AI300学院

机器学习之无监督学习

聚类算法

倪冰冰
上海交通大学

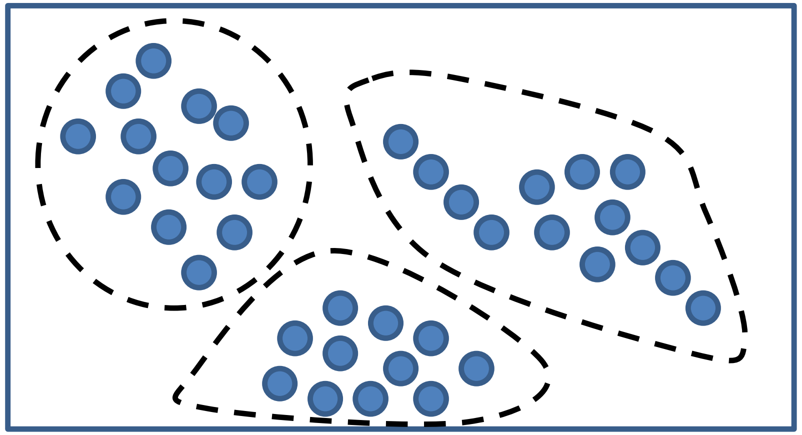
引言

- 什么是无监督学习?

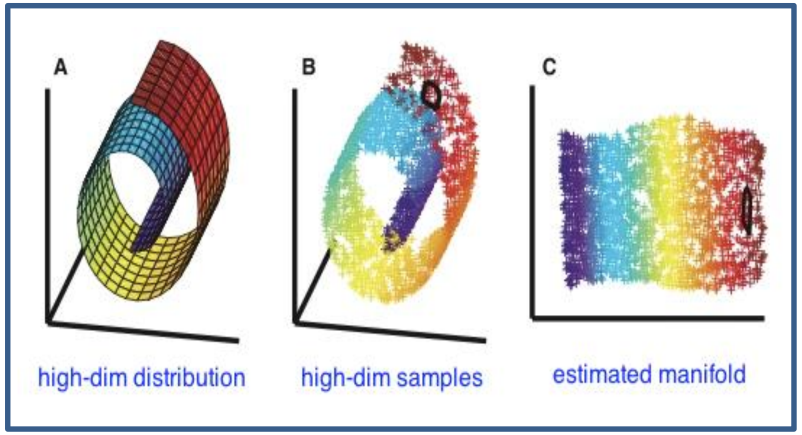
  1.数据没有明确的标签信息。

  2.我们希望仅依赖数据本身来探索其具有的内在结构信息。

- 无监督学习的种类有哪些?



