skin	imperfect-four	unregulated	internal
'amphibian'	moist-skin	three	unregulated	external
'fish'	scales	two	unregulated	external



Conceptual clustering is a form of clustering in machine learning that given a set of Unlabeled, produces a classification scheme over the objects.

Conceptual clustering goes one step further by finding characteristic descriptions for each group, representing a concept or class.

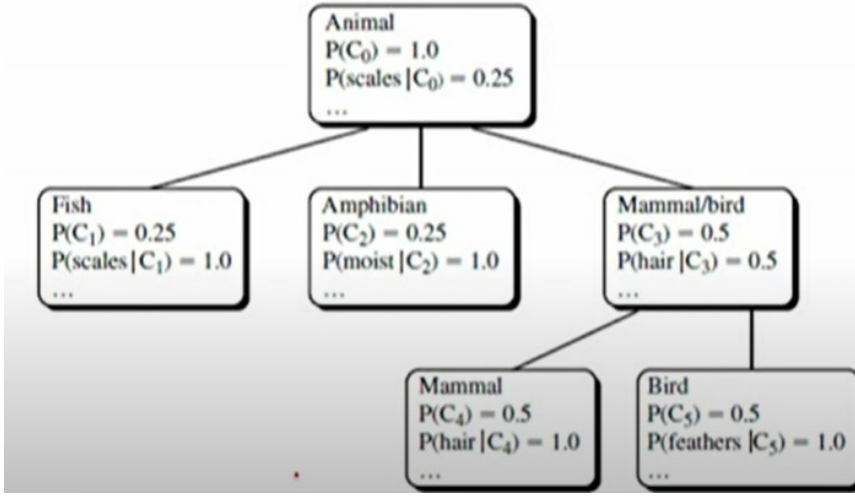
Clustering quality is not solely a function of individual objects. Rather it incorporates factors such as the generality and simplicity of the derived concept descriptions.

Conceptual Based Clustering

- Most methods of conceptual clustering adopt a statistical approach that uses probability measurements in determining the concepts or clusters.
  - COBWEB
  - CLASSIT

Conceptual Based Clustering: COBWEB

- COBWEB is a method of incremental conceptual clustering.
- A categorical attribute value pair describes its input objects.
- It is an overlapping technique. Clusters are not necessarily disjoint and may share components.
- It creates a hierarchical clustering in the form of a classification tree. The hierarchy is incrementally built and regularly rearranged to correct the insertion-order bias.
  - Each node of the tree refers to a concept and contains the probabilistic description.
  - Probability of the concept and conditional probability



Knowledge Acquisition Via Incremental Conceptual Clustering, Machine Learning 2: 139-172, 1987 © 1987 Kluwer Academic Publishers, Boston Manufactured in The Netherlands

Category Utility: Heuristic Measure

- The Goodness held in cluster generally-
  - Similarity of objects within same class => Intra Class
  - Dissimilarity of objects in different classes => Inter Class
- Intra-class similarity is reflected by:

$$P(A_i = V_{ij}|C_k),$$

Where $A_i=V_{ij}$ is an attribute-value pair [Table-1] and C_k is a class

The larger this probability, the greater the proportion of class members sharing the value and the more predictable the value is for class members.

- Inter-class similarity is reflected by:

$$P(C_k|A_i = V_{ij})$$

The larger this probability, the fewer the objects in contrasting classes that share this value and the more predictive the value is of the class.

Category Utility: Heuristic Measure

These probabilities are dispositions of individual values, but they can be combined to give an overall measure of partition quality, where a partition is a set of mutually-exclusive object classes, $\{C_1, C_2, ..., C_n\}$. Specifically,

$$\sum_{k=1}^{n} \sum_{i} \sum_{j} P(A_i = V_{ij}) P(C_k | A_i = V_{ij}) P(A_i = V_{ij} | C_k), \qquad 3-1$$

Classes

Attributes

Values

Weight

The Importance of Individual Value

Inter-Cluster

Intra-Cluster

It helps to increase the class-conditioned predictability & predictiveness of frequently occurring values rather than infrequently occurring values.

