Orient BlackSwan

Nearest Neighbor Algorithms

CHAPTER 2: MACHINE LEARNING – THEORY & PRACTICE

Universities Press

Proximity Measures

- Used to quantify the degree of similarity or dissimilarity between two or more patterns.
- Types:
  - **Euclidean distance:** This the most popular distance as it is intuitively appealing. It measures the straight-line distance between two points in a multidimensional space.
  - **Cosine similarity:** This measures the cosine of the angle between two vectors and is often used in text analysis to compare documents.
  - **Jaccard similarity:** This measures the intersection over union of two sets and is often used in recommendation systems to compare users' preferences.
  - **Hamming distance:** This measures the number of positions at which two binary strings differ and is often used in error correction codes.

Distance Measures

- Used to find the dissimilarity between pattern representations.
- Key attributes of distance measures:
  - **Positive reflexivity:** d(x,x) = 0
  - **Symmetry:** d(x,y) = d(y,x)
  - **Triangular** inequality: d(x,y) d(x,z) + d(z,y)
- **Minkowski Distance:**

$$d^r(p,q) = \left(\sum_{k=1}^{L}(|p_k - q_k|^r)\right)^{\frac{1}{r}}$$

where r is a parameter that determines the type of metric being used, p and q are L dimensional vectors.

Different Norms based on "r":

1. L_∞ *Norm:* Here, $r = \infty$ and $d(p,q) = maximum_k(|p_k - q_k|)$, $k \in \{1, \cdots, L\}$.
2. L_2 *Norm:* In this case, $r = 2$ and $d(p,q) = (\sum_{k=1}^{L}(|p_k - q_k|^2))^{\frac{1}{2}}$ is the euclidean distance and is the most popular among the family.
3. *L1 Norm:* In this case, $r = 1$ and $d(p,q) = (\sum_{k=1}^{L}(|p_k - q_k|))$ is the city-block distance.
4. *Fractional Norm*: It is possible that r is a fraction. In such a case the resulting distance is called fractional norm. It is not a metric as it violates the triangle inequality.

Weighted Distance Measure

- To assign greater importance to certain attributes, a weight can be applied to their values in the weighted distance metric.

$$d(x,y) = (\sum_{k=1}^{L} w_k \times (x_k - y_k)^r)^{\frac{1}{r}}$$

where w_k represents the weight associated with the kth dimension or feature.

Non-Metric Similarity Functions

- This category includes similarity functions that do not follow the triangular inequality or symmetry.
- They are commonly used for image or string data, and they are resistant to outliers or extremely noisy data.
- Example: *k*-median distance between two vectors.
  - Given $x = (x_1, x_2, \cdots, x_n)$ and $y = (y_1, y_2, \cdots, y_n)$, the formula for the *k*-median distance is $d(x, y) = k\text{-median}\{|x_1\ y_1|, \cdots, |x_n\ yn|\}$, where the *k*-median operator returns the kth value of the ordered difference vector.
- Example: Cosine similarity between two vectors *x* and *y*.

$$S(x, y) = \frac{x^t y}{\|x\| \|y\|}$$

Levenshtein distance

- Also known as edit distance, is a measure of the distance between two strings. It is determined by calculating the minimum number of mutations needed to transform string *s1* into string *s2*, where a mutation can be one of three operations: *changing a letter*, *inserting a letter*, or *deleting a letter*.
- The edit distance can be defined using the following recurrence relation:
  - d(“ ”, “ ”) = 0, (two empty strings match)
  - d(*s*, “ ”) = d(“ ”, *s*) = ||*s*||, (distance from an empty string) and
  - d(*s1* + *ch1*, *s2* + *ch2*) = min [d(*s1*, *s2*) + {if *ch1* = *ch2* then *0* else *1*}, d(*s1* + *ch1*, *s2*) + 1, d(*s1*, *s2* + *ch2*) + 1]

Mutual Neighbourhood Distance (MND)

- In this case, the function used to measure the similarity between two patterns, x and y, is defined as $S(x, y) = f(x, y, \varepsilon)$, where ε denotes the context, i.e., the surrounding points.
- All other data points are labeled in increasing order of some distance measure, starting from the nearest neighbor as 1 and ending with the farthest point as N-1.
- Mutual neighborhood distance (MND) is calculated as $MND(x, y) = NN(x, y) + NN(y, x)$.
  - Note: NN(x,y) – denotes nearest neighbor distance from x to y.

