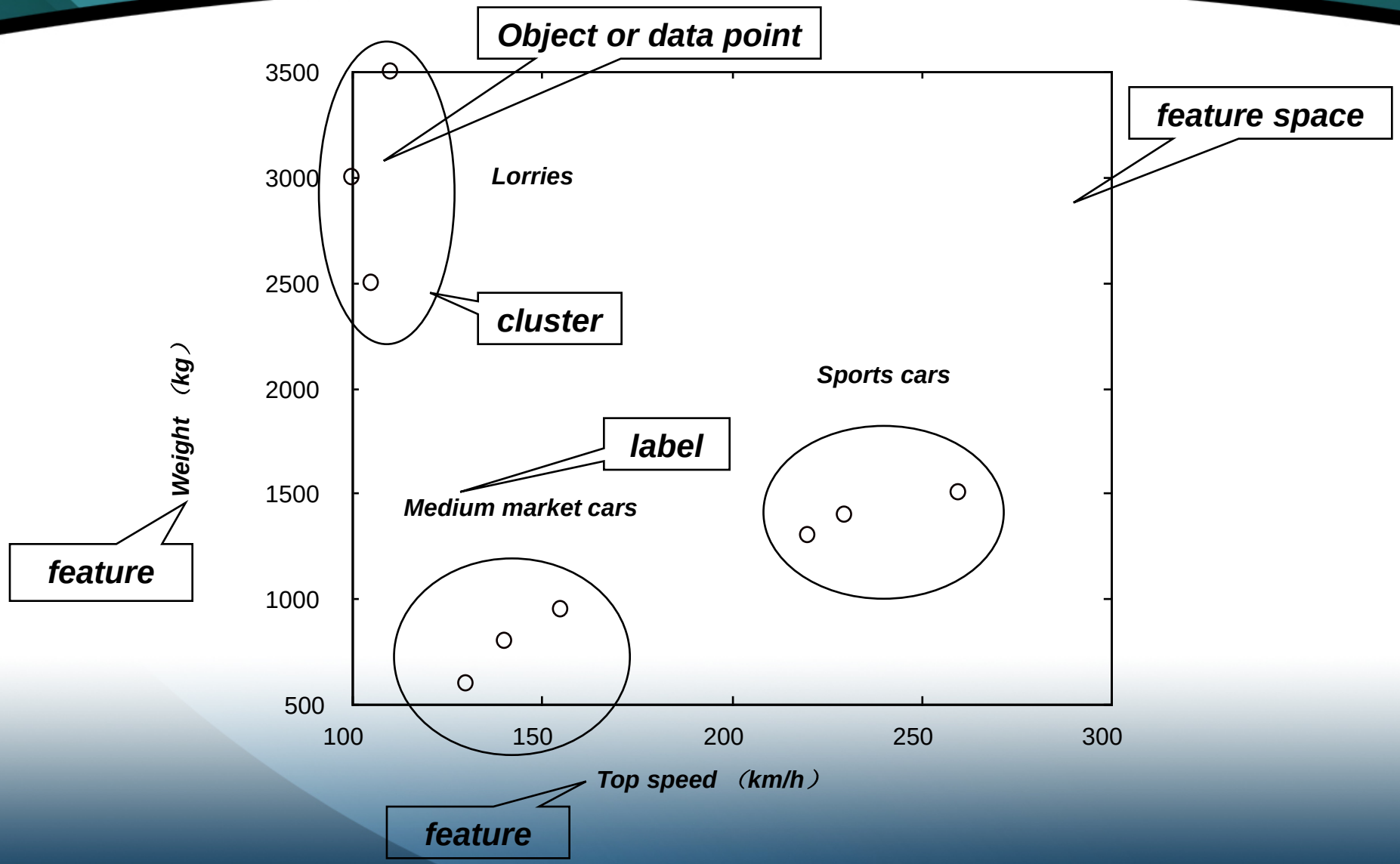
K-Means

Terminology



K-Means聚类法（C-Means）

- 将N个数据依照其数据特征聚类为K类的聚类算法，K为一正整数
- 目标在于求各个数据与其对应聚类中心点距离平方和的最小值

$$J=\sum_{i=1}^{K}J_i=\sum_{i=1}^{K}\sum_{j=1}^{N}w_{ji}\left\|X_j-C_i\right\|^2 \qquad (1)$$

  - J_i为第 i 类聚类的目标函数
  - K为聚类个数
  - X_j为第 j 个输入向量
  - C_i为第 i 个聚类中心（向量）
  - w_{ji} 为权重 （X_j 是否属于聚类C_i）

K-means的归属矩阵

$$\sum_{i=1}^{K} w_{ji} = 1, \forall j = 1, ..., N; \quad \sum_{i=1}^{K}\sum_{j=1}^{N} w_{ji} = N \qquad (2)$$

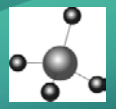
$$w_{ji} = \begin{cases} 1, & \text{if } \left\| X_j - C_i \right\| \le \left\| X_j - C_m \right\|, \quad \forall m \ne j \\ 0, & \text{otherwise} \end{cases} \qquad (3)$$

数据点 X_j

$$W = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

聚类C_i

K-means实现步骤



1. 随机选取k个数据点C_i，$i=1$，…，k，并将之分别视为各聚类的初始中心

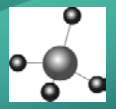
2. 决定各数据点所属之聚类，若数据点X_j判定属于第 i 聚类，则权重值$w_{ji}=1$，否则为 0

$$w_{ji}=\begin{cases}1, & \text{if } \left\|X_j-C_i\right\|\le\left\|X_j-C_m\right\|, \quad \forall m\neq j \\ 0, & \text{otherwise}\end{cases} \qquad (3)$$