More precisely, for any i, j, and k, $P(A_i = V_{ij}) P(C_k | A_i = V_{ij}) = P(C_k) P(A_i = V_{ij} | C_k)$ by Bayes rule, so by substitution function 3–1 equals

$$\sum_{k=1}^{n} P(C_k) \sum_{i} \sum_{j} P(A_i = V_{ij} | C_k)^2. \qquad 3-2$$

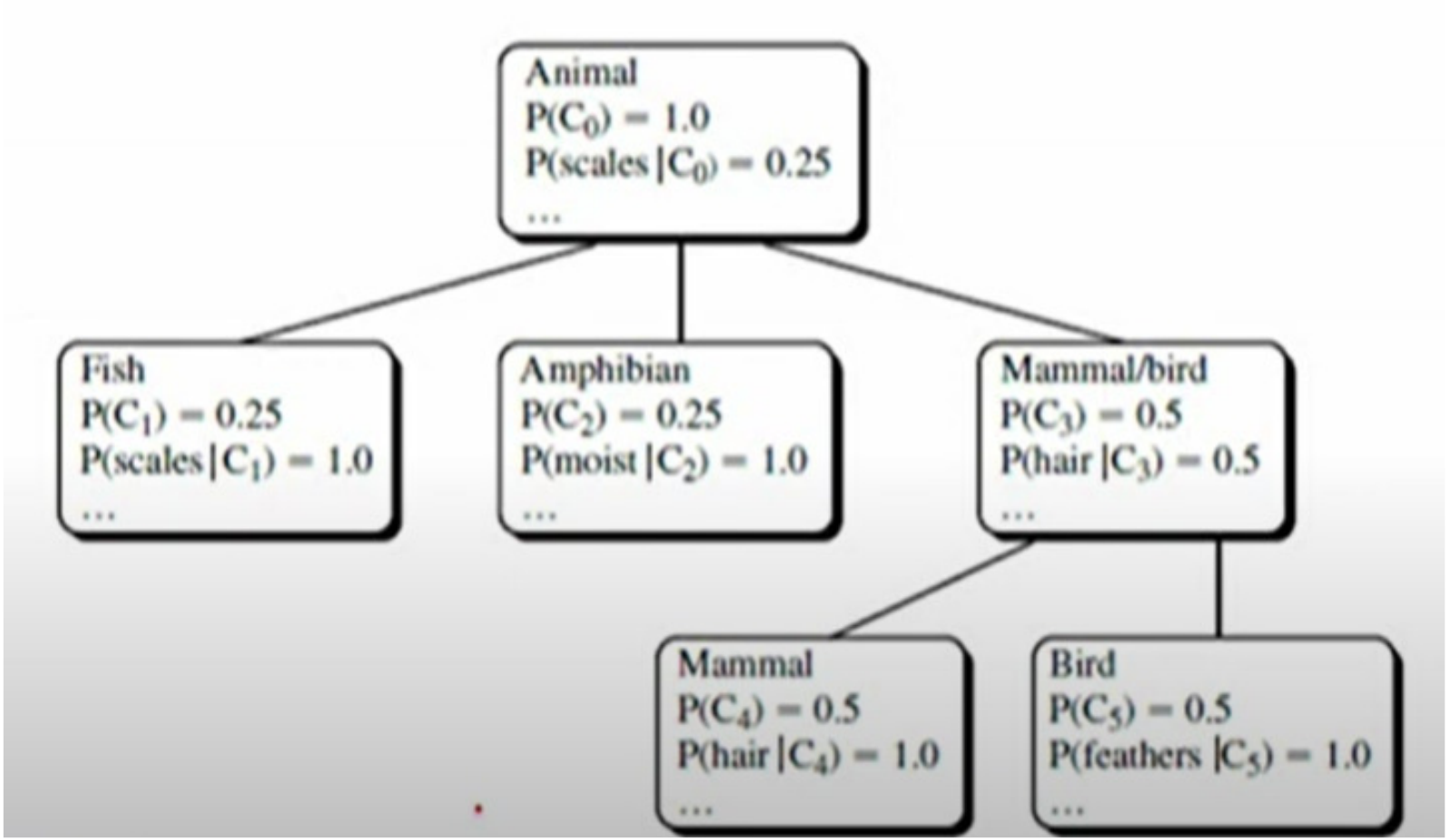
Category Utility: Heuristic Measure

Finally, Gluck and Corter define category utility as the *increase* in the expected number of attribute values that can be correctly guessed ($P(C_k) \sum_i \sum_j P(A_i = V_{ij}|C_k)^2$) given a partition $\{C_1, ..., C_n\}$ over the expected number of correct guesses with no such knowledge ($\sum_i \sum_j P(A_i = V_{ij})^2$). More formally, $CU(\{C_1, C_2, ..., C_n\})$ equals

$$\frac{\sum_{k=1}^{n} P(C_k)[\sum_i \sum_j P(A_i = V_{ij}|C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2]}{n} \qquad 3-3$$

The denominator, n, is the number of categories in a partition.