Proximity Between Binary Patterns

- Let p and q be two binary strings. Some of the popular proximity measures on such binary patterns are:
  - **Hamming Distance (HD):** If pi = qi then we say that p and q match on their ith bit, else (pi ≠ qi) p and q mismatch on the ith bit. Hamming distance is the number of mismatching bits out of the l-bit locations.
  - **Simple Matching Coefficient (SMC):**

$$SMC(p,q) = \frac{M_{11} + M_{00}}{M_{00} + M_{01} + M_{10} + M_{11}}$$

M01 is the number of bits where p is 0 and q is 1
M10 is the number of bits where p is 1 and q is 0
M00 is the number of bits where p is 0 and q is 0
M11 is the number of bits where p is 1 and q is 1

  - **Jaccard Coefficient (JC):**

$$JC(p,q) = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}$$

Different classification algorithms based on the distance measures

- Nearest Neighbour Classifier (NNC)
- *k*-Nearest Neighbour Classifier (*k*NNC)
- Weighted *k*-Nearest Neighbour (W*k*NN)
- Radius distance Near Neighbours
- Tree Based Nearest Neighbours
- Branch and Bound Method
- Leader clustering
- KNN Regression

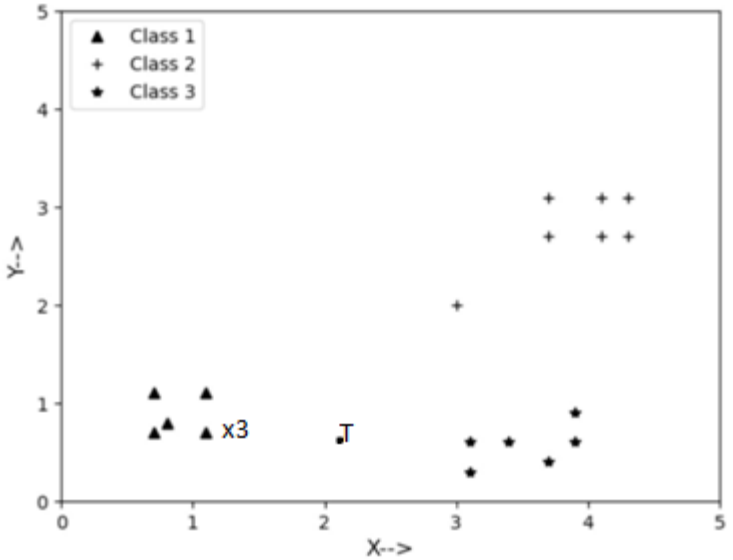
Nearest Neighbour Classifier (NNC)

- Let $\mathcal{X} = \{(x_1, l_1), (x_2, l_2), \cdots, (x_n, l_n)\}$, each pattern be a vector in some L dimensional space and l_i is its class label.
- Now the nearest neighbor of x (i.e., the test pattern) is given by

$$NN(x) = argmin_{x_j \in \mathcal{X}} d(x, x_j)$$

where x_j is the jth training pattern and $d(x, x_j)$ is the distance between x and x_j.