且满足：

$$\sum_{i=1}^{k} w_{ji}=1,\ \forall j=1,\ ...,\ n, \quad \sum_{i=1}^{k}\sum_{j=1}^{n} w_{ji}=n \qquad (2)$$

K-means实现步骤

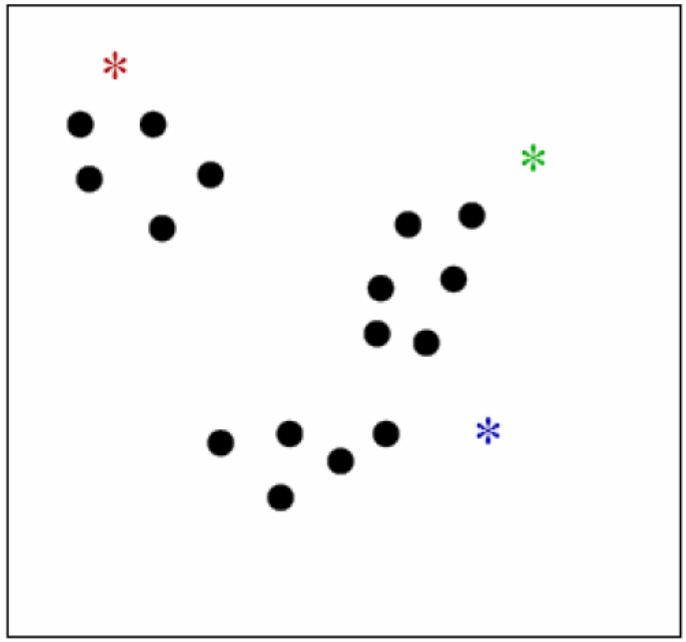


3. 由（1）式计算目标函数 J，如果 J 保持不变，代表聚类结果已经稳定不变，则可结束此迭代方法，否则进入步骤4

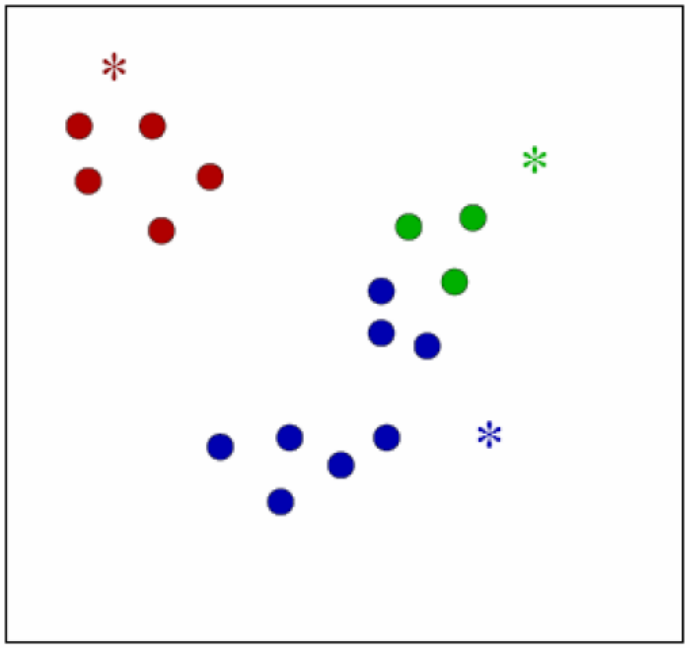
$$J = \sum_{i=1}^{k} J_i = \sum_{i=1}^{k} \sum_{j=1}^{n} w_{ji} \left\| X_j - C_i \right\|^2 \qquad (1)$$

4. 以（4）式更新聚类的中心点。回到步骤2

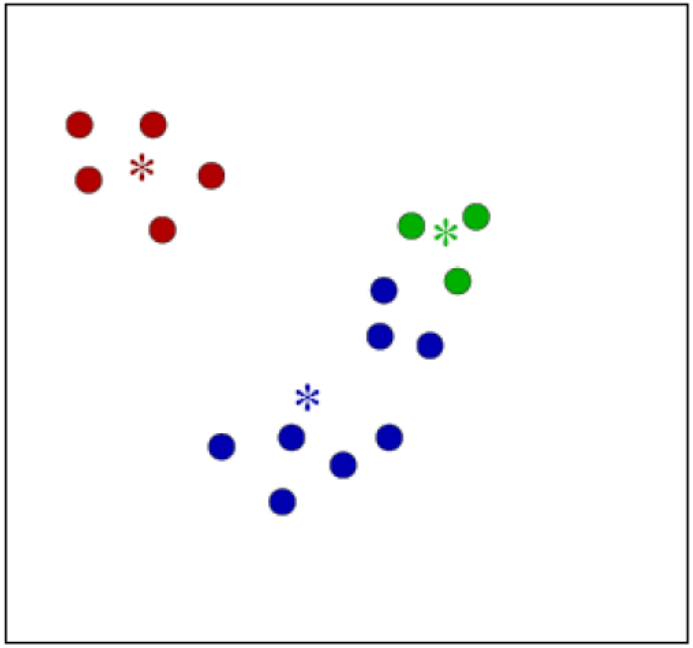
$$C_i = \frac{\sum_{j=1}^{n} w_{ji} X_j}{\sum_{j=1}^{n} w_{ji}} \qquad (4)$$



