Partitioning Method

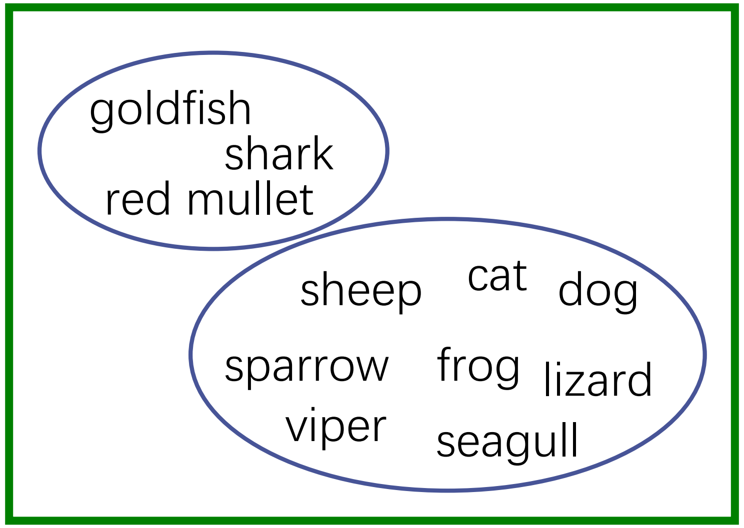
CS385 – Machine Learning - Clustering

Clustering Conception

- Cluster
  - Collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters
- Clustering Analysis
  - Birds of a feather flock together

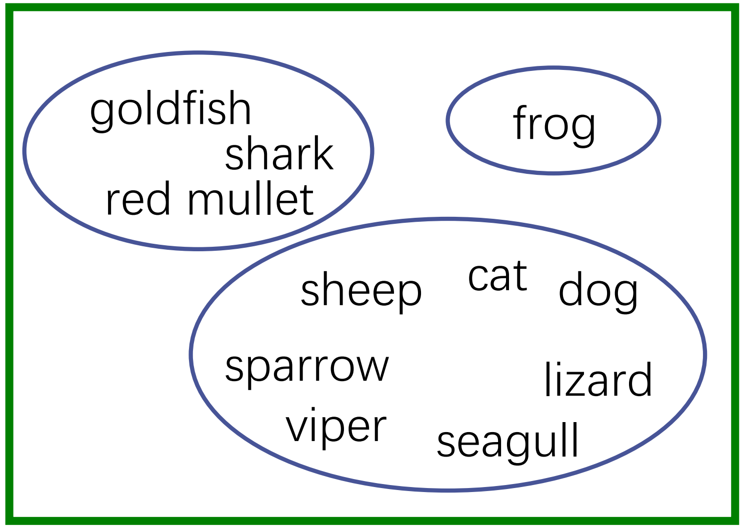
Byname → Unsupervised learning
Learning without a teacher
Numerical taxonomy
Typology
Partition

Clustering Criterion



The existence of lungs

The environment to live



Clustering Similarity

- Numerical
  - Euclidean distance
  - Manhattan distance
  - Minkowski distance
  - …
- Binary, Nominal, Ordinal etc.
  - Jaccard coefficient
    - sim(p_i, p_j) = $| p_i \cap p_j | \,/\, | p_i \cup p_j |$
- Mixed

{fruit, veg, milk, crab}

{milk, fruit, icecream}

$\frac{2}{5}$

Typical Application

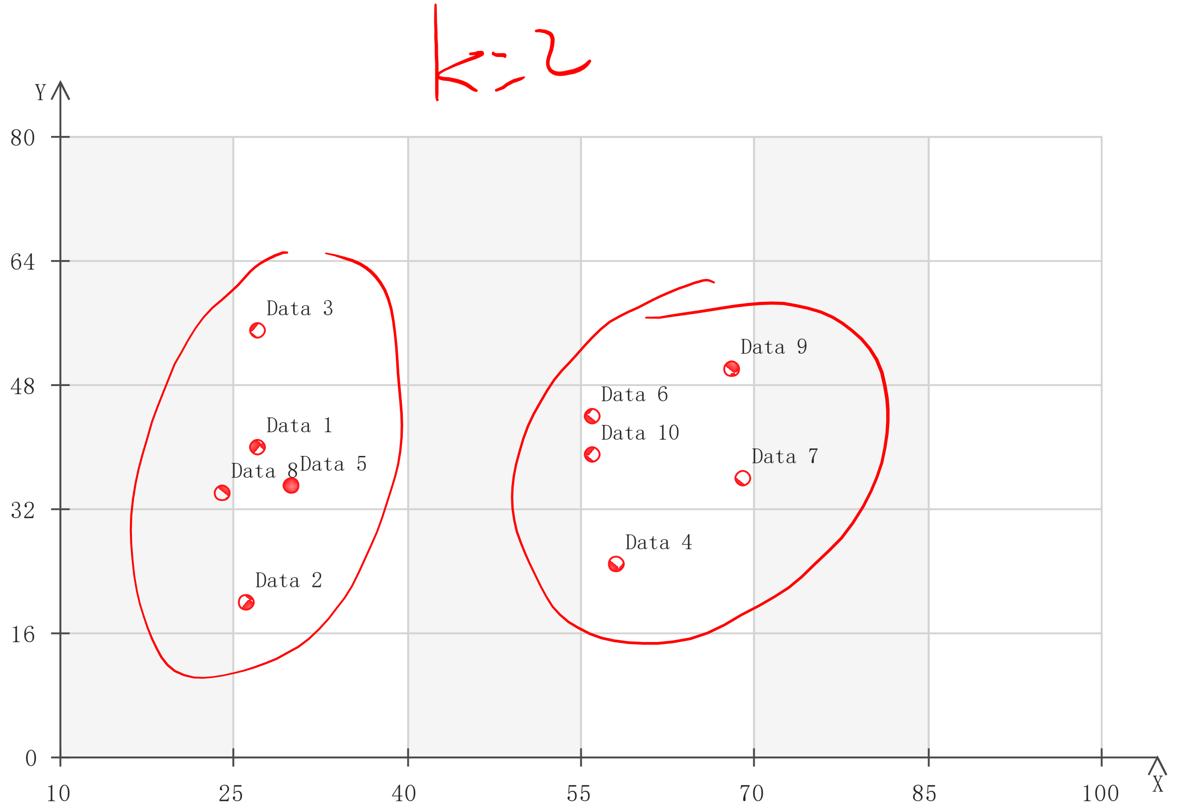
- Business: CRM
- Biology: Gene
- Identification of groups of ...

- Image processing
- Gain distribution of data
- Web for information discovery
- Preprocessing step

Clustering – input and result

- To find structure from the training data set

$$\begin{bmatrix} x_{11}x_{12}\ldots x_{1n} \\ x_{21}x_{22}\ldots x_{2n} \\ \ldots \\ x_{m1}x_{m2}\ldots x_{mn} \end{bmatrix}$$



Criterion

- Given
  - *n* objects
  - *k* represents number of clusters
  - *Criterion function*
- Gain
  - *n* objects are organized into *k* cluster
  - the formed clusters optimize the *criterion function*

$$E = \frac{Total\ Distance(intraCluster)}{Total\ Distance(interCluster)}$$

better

Clustering – 1-d example

Collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters

D = {o1, o2, o3, o4, o5}={3, 1, 9, 10, 2 }, K=2

Clustering1: {3,1,9}, {10,2}

$$E1 = \frac{[\overset{2}{d(3,1)} + \overset{6}{d(3,9)} + \overset{8}{d(1,9)}] + [\overset{8}{d(10,2)}]}{\underset{7}{d(3,10)} + \underset{1}{d(3,2)} + \underset{9}{d(1,10)} + \underset{1}{d(1,2)} + \underset{1}{d(9,10)} + \underset{7}{d(9,2)}}$$

Clustering2: {3,1,2}, {9,10}

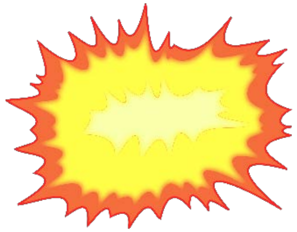
$$E2 = \frac{[d(3,1) + d(3,2) + d(1,2)] + [d(9,10)]}{d(3,10) + d(3,9) + d(1,10) + d(1,9) + d(2,10) + d(2,9)}$$

...

ClusteringN: ⋯

$$EN = \cdots$$

$$E = \frac{\sum_{m=1}^{K} \sum_{Oi,Oj \in C_m} d(O_i, O_j)}{\sum_{m=1}^{K} \sum_{n=1}^{K} \sum_{O_i \in C_m, O_j \in C_n} d(O_i, O_j)}$$



When the size of D grows → combination explosion

K-medoid

- medoid: an actual object, representative object centrally located in a cluster

$$\boxed{E = \sum_{i=1}^{K} \sum_{O \in C_i} d(o, medoid_i)}$$

better

- Groups n objects into k clusters by minimizing the E
- Find k medoids that minimize E
  - Brute-force algorithm – exhaustive search

K-medoid – exhaustive search

D = {o1, o2, o3, o4, o5}={3, 1, 9, 10, 2 }, K=2

Iteration	Medoids	Clustering	E
1	**3**, **1**	C1={**3**,9,10} C2={**1**,2}	13+1=14
2	**3**, **9**	C1={**3**,1,2} C2={**9**,10}	3+1=4
3	**3**,**10**	C1={**3**,1,2} C2={**10**,9}	3+1=4
4	**3**, **2**	…	…
…			
10			

7

$O(C_n^k k(n-k))$

Global minimum

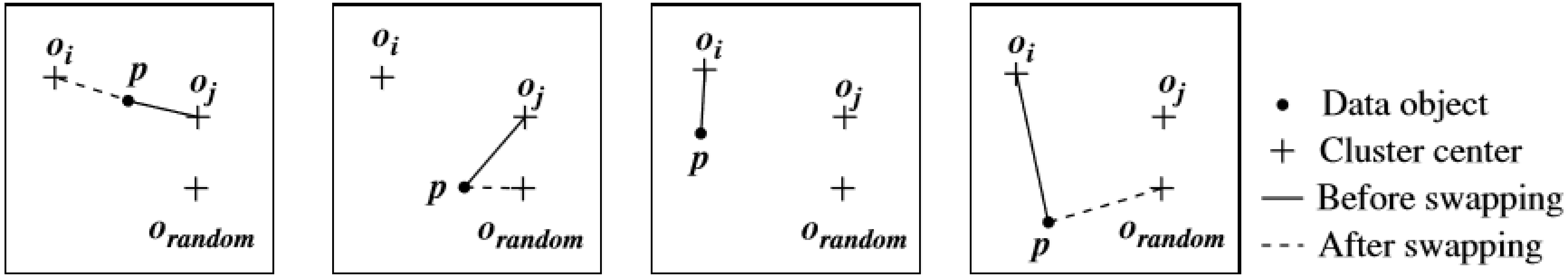
K-medoid – PAM

- Partitioning Around **Medoids**
  - Arbitrarily choose *k* medoids
  - **Repeat**
  - assign each remaining object to the cluster with the nearest medoid
  - randomly select a non-medoid object → O_{random}
  - compute the total cost *S* of swapping medoid O_j with O_{random}
  - if $S < 0$ then swap O_j with O_{random} to form the new set of medoids
  - **Until** no change

Greedy

Local Minimum

PAM – cost on p of swapping O_j with O_{random}



(a) Reassigned to o_i | (b) Reassigned to o_{random} | (c) No change | (d) Reassigned to o_{random}

Fig 10.4

(a)

Before: $d(O_j, p)$

After: $d(O_i, p)$

$d(O_j, p) < d(O_i, p)$

$C_p = d(O_i, p) - d(O_j, p)$ **+**

$m(p) = O_j$ and $d(O_{random}, p) > d(O_i, p)$

(b)

Before: $d(O_j, p)$

After: $d(O_{random}, p)$

$C_p = d(O_{random}, p) - d(O_j, p)$ **+/-**

$m(p) = O_j$ and $d(O_{random}, p) < d(O_j, p)$

(c)

Before: $d(O_i, p)$

After: $d(O_i, p)$

$C_p = 0$

$m(p) = O_i$ and $d(O_{random}, p) > d(O_i, p)$

(d)

Before: $d(O_i, p)$

After: $d(O_{random}, p)$

$d(O_{random}, p) < d(O_i, p)$

$C_p = d(O_{random}, p) - d(O_i, p)$ **-**

$m(p) = O_i$ and $d(O_{random}, p) < d(O_i, p)$

K-medoid – PAM IDEA

D = {o1, o2, o3, o4, o5}={3, 1, 9, 10, 2 }, K=2

Iteration	Medoids	Clustering		E/swapping cost
1	**3**, **1**	C1={**3**,9,10}	C2={**1**,2}	13+1=14 (E)

$d(3,9)+d(3,10)+d(2,1) = 14$

- Next step – swapping 3 with 9 (random chosen)
- Exhaustive search idea
  - Calculate E for new medoids (9,1)
  - 3 assigned to 1, 10 assigned to 9, 2 assigned to 1

$d(3,1)+d(9,10)+d(2,1) = 4$

- PAM Idea
  - Calculate the cost of the swapping
  - Save the time on assigning

cost on 3== $d(3,1)-d(3,9)$
cost on 10== $d(9,10)-d(3,10)$
cost on 2==$d(2,1)-d(2,1)$

K-medoid – PAM – cost - example

D = {o1, o2, o3, o4, o5}={3, 1, 9, 10, 2 }, K=2

Iteration	Medoids	Clustering		E/swapping cost
1	**3**, **1**	C1={**3**,9,10}	C2={**1**,2}	13+1=14 (E)

9, 1

Orandom = 9

Oj = 3

(a) $m(p) = O_j$ and $d(O_{random}, p) > d(O_j, p)$

(b) $m(p) = O_j$ and $d(O_{random}, p) < d(O_j, p)$

(c) $m(p) = O_i$ and $d(O_{random}, p) > d(O_i, p)$

(d) $m(p) = O_i$ and $d(O_{random}, p) < d(O_i, p)$

3 9

1

9 d(1,3) - d(3,9) = -4

3

10 case b $cost_{10}$ = d(9,10)-d(3,10)=-6

p

2 case c $cost_2$ = 0

p

Cost of 3←→9: -4+-6+0=-10

$O(n-k)$

Per swapping/solution

global minimum

$O(C_n^k(n-k))$

local minimum

$O(k(n-k))$

Per iteration/

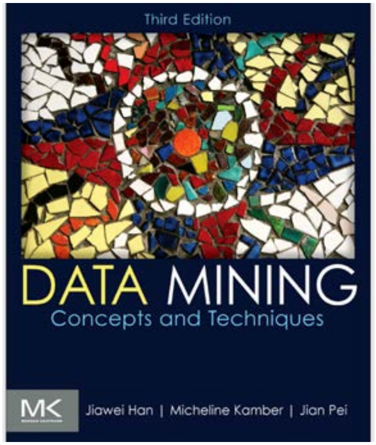
k solution

References

Section 10.2.2 *K*-Medoids: A Representative Object-Based Technique

from

Data Mining: Concepts and Techniques by Jiawei Han etc.



The e-book can be found via DigiPen Resource Library – Online Safari Books