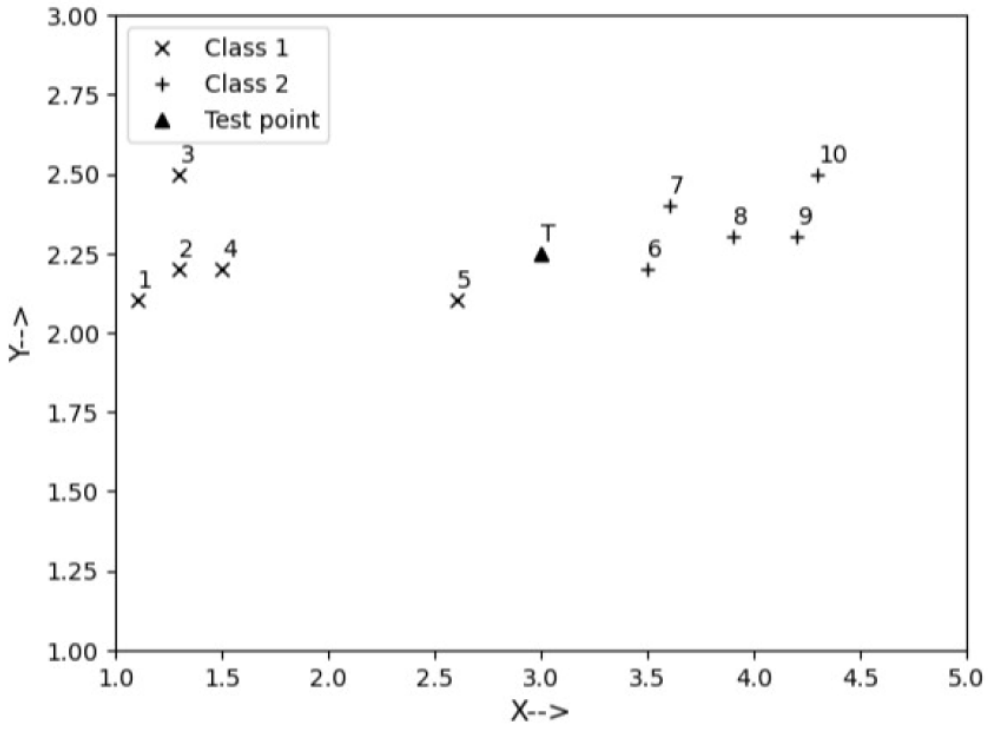
- Test pattern T(2.1, 0.7) is assigned to class 1, since its Euclidean distance from x3 is minimum (i.e., 1 unit)



k-Nearest Neighbour Classifier

- Similar to Nearest Neighbors Classifier (NNC) algorithm, where we find the *k* nearest neighbors of a test pattern *x* from the training data χ , and then assigning the majority class label among the *k* neighbors to *x*.

- By using this method of selecting the majority class label among the *k* nearest neighbors, the error in classification can be reduced, especially when the training patterns are noisy.
- T is assigned to class 2, even though *x5* is closest, but is an outlier.



Weighted *k*-Nearest Neighbour (WkNN)

- Similar to *k*NN algorithm, but it takes into account the distance of each of the k neighbors from the test point by weighting them accordingly.
- Each neighbor is associated with a weight *w*, which is determined by the following formula:

$$w_j = \begin{cases} \frac{(dk-dj)}{(dk-d1)} & if\ (d_k \neq d_1) \\ 0 & if\ (d_k = d_1) \end{cases}$$

Here, *j* represents the neighbor's index in the list of *k* nearest neighbors, while d_k and d_j are the distances between the test point and the *k*-th neighbor and the *j*-th neighbor, respectively.

Weighted *k*-Nearest Neighbour (WkNN)

- For example the distances from T to its 5 nearest data points are

  d(T, x3) = 1.0; d(T, x14) = 1.01; d(T, x13) = 1.08;

  d(T, x5) = 1.08; d(T, x16) = 1.30;

- The weight values will be

$$w_3 = 1.0$$

$$w_{14} = \frac{(1.30-1.01)}{(1.30-1.00)} = 0.97$$

$$w_{13} = \frac{(1.30-1.08)}{(1.30-1.00)} = 0.73$$

$$w_5 = \frac{(1.30-1.08)}{(1.30-1.00)} = 0.73$$

$$w_{16} = 0$$

Summing up for each selected class,
Class 1 to which x3 and x5 belong sums to 1.73, and
Class 3 to which x14, x13 and x16 belong sums to 1.7.
Therefore, the point T belongs to Class 1.