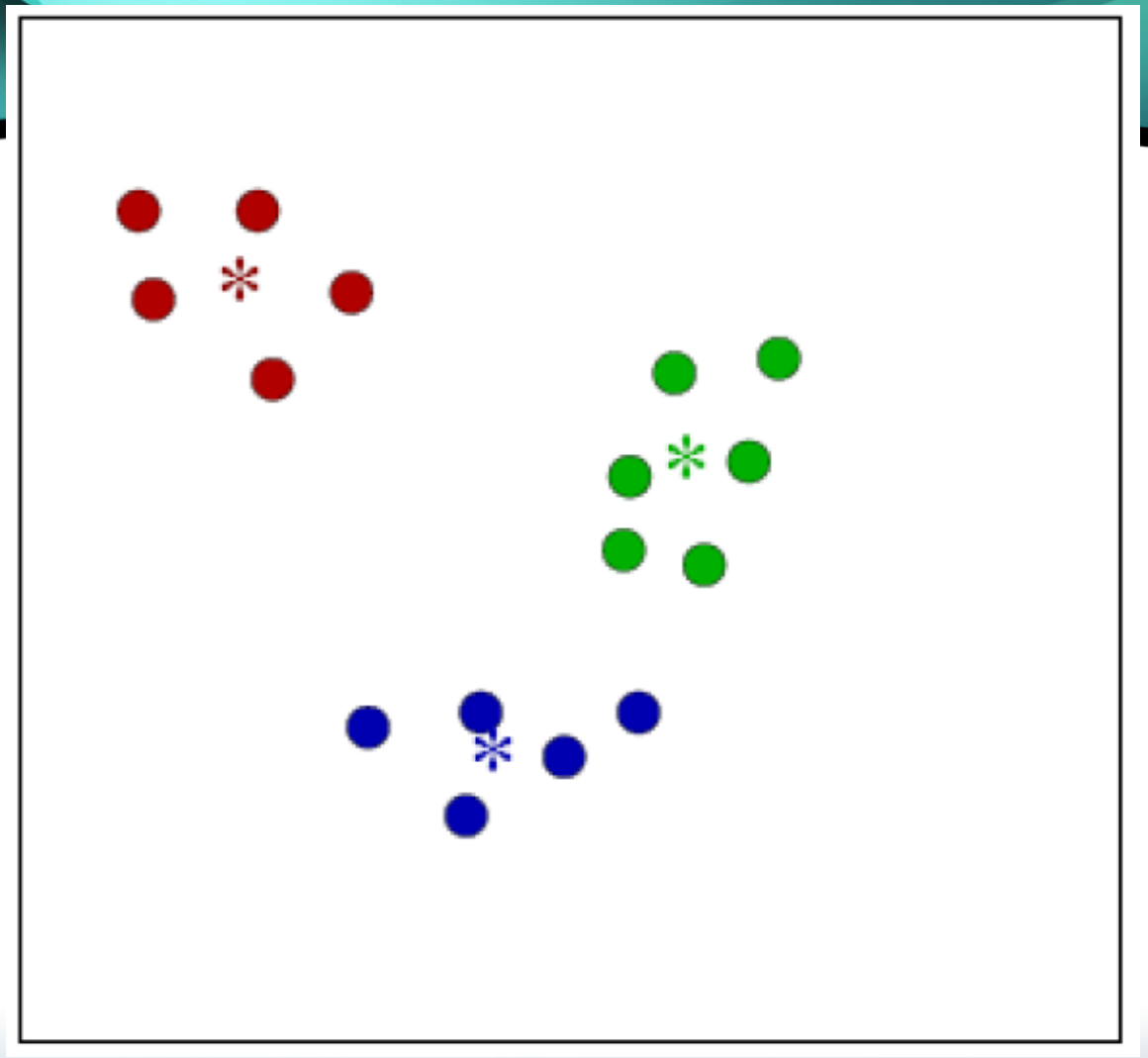
Initialize representatives (“means”)



Assign to nearest representative



Re-estimate means



N 次迭代以后

K-means编程步骤

1. 设定聚类数目K，最大执行步骤t_{max}，一个很小的容忍误差$\varepsilon>0$
2. 决定聚类中心起始位置$C_j(0)$，$0<j\le K$
3. *for* $t=1,.....,t_{max}$

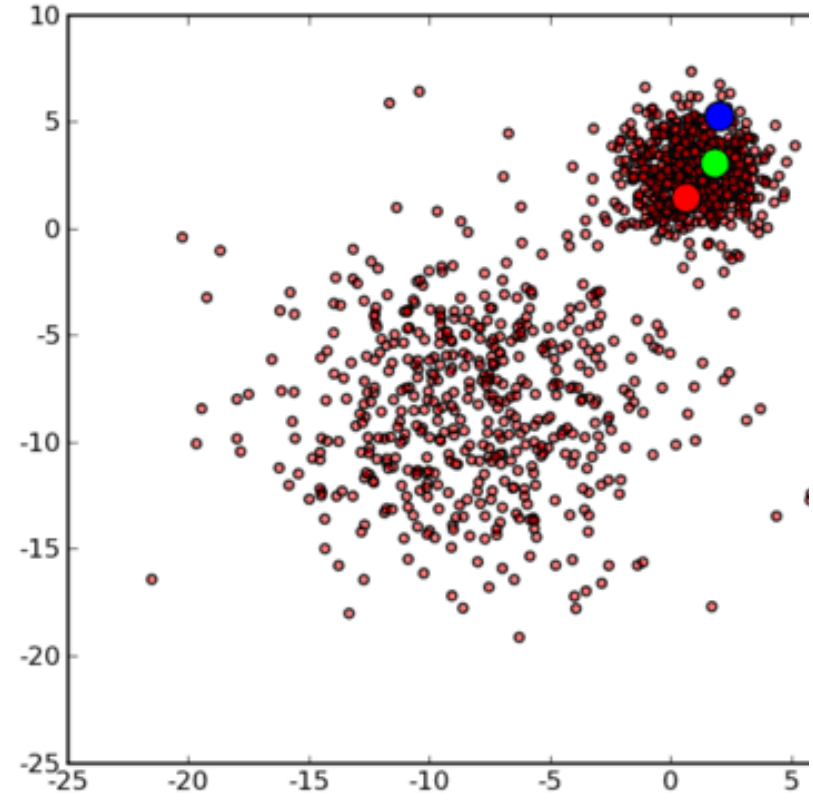
- （A）*for* $j=1,.....,N$
  - （i）计算各数据点到聚类中心的距离 $d_{ij}^{(t)}=\left\|X_j-C_i^{(t-1)}\right\|;i=1,...K$
  - （ii）计算数据点属于哪一聚类（隶属度矩阵）$w_{ji}=\begin{cases}1,\arg\min_{i=1}^{K}\{d_{ji}^{(t)}\}\\0,otherwise\end{cases}$
- （B）更新聚类中心 $C_i^t=\sum_{j=1}^{N}w_{ji}^{(t)}X_j\Big/\sum_{j=1}^{N}w_{ji}^{(t)};i=1,...K$