Knowledge Acquisition Via Incremental Conceptual Clustering, Machine Learning 2: 139-172, 1987 © 1987 Kluwer Academic Publishers, Boston Manufactured in The Netherlands


Animal
$P(C_0) = 1.0$
$P(\text{scales} \mid C_0) = 0.25$
...
Fish
$P(C_1) = 0.25$
$P(\text{scales} \mid C_1) = 1.0$
...
Amphibian
$P(C_2) = 0.25$
$P(\text{moist} \mid C_2) = 1.0$
...
Mammal/bird
$P(C_3) = 0.5$
$P(\text{hair} \mid C_3) = 0.5$
...
Mammal
$P(C_4) = 0.5$
$P(\text{hair} \mid C_4) = 1.0$
...
Bird
$P(C_5) = 0.5$
$P(\text{feathers} \mid C_5) = 1.0$
...

There are four operations that the Cobweb uses while making the tree:

- classifying the object for an existing class,
- creating a new class,
- combining two classes into a single class, and
- dividing a class into several classes.

Knowledge Acquisition Via Incremental Conceptual Clustering, Machine Learning 2: 139-172, 1987 © 1987 Kluwer Academic Publishers, Boston Manufactured in The Netherlands

Operator I: Placing an object in an existing class

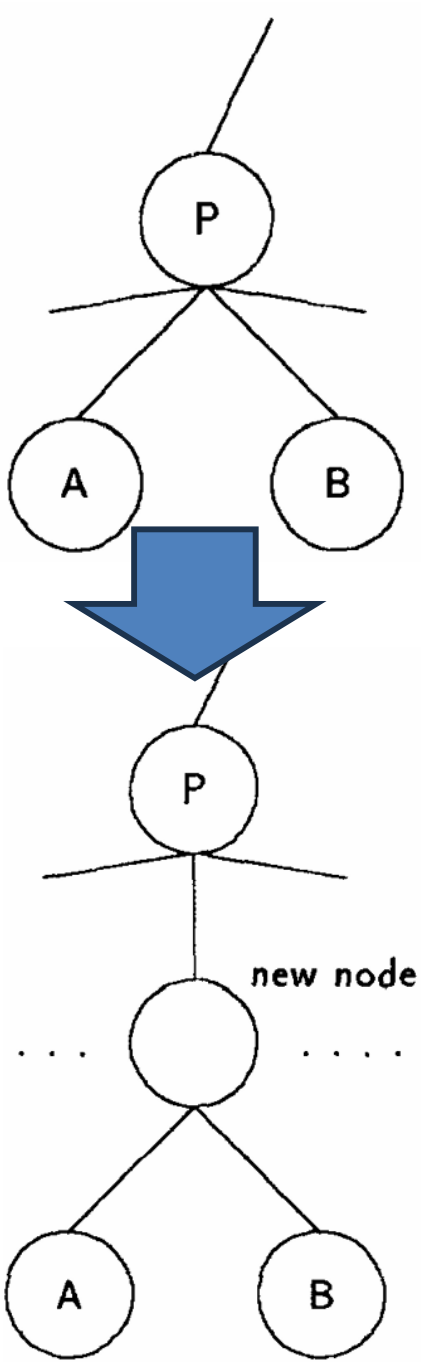
- To determine which category 'best' hosts a new object, COBWEB tentatively places the object in each category.
- The partition that results from adding the object to a given node is evaluated using category utility (3 3).

Operator 2: Creating a new class

- In addition to placing objects in existing classes, there is a way to create new classes.
- Specifically, the quality of the partition resulting from placing the object in the best existing host is compared to the partition resulting from creating a new singleton class containing the object.
- Depending on which partition is best concerning **category utility**, the object is placed in the best existing class, or a new class is created.