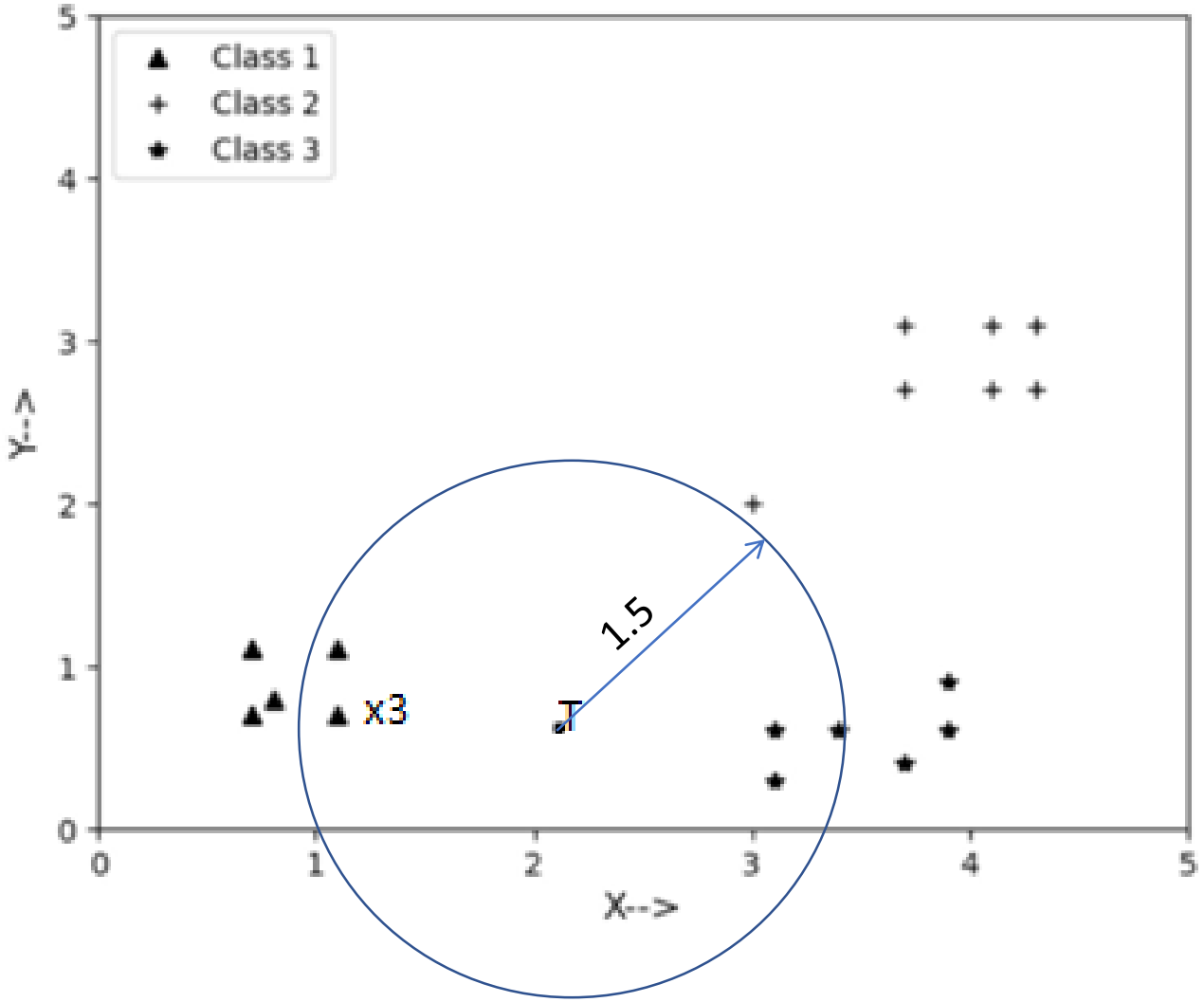
Radius distance Near Neighbours Algorithm

- This algorithm is an alternative to the *k*NN algorithm that considers all the neighbors within a specified distance *r* of the point of interest.
- Steps:
  - Given a point T , identify the subset of data points that fall within the radius r centered at T , denoted by

$$Br(T) = \{x_i \in \mathcal{X} s.t. \|T - Xi\| \leq r\}$$

  - If *Br(T)* is empty, output the majority class of the entire dataset.
  - If *Br(T)* is not empty, output the majority class of the data points within *Br(T)*.

Radius distance Near Neighbours Algorithm



Class assigned to T is Class 3

Tree Based Nearest Neighbours Algorithm

- Based on the transactional database
- Mainly for association rule mining, aims to identify the occurrence of one item based on the occurrence of other items.
- Frequent Pattern (FP) tree:
- Steps:
  - Construct 1-frequent itemset, sort them in descending order of frequency
  - Arrange each transaction in the same order of items as that of frequent 1-itemset.
  - Add the transaction to the branch of the FP-tree such that the for the common prefix part, the node-count of the items in the FP-tree are incremented and the new nodes are added to the tree for the remaining part of the transaction.