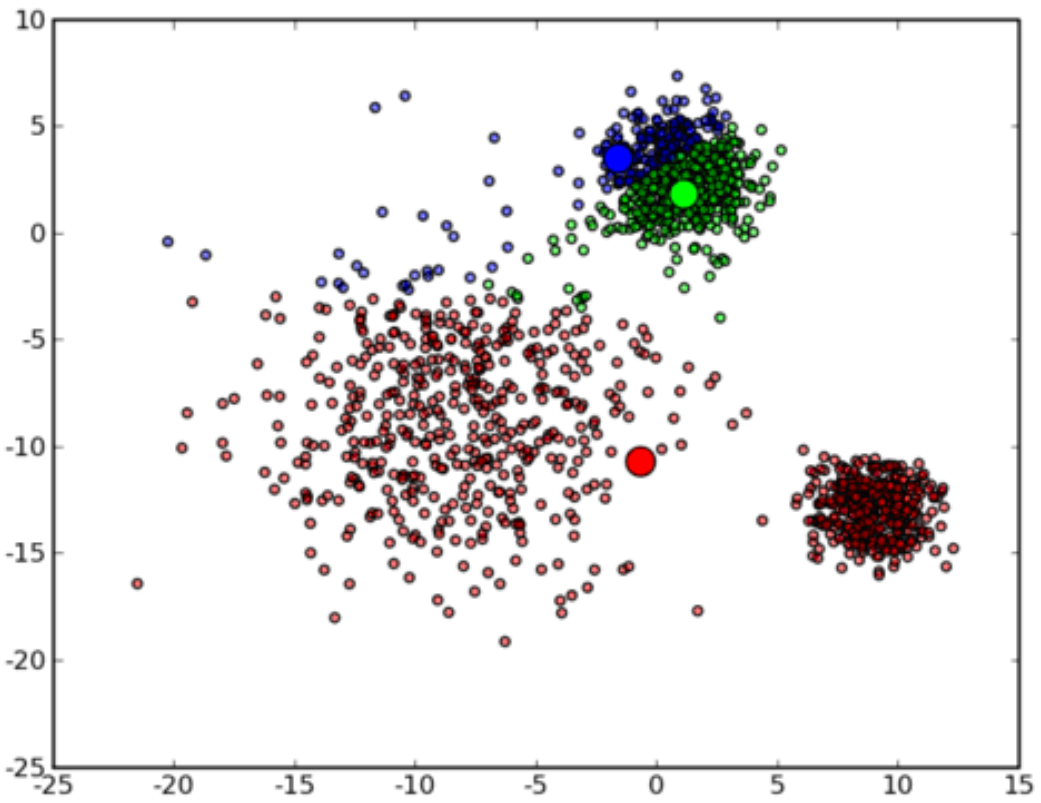
1. （C）计算收敛准则，若 $E(t)=\left\|J^{(t)}-J^{(t-1)}\right\|<\varepsilon$ 成立则停止运算，否则进行下一轮迭代 $\mathrm{E(t)}=\left\|C^{(t)}-C^{(t-1)}\right\|<\varepsilon$

使用*K-Means*聚类法

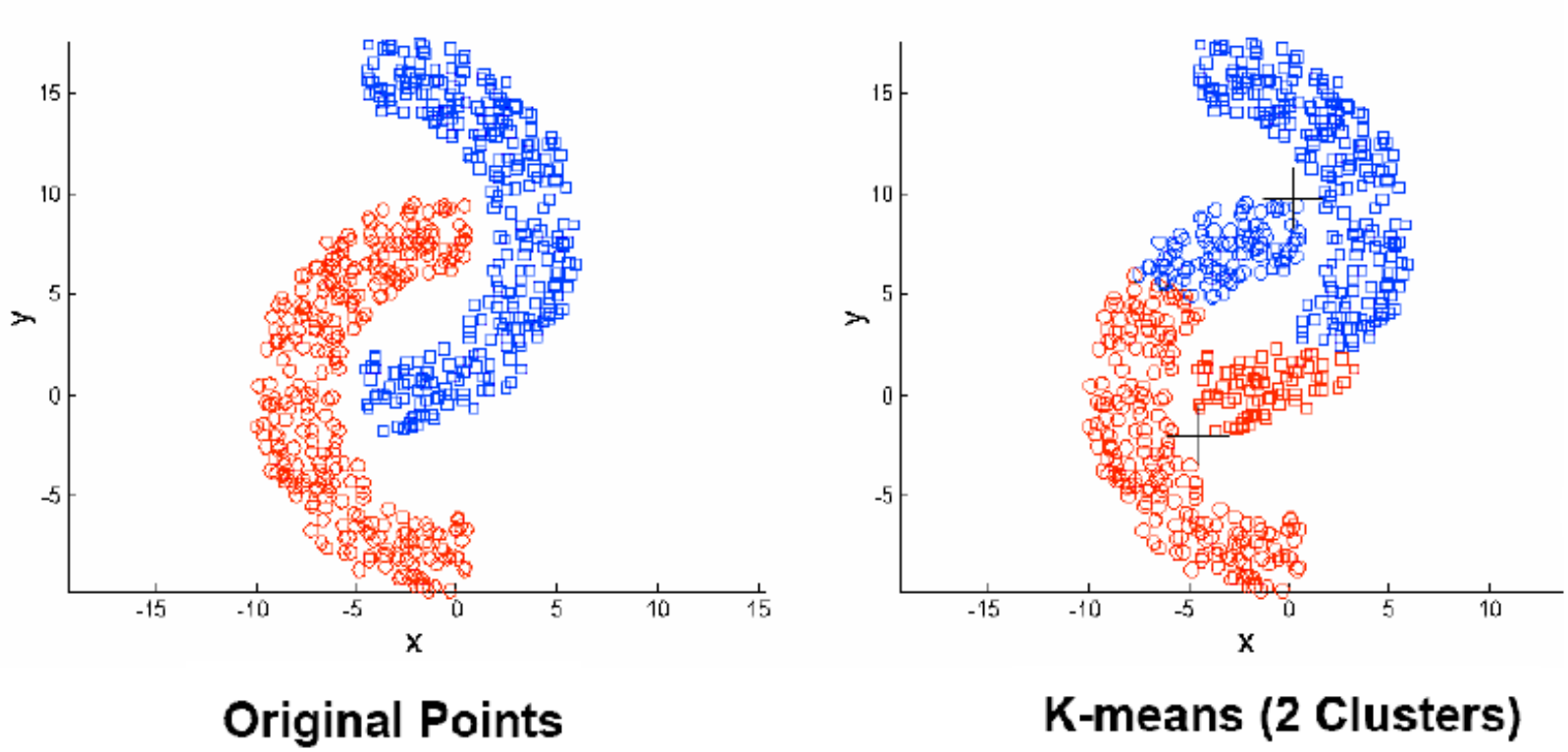
- 需事先确定聚类的数目K
- 若初始聚类中心位置不理想，使得目标函数J落入局部解，最后分类出来的群集将不甚理想

