The K-Medoids Clustering Algorithm
From "means" to "medoids"

- **Introduction**
  - K-Medoids (also called as PAM: Partitioning Around Medoid) algorithm was proposed in 1987 by Kaufman and Rousseeuw
  - K-medoids clustering is a variant of K-means
  - It is more robust to noises and outliers: A medoid is less influenced by outliers
  - Instead of using the mean point as the center of a cluster, K-medoids uses an actual point (Medoid) in the cluster to represent it.
  - A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum.
  - The dissimilarity of the medoid(Ci) and object(Pi) is calculated by using the Manhattan distance:

$$E = |Pi - Ci|$$

- **PAM Algorithm:**
  - The medoid of a set is the object with the least distance to all others.
    - The most central, most representative object
  - k-medoids objective function: total deviation criterion (absolute errors)

$$TD = \sum_{i=1}^{k} \sum_{x_j \in C_i} dist(x_j, m_i)$$

where mi is the medoid of cluster Ci.

  - As with k-means, the k-medoid problem is NP-hard
  - **Algorithm:**
    1. Given k
    2. Randomly pick k instances as initial medoids
    3. Assign each instance to the nearest medoid x

4. Calculate the objective function
    - The sum of dissimilarities of all instances to their nearest medoids
5. For each medoid m, for each data point O which is not a medoid (or choose randomly a data point that is not a medoid):
    i. Swap m and O, associate each data point to the closest medoid, recompute the cost
    ii. If the total cost is more than that of the previous step, undo the swap

- Time Analysis:

  $O(k(n-k)^2)$ for each iteration

  where n is # of data and k is # of clusters

- **Example:**
    - Given the following dataset:

Item #	X	Y	Dissimilarity from C1(6,7)	Dissimilarity from C2(7,9)	Cluster
0	5	6	|5-6|+|6-7|=2	|5-7|+|6-9|=5	C1
1	4	5	|4-6|+|5-7|=4	|4-7|+|5-9|=7	C1
2	4	7	|4-6|+|7-7|=2	|4-7|+|7-9|=5	C1
3	6	7	-	-	

4	7	8	|7-6|+|8-7|=2	|7-7|+|8-9|=1	C2
5	7	9	-	-	
6	8	4	|8-6|+|4-7|=5	|8-7|+|4-9|=6	C1
7	8	9	|8-6|+|9-7|=4	|8-7|+|9-9|=1	C2
8	4	9	|4-6|+|9-7|=4	|4-7|+|9-9|=3	C2

1. Randomly select two medoids: C1= (6.7) and C2=(7,9)
2. Calculate Cost:

   Manhattan Distance:

   The Manhattan distance of two points (x1,y1) and (x2,y2) is:

   Mdist ((x1,y1), (x2,y2)) = |x1-x2|+|y1-y2|

3. Calculate the total cost of the current cluster:

   C1 = {(5,6), (4,5), (4,7), **(6,7)**, (8,4)}

   Note that (6,7) is the medoid of C1

   C2 = {(7,8), **(7,9)**, (8,9), (4,9)}

   Note that (7,9) is the medoid of C2

Total Cost = $Cost(c,x) = \sum_i |c_i - x_i|$

Total Cost = Cost((6,7),(5,6)) + Cost((6,7),(4,5)) +

Cost((6,7),(4,7)) + Cost((6,7),(8,4)) +
Cost((7,9),(7,8)) + Cost((7,9),(8,9)) +
Cost((7,9),(4,9))

= 2+4+2+5+1+1+3=18

4. Choose randomly another data point O different from C1 and C2 and randomly replace it with either C1 or C2

Let assume we picked O = (5,6) and replace C1. Now the two medoids are O=(5,6) and C2=(7,9)

Item #	X	Y	Dissimilarity from O(6,7)	Dissimilarity from C2(7,9)	Cluster
0	5	6	-	-	O
1	4	5	|4-5|+|5-6|=2	|4-7|+|5-9|=7	O
2	4	7	|4-5|+|7-6|=2	|4-7|+|7-9|=5	O
3	6	7	|6-5|+|7-6|=2	|5-7|+|7-9|=4	O
4	7	8	|7-5|+|8-6|=4	|7-7|+|8-9|=1	C2
5	7	9	-	-	C2
6	8	4	|8-5|+|4-6|=5	|8-7|+|4-9|=6	O
7	8	9	|8-5|+|9-6|=6	|8-7|+|9-9|=1	C2
8	4	9	|4-5|+|9-6|=6	|4-7|+|9-9|=3	C2

5. Calculate the total cost of the current cluster:

Total Cost = $Cost(c, x) = \sum_i |c_i - x_i|$

Total Cost = Cost((5,6),(4,5)) + Cost((5,6),(4,7)) + Cost((5,6),(6,7)) + Cost((5,6),(8,4)) + Cost((7,9),(7,8)) + Cost((7,9),(8,9)) + Cost((7,9),(4,9))

= 2+2+2+5+1+1+3 = 17

6. Cost of swapping of medoid C1 with O is:

S = current total cost – Previous Total cost = 17-18 = -1 < 0

Swapping C1 with O gives us a better clustering. So, the medoids are O and C2 instead of C1 and C2

- **Advantages:**
  - It is simple to understand and easy to implement
  - K-medoid algorithm is fast and converges in a fixed number of steps
  - K-medoid is less sensitive to outliers than another partitioning algorithm

- **Disadvantages:**
  - K-medoid is not suitable for clustering non-spherical (arbitrary shaped) groups of objects
  - It may give different results for different runs on the same dataset because the first k medoids are chosen randomly.
  - PAM works efficiently for small data sets but does not scale well for large data sets.