Tree Based Nearest Neighbours Algorithm

Digit	Transaction (Positional information of a digit)
0	1, 2, 3, 4, 5, 8, 9, 12, 13, 14, 15, 16
1	4, 8, 12, 16
4	1, 5, 7, 9, 10, 11, 12, 15
6	1, 5, 9, 10, 11, 12, 13, 14, 15, 16
7	1, 2, 3, 4, 8, 12, 16

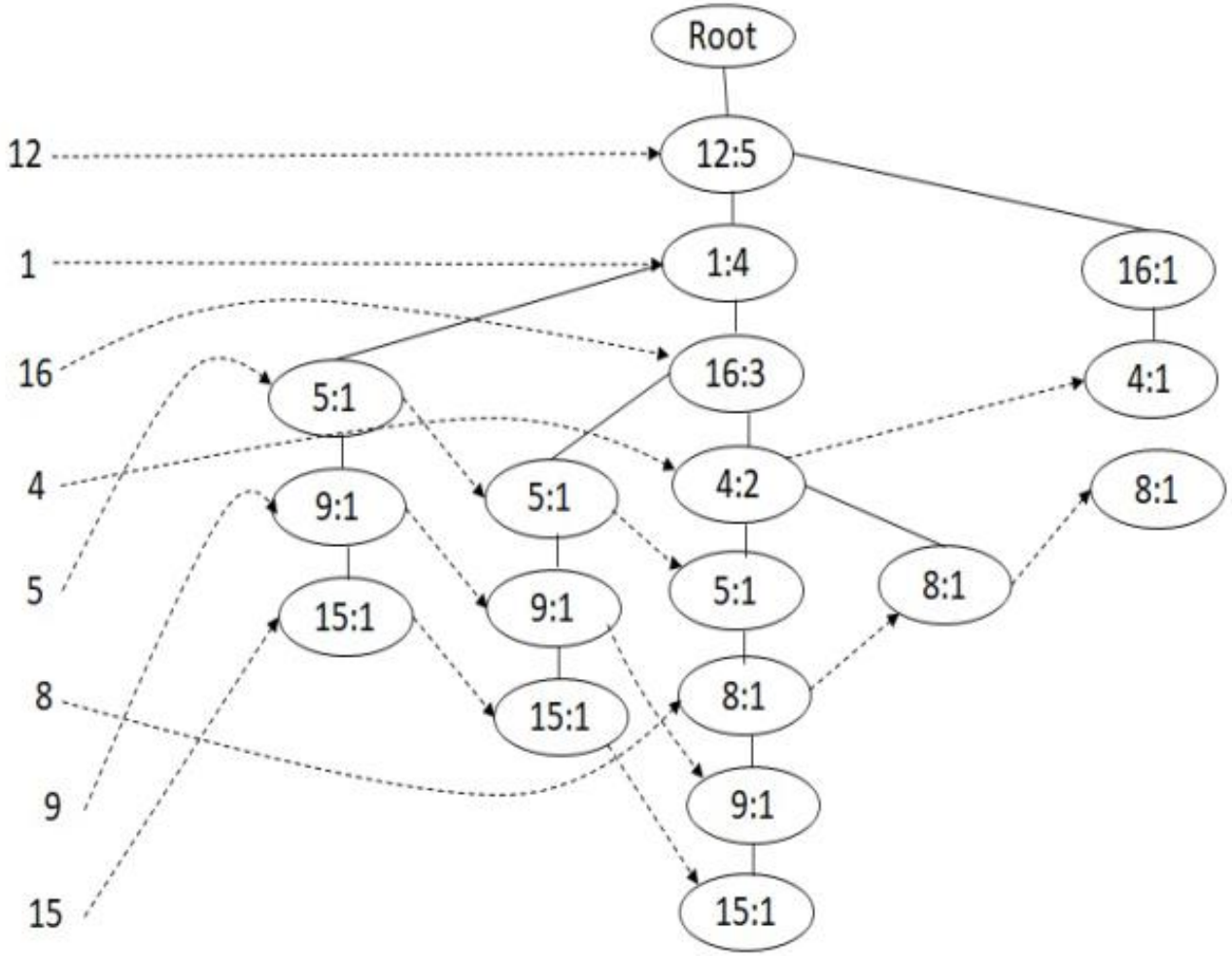
Transaction database

1-frequent itemset :
(12: 5), (1: 4), (16: 4), (4: 3), (5: 3), (8:3), (9: 3), and (15: 3)
(With minimum support = 3)