使用*K-Means*聚类法





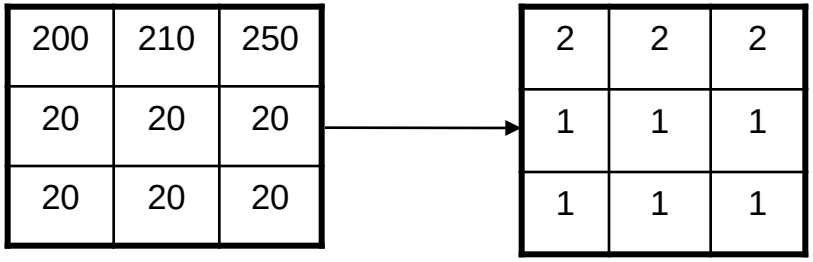
使用*K-Means*聚类法



***Cluster* 形状以类圆形为主**

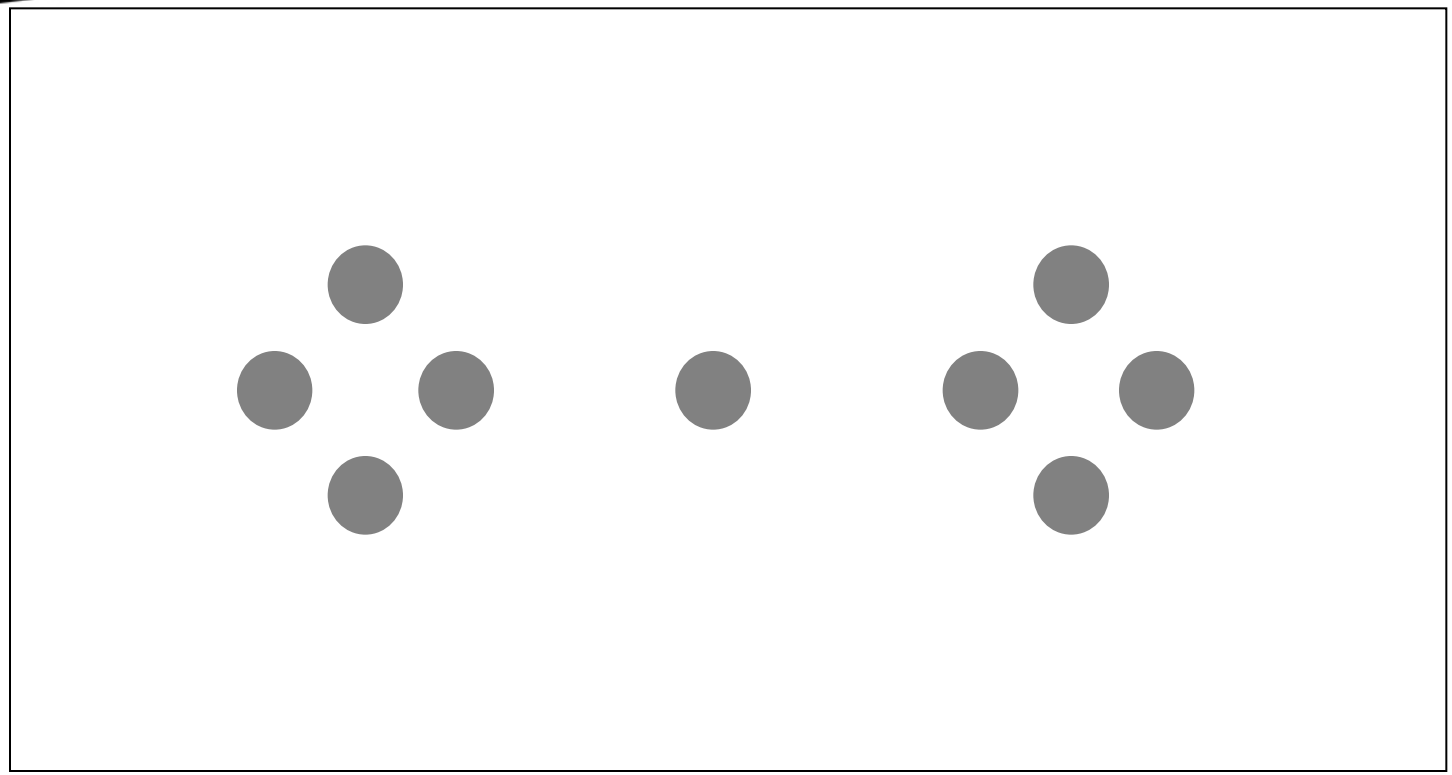
使用*K-Means*聚类法

*K-Means*聚类分析是一种硬划分（***Hard Clustering***），它把每个待辨识的对象严格地划分到某个类中，具有非此即彼的性质



Cluster 1: mean=35

Cluster 2: mean=230

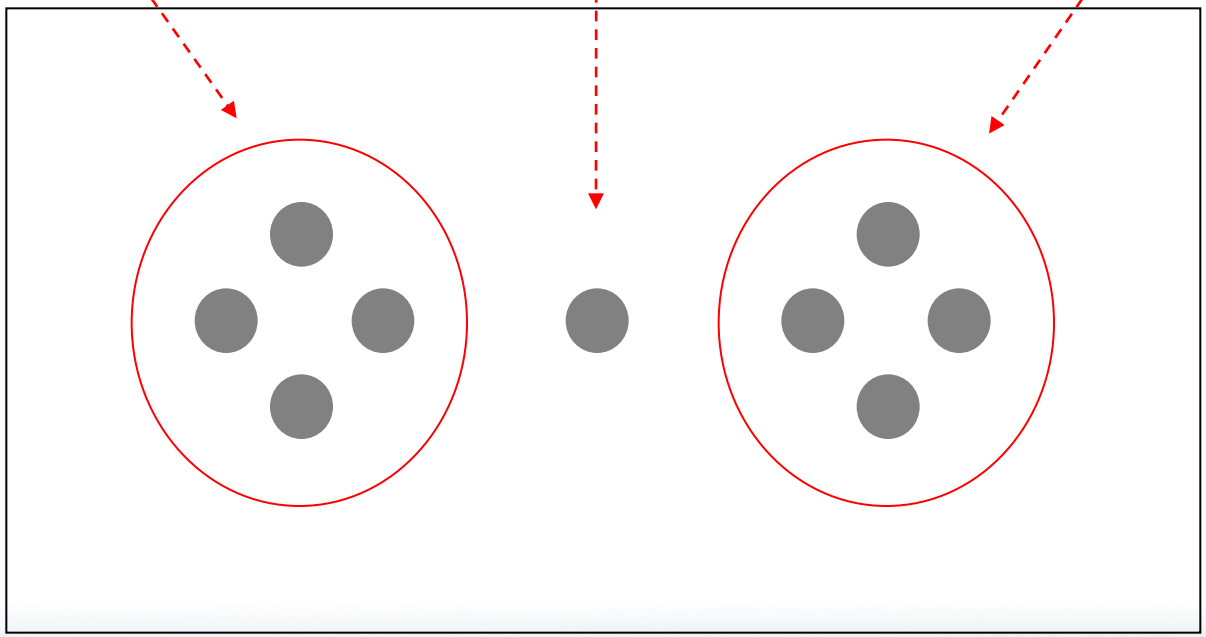


蝴蝶型数据集

属于*Cluster* 1 *or* 2 ?