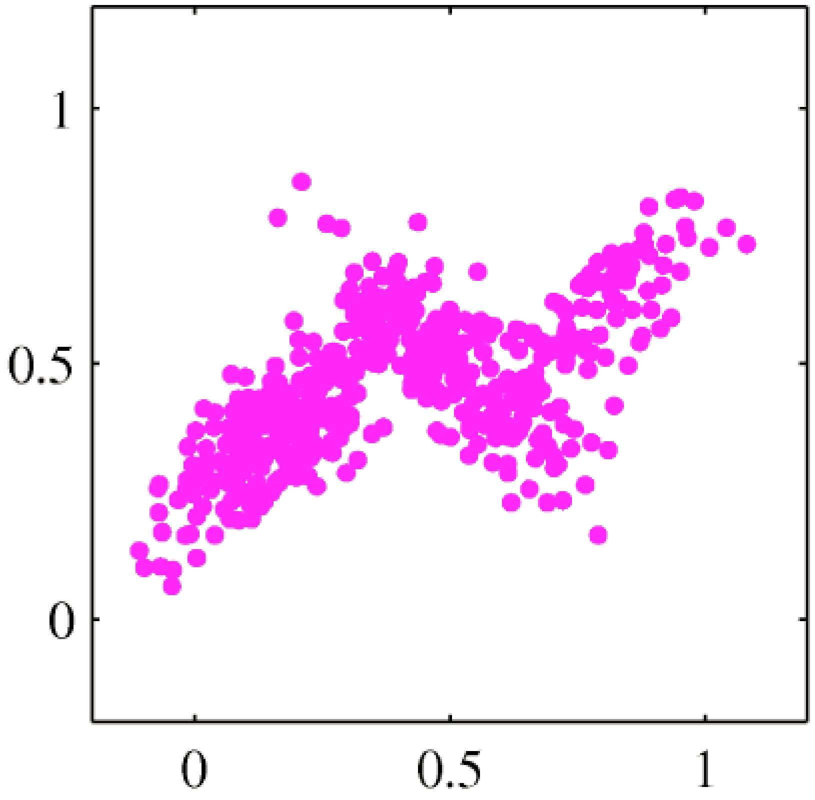
聚类学习

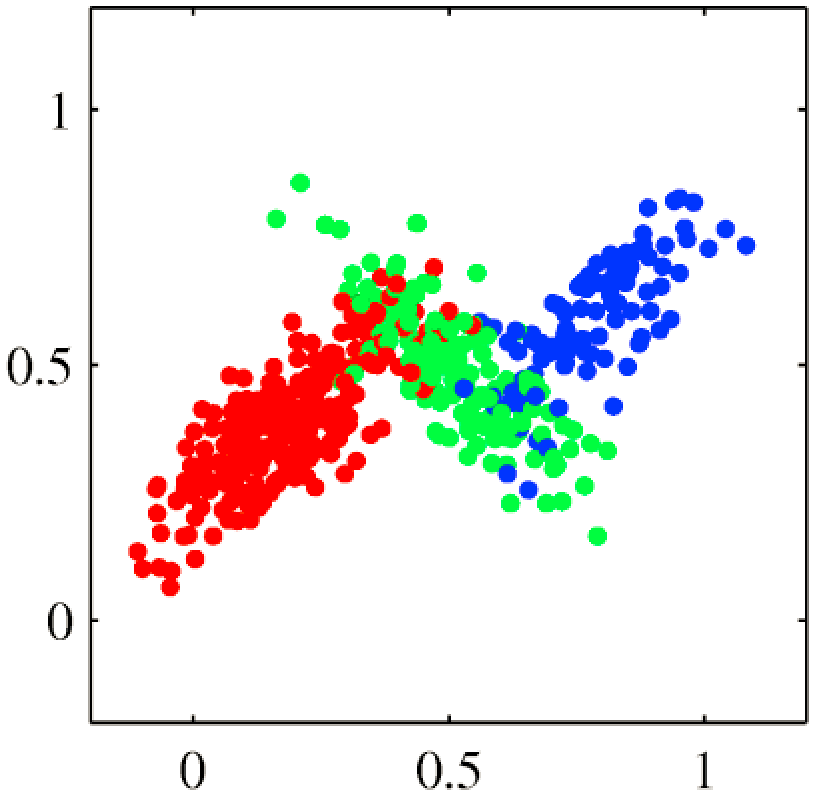


表征学习(降维)

聚类分析

- 聚类分析的基本目标是通过对采样数据分类，使得属于同一类别的数据相似，而不同类别间的数据不同。





AI300学院

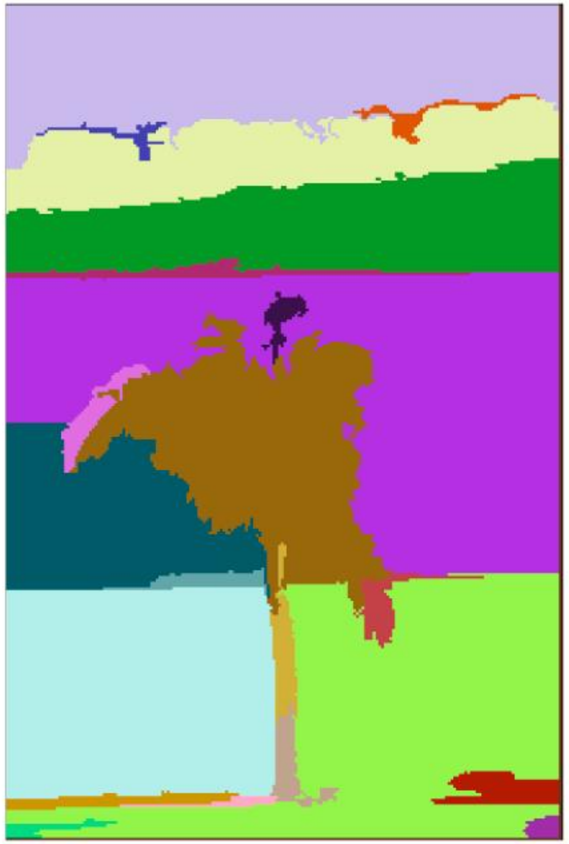
课程脉络

聚类分析

- 层次聚类(Agglomerative Clustering)
- K-均值聚类(K-Means)
- 高斯混合模型（GMM）
- Expectation-Maximization
- 谱聚类(Spectral Methods)