Operator 3 : Merging

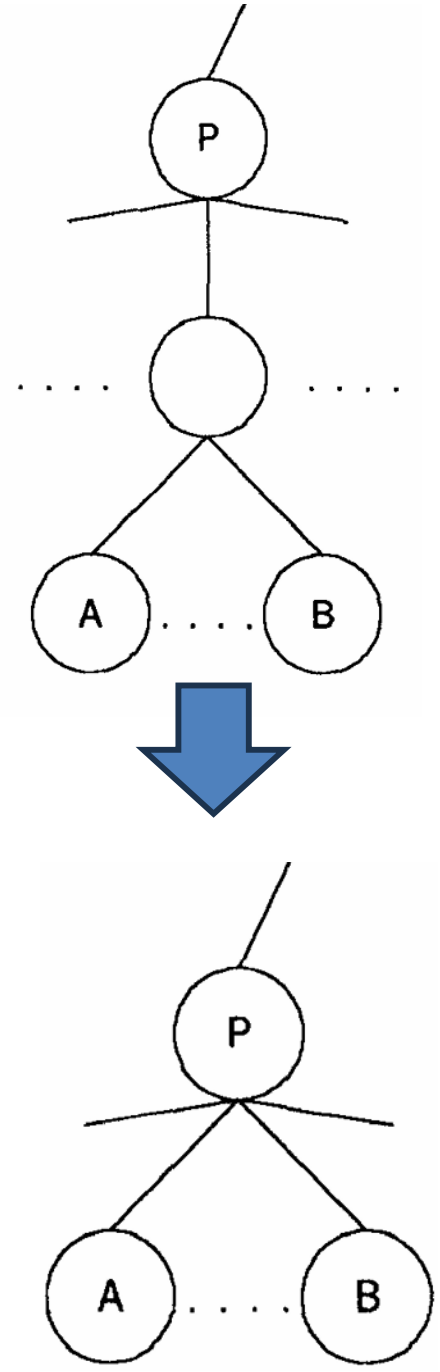
- To guard against the effects of initially skewed data, COBWEB includes operators for node merging and splitting.
- Merging takes two nodes of a level (of n nodes) and 'combines' them in hopes that the resultant partition (of n - 1 nodes) is of better quality.
- Merging two nodes involves creating a new node and summing the attribute-value counts of the nodes being merged. Tile two original nodes are made children of the newly created node as shown in the Figure.



Knowledge Acquisition Via Incremental Conceptual Clustering, Machine Learning 2: 139-172, 1987 © 1987 Kluwer Academic Publishers, Boston Manufactured in The Netherlands

Operator 4 : Splitting

- Splitting may increase partition quality.
- A node of a partition (of n nodes) may be deleted and its children promoted, resulting in a partition of n + m - 1 nodes, where the deleted node had m children as shown in the Figure.



The Algorithm Steps

FUNCTION COBWEB (Object, Root ⟨ of a classification tree ⟩)
1) Update counts of the Root
2) IF Root is a leaf
   THEN Return the expanded leaf to accommodate the new object
   ELSE Find that child of Root that best hosts Object and
        perform **one** of the following
        a) Consider creating a new class and do so if appropriate
        b) Consider node merging and do so if appropriate and
           call COBWEB (Object, Merged node)
        c) Consider node splitting and do so if appropriate and
           call COBWEB (Object, Root)
        d) IF none of the above (a, b, or c) were performed
           THEN call COBWEB (Object, Best child of Root)

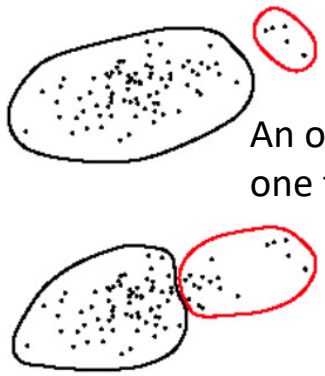
Criterion Functions For Clustering

- How should one evaluate a partitioning of a set of samples into clusters-optimal partition?