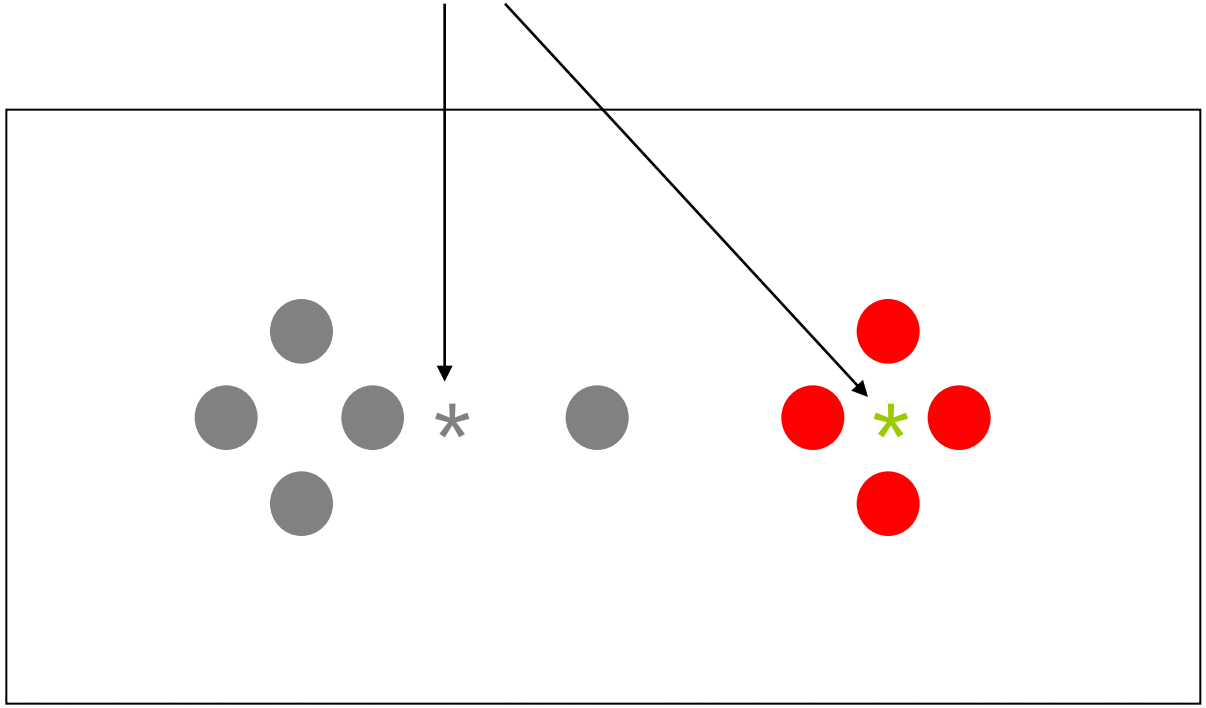
很明确的属于 *Cluster* 1

很明确的属于*Cluster* 2

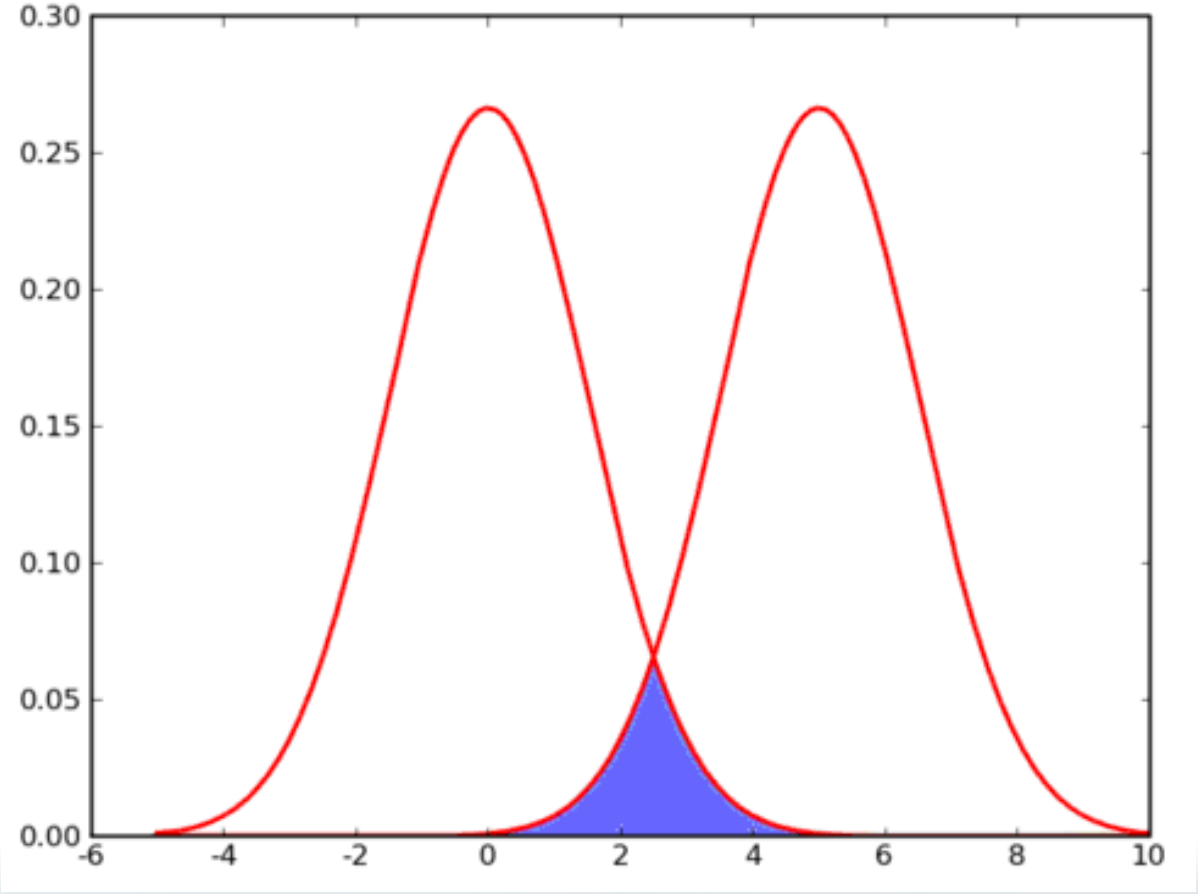


蝴蝶型数据集

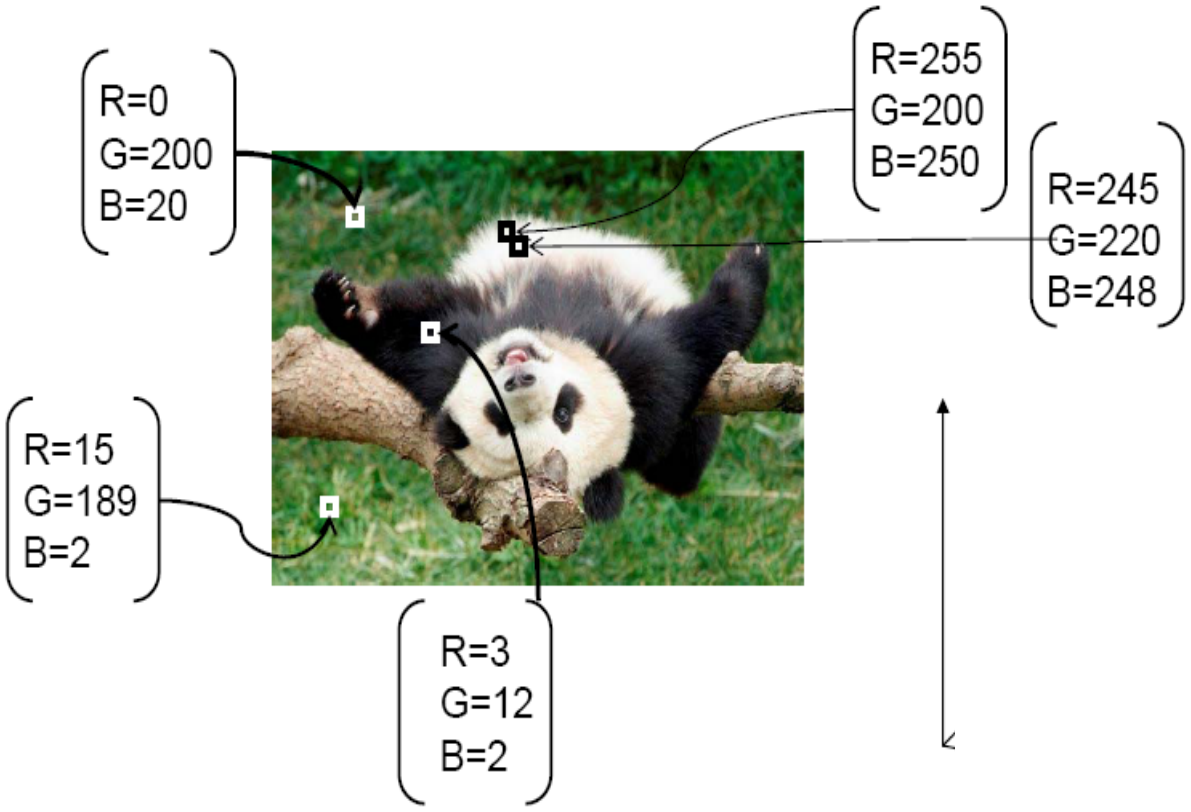
聚类中心



蝴蝶型数据集


0.30
0.25
0.20
0.15
0.10
0.05
0.00
-6
-4
-2
0
2
4
6
8
10

Feature Space



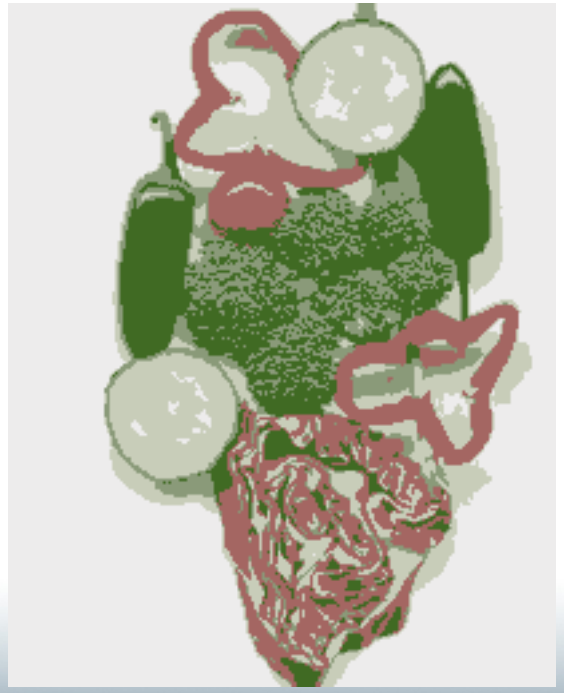
Source: K. Grauman

K-means clustering using intensity alone and color alone

Image

Clusters on intensity

Clusters on color