Digit	Transaction (After removing non-frequent items)
0	12, 1, 16, 4, 5, 8, 9, 15
1	12, 16, 4, 8
4	12, 1, 5, 9, 15
6	12, 1, 6, 5, 9, 15
7	12, 1, 16, 4, 8

Transaction database with transactions ordered according to frequency of items

Tree Based Nearest Neighbours Algorithm



Let the test pattern, T = 1, 2, 3, 4, 6, 7, 8, 12, and 16.

After removing the non-frequent items (Minimum support = 3),

T` = 1, 4, 8, 12, 16 .

By arranging these items in the order they appear in the FP tree, we get 12, 1, 16, 4, 8.

Starting from the root node of the FP tree (12), we can compare the remaining items in the test pattern.

It is observed that the test pattern has the maximum number of items in common with digit 7. Therefore, it can be classified as belonging to digit 7.

Branch and Bound Method

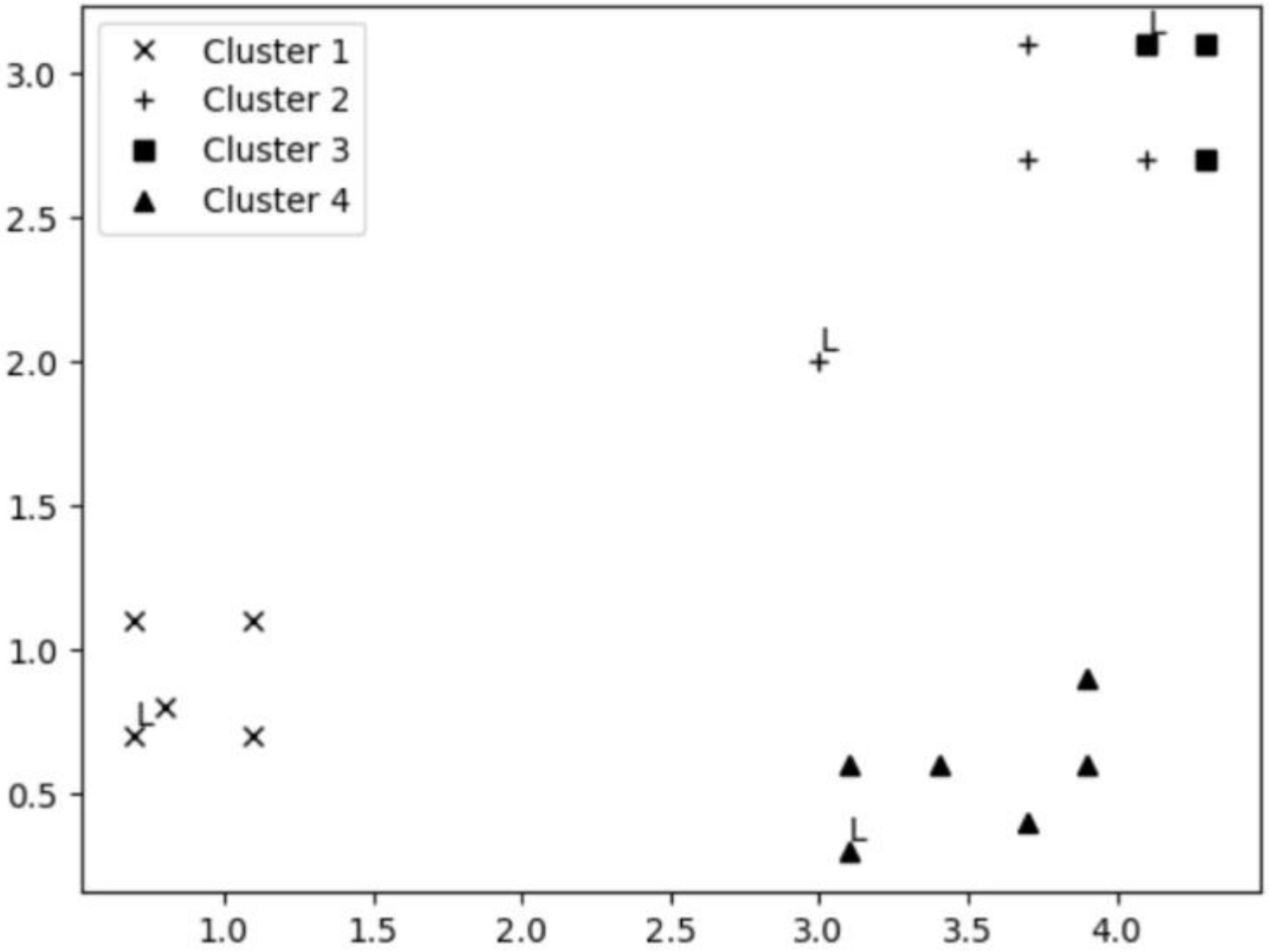
- By clustering the data into representative groups with the smallest possible radius, we can search for the nearest neighbor while avoiding branches that cannot possibly have a closer neighbor than the current best value found.
- Each group is represented by cluster centroid and radius (μ, r).
- To identify the belongingness of the point *T* to the group, the lower bond, bj with reference to the cluster *j*, and recursively branching to the cluster with the smallest b_j until the nearest neighbor is found or the bound is not satisfied. Note that b_j for a cluster *j* is obtained by