- Sum-of-Squared-Error Criterion
  - Let n_i be the # of samples in D_i and let m_i be the mean of those samples

$$m_i = \frac{1}{n_i} \sum_{x \in D_i} x$$

$$J_e = \sum_{i=1}^{c} \sum_{x \in D_i} \|x - m_i\|$$



An optimum partition is defined as one that minimizes Je

FIGURE 10.10. When two natural groupings have very different numbers of points, the clusters minimizing a sum-squared-error criterion J_e of Eq. 54 may not reveal the true underlying structure. Here the criterion is smaller for the two clusters at the bottom than for the more natural clustering at the top. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Clustering of this type are often called **Minimum Variance partitions**

Criterion Functions For Clustering

- Related Minimum Variance Criteria

$$J_e = \frac{1}{2}\sum_{i=1}^{c} n_i \bar{s}_i$$

We can eliminate the mean vectors from the SSE and obtain the left expression.

$$\bar{s}_i = \frac{1}{n_i^2}\sum_{x \in D_i}\sum_{x' \in D_i} (\|x - x'\|^2)$$

Average squared distance between points in the ith cluster and it emphasizes the fact that the sum-of-squared-error criterion uses Euclidean distance as the measure of similarity.

$$\bar{s}_i = \frac{1}{n_i^2}\sum_{x \in D_i}\sum_{x' \in D_i} S(x - x')$$

More generic representation.

$$\bar{S}_i = \min_{x, x' \in D_i} S(x, x')$$

Optimal Clustering Methods

- Clustering validity indexes are usually defined by combining the compactness and separability of the clusters.

Measures the closeness of cluster elements.
A common measure of compactness is variance.

Separability indicates how distinct two clusters are.

- There are two types of validity techniques used for clustering evaluation- external criteria and internal criteria.
- When a clustering result is evaluated based on the data that was clustered itself, this is called internal evaluation
- In external evaluation, clustering results are evaluated based on data not used for clustering, such as known class labels and external benchmarks. Such benchmarks consist of a set of pre-classified items, and (expert) humans often create these sets.

Internal Validation Indexes

- Davies-Bouldin Index: Davies Bouldin (DB) index measures the average similarity between each cluster and its most similar one.
- A lower value of the DB Index indicates that clusters are tight compact and well-separated, reflecting better clustering.
- The goal of this index is to achieve minimum within-cluster variance and maximum between-cluster separations. It measures the similarity of the cluster (Rij) by the variance of a cluster (Si) and the separation of the cluster (dij) by the distance between two clusters (vi and vj). The formulae of the DB index are-

$$S_i = \frac{1}{n_i - 1} \sum_{x \epsilon c_i} d(x, v_i)^2$$

$$d_{ij} = d(v_i, v_j)$$

$$R_{ij} = \frac{S_i + S_j}{d_{ij}}$$

$$R_i = \max_{0 \le j < n_c, i \ne j} (R_{ij}) \ , i = 1 \cdots \cdots n_c R_i \ge 0$$

$$DB = \frac{1}{n_c} \sum_{i=1}^{n_c} R_i$$

TABLE. I. LIST OF SYMBOLS AND THEIR DESCRIPTION

SL No	Symbol/Notation	Description
1.	n_c	Number of total cluster
2.	C_i	i^{th}cluster
3.	d(x,y)	Manhattan distance between two data element
4.	n_i	Number of element in the i^{th} cluster
5.	v_i	Value of the center of the i^{th}cluster
6.	$d(v_i,v_j)$	Distance between two center
7.	S_i	Variance of i^{th}cluster
8.	C_{kmax}	Maximum number of cluster
9.	d	No of dimension

Internal Validation Indexes

- Dunn Index: The value of the Dunn index (DI) is expected to be large if clusters of the data set are well separated. If the dataset has compact and well-separated clusters, the distance between the clusters is expected to be large and the diameter of the clusters is expected to be smaller.
- The clusters are compact and well separated by maximizing the inter-cluster distance while minimizing the intra-cluster distance. The large value of the Dunn index indicates compact and well-separated clusters. The formulae of the Dunn index are-

$$D = \frac{\min_{0 \le i < n_c, 0 \le j < n_c, i \ne j}(d(C_i, C_j))}{\max_{0 \le k < n_c}(diam(C_i))}$$