聚类分析

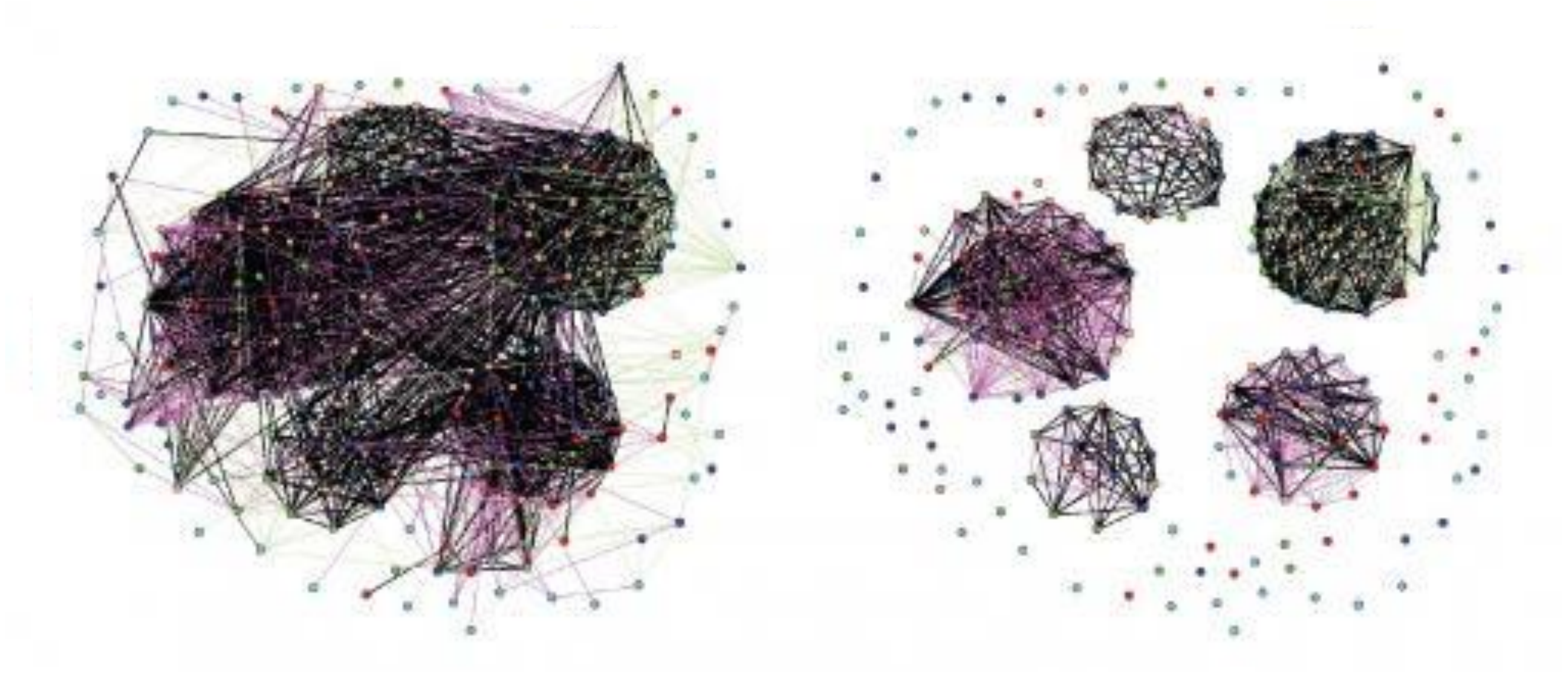
- 聚类分析的实际应用



图像分割

聚类分析

- 聚类分析的实际应用



用户聚类

AI300学院

聚类分析

何谓远近?

⇩

距离度量、特征

如何聚类?

⇩

算法

聚类分析

- 特征空间

每个数据样本$\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})$中的$x_{i1}, x_{i2}, \ldots, x_{id}$表示的是不同的特征维度。比如对图像中的某个物体来说，可以用位置、颜色、纹理、运动向量以及大小等作为不同的特征维度。

如何选择并量化特征就涉及到所