$$b_j = d(T, \mu_j) - r_j$$

Leader clustering

- It is an incremental clustering approach that is commonly used to cluster large data sets that cannot be accommodated in the main memory of the machine processing the data.
- It scan the dataset only once.
- **Idea:** A data point is assigned to an existing nearest cluster if the point falls within a threshold distance from the representative (leader) of the cluster; if there is no cluster in the threshold distance of the point, then a new cluster is initiated with the data point becoming the leader of the new cluster.
- It is order dependent algorithm. i.e, the order in which the data is presented to the algorithm can affect the resulting clusters.

Leader clustering



- Data be clustered are processed in the order x1, x2, · · · , x18.
- Threshold T be set to 1.5.
- Initial cluster centre (leader) is taken as x1.
- Figure shows the 4 clusters formed with the leaders (L)

KNN Regression

- Let $\mathcal{X} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$. The regression model needs to use χ to find the value of *y* for a new vector *x*.
- Idea:
  - Find *k* nearest neighbors of *x* from the n data vectors. Let them be $x_1, x_2, \cdots, x_k$.
  - Consider the *y* values associated with these x_i's. Let them be $y^1, y^2, \cdots, y^k$.
  - Predicted value of y, i.e., $\hat{y} = \frac{1}{k}(y^1 + y^2 + \cdots + y^k)$

Concentration Effect and Fractional Norms

- A major difficulty encountered while using some of the popular distance measures like the Euclidean distance is that the distance values, between various pairs of points, may not show much dynamic range.

Example *Let $p = (4, 2)$ and $q = (2, 4)$ be two points in a two-dimensional space. Values of the distance using some popular distance norms are:*

1. L_∞ **Norm or the Max Norm**: $Max(|4-2|, |2-4|) = 2$.
2. L_2 **Norm or Euclidean Distance**: $2\sqrt{2}$
3. L_1 **Norm or City-Block Distance**: $|4-2| + |2-4| = 4$.

- Observe that as the *r* value in the Minkowski norm keeps decreasing the distance between the pair *(p,q)* keeps increasing.

Concentration Effect and Fractional Norms

- This behaviour prompted researchers to go for fractional norms r is a fraction) to increase the dynamic range of the values or decrease the concentration effect.

Example *We have* $p = (4, 2)$ *and* $q = (2, 4)$. *The fractional norms give the distances as shown next.*

1. $L_{0.5}$ **Norm:** $(\sqrt{|4-2|} + \sqrt{|2-4|})^{\frac{1}{0.5}} = (2\sqrt{2})^2 = 8.$
2. $L_{0.25}$ **Norm:** $(2^{0.25} + 2^{0.25})^4 = 32.$
3. $L_{0.1}$ **Norm** : $(2^{0.1} + 2^{0.1})^{10} = 2048.$

- An important observation is that in the process of improving the dynamic range of distance values the fractional norms can improve the classification performance.

Concentration Effect and Fractional Norms

- Wisconsin breast-cancer data as with different norms shows the increase in accuracy