Where,

$$d(C_i, C_j) = \min_{x \in C_i, y \in C_j}(d(x, y))$$

$$diam(C_i) = \max_{x, y \in C_i}(d(x, y))$$

Internal Validation Indexes

- Silhouette Coefficient: Silhouette Coefficient (SC) shows- how well the objects can fit within the cluster.
- It measures the quality of the cluster by ranging between -1 and 1. A value near to one (1) indicates that the point x is affected to the right cluster.
- There are two terms- cohesion and separation. Cohesion is intra clustering distance, and separation is distance between cluster centroids. A(x) is the average dissimilarity between x and all other points of its cluster. B(x) is the minimum dissimilarity x and its nearest cluster. A cluster which has a value near -1, indicates that the point should be affected to another cluster. The formulae of SC are-

$$a(x) = \frac{1}{n_i - 1} \sum_{y \in C_i y \neq x} d(x, y)$$

$$b(x) = \min_{j, j \neq i} [\frac{1}{n_j} \sum_{y \in C_j} d(x, y)]$$

$$SC = \frac{1}{n_c} \cdot \sum_i \{\frac{1}{n_i} \sum_{x \in C_i} \{\frac{b(x) - a(x)}{\max[b(x), a(x)]}\}\}$$

Internal validation measures for K-means clustering

TABLE. III. OPTIMAL NUMBER OF CLUSTER AND ITS VALUE OF SSE OF RAINFALL, TEMPERATURE, WIND, HUMIDITY AND PEAK HOUR USING K-MEAN WITI DB, DUNN AND SC INDICES

Feature / Method	Rainfall		Temperature		Wind		Humidity		Peak hour	
	Optimal k	*SSE*	*Optimal k*	*SSE*	*Optimal k*	*SSE*	*Optimal k*	*SSE*	*Optimal k*	*SSE*
DB	3	9574.97	2	3690.40	2	9301.24	3	23146.38	2	183452.98
Dunn	3	9574.97	3	2106.56	4	4657.67	6	6339.761	5	49541.539
SC	2	25051.32	2	3690.40	2	9301.24	3	23146.38	2	183452.98
Optimal k	3		3		4		6		5	

No of cluster k	Dunn index value
2	0.08
3	0.09
4	0.10
5	0.11
6	0.11
7	0.13
8	0.12
9	0.12

TABLE. VI. SAMPLE ROAD WEIGHT CLUSTERING RESULT

Data	Rainfall	Temperature	Wind	Humidity	Peak hour	Road weight
1	0	1	0	3	1	0
2	0	1	0	4	3	5
3	0	0	0	3	2	2
4	0	0	0	3	3	5
5	0	2	1	4	0	4
6	0	2	2	4	1	6
7	0	2	0	3	1	0
8	0	2	1	4	3	3
9	0	2	0	3	0	4
10	0	2	1	5	0	4

Exploreing K-Means with Internal Validity Indexes for Data Clustering in Traffic Management System, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 3, 2017

External Validation Indexes

Rand Index: The Rand index computes how similar the clusters (returned by the clustering algorithm) are to the benchmark classifications. It can be computed using the following formula:

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

Where: TP=# of True Positive, TN=# of True Negative

FP=# of False Positive, FN=# of False Negative

If the dataset size is N then $TP + TN + FP + FN = \binom{N}{2}$

- One issue with the Rand index is that false positives and false negatives are equally weighted. This may be an undesirable characteristic for some clustering applications.
- The F-measure addresses this concern as does the chance-corrected adjusted Rand index.

External Validation Indexes

- F-measure: The F-measure can be used to balance the contribution of false negatives by weighting recall through a parameter $\beta \geq 0$. Let precision and recall (both external evaluation measures in themselves) be defined as follows:

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN}$$

We can calculate the F-measure by using the following formula:

$$F_\beta = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R}$$

When $\beta=0$, $F0=P$. In other words, the recall has no impact on the F-measure when $\beta=0$, and increasing β allocates an increasing amount of weight to recall in the final F-measure. Also, TN is not considered and can vary from 0 upward without bound.

External Validation Indexes

Jaccard index: The Jaccard index is used to quantify the similarity between two datasets. The Jaccard index takes on a value between 0 and 1.

An index of 1 means that the two datasets are identical, and an index of 0 indicates that the datasets have no common elements. The following formula defines the Jaccard index :

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

This is simply the number of unique elements common to both sets divided by the total number of unique elements in both sets. Note that TN is not taken into account.

External Validation Indexes

Dice index: The Dice symmetric measure doubles the weight on TP while still ignoring TN:

$$DSC = \frac{2TP}{2TP + FP + FN}$$