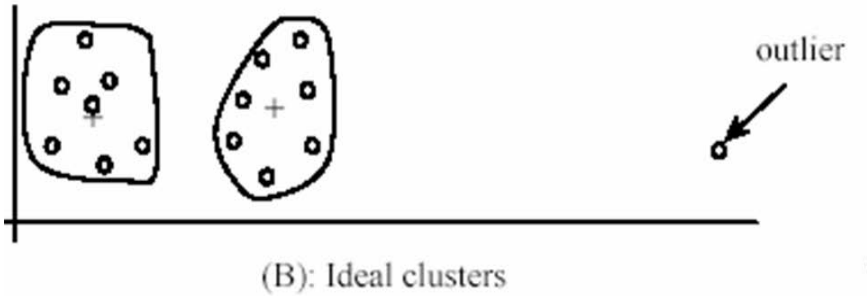


Segmentation as clustering

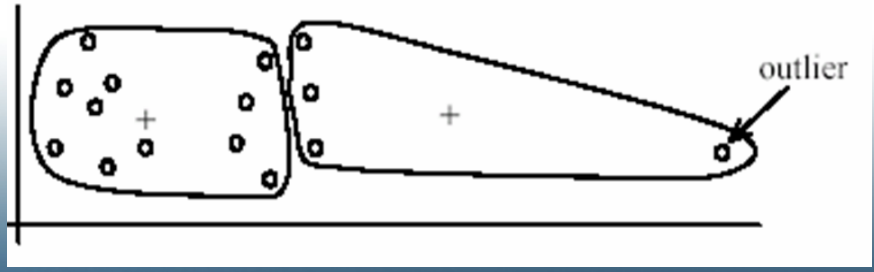
- Clustering based on (r,g,b,x,y) values enforces more spatial coherence

K-Means pros and cons

- Pros
  - Simple and fast
  - Easy to implement
- Cons
  - Need to choose K
  - Sensitive to outliers
- Usage
  - Rarely used for pixel segmentation



(B): Ideal clusters



*Fuzzy C-Means*聚类法

Dunn 利用 Ruspini 提出的模糊划分的概念, 将硬聚类推广到模糊聚类, 1973年 Jim Bezdek 将 Dunn 的工作推广到基于模糊度 *m* 的一般 *Fuzzy C-Means* 形式，其目标函数定义如同*K-Means*聚类法，但其权重矩阵 *W* 不再是*二元矩阵*，而是应用了模糊理论的概念，使得每一输入向量不再仅归属于某一特定的聚类，而以其归属程度来表现属于各聚类的程度

http://www.cs.uwf.edu/~jbezdek/

Fuzzy C-Means聚类法

目标函数 J 为（5）式

$$J = \sum_{i=1}^{K} J_i = \sum_{i=1}^{K} \left(\sum_{j=1}^{N} w_{ji}^{m} \left\| X_j - C_i \right\|^2 \right) \qquad (5)$$

其中：

- X_j 为数据点
- C_i 为聚类中心点
- N 为数据个数
- K 为聚类中心点个数
- m 为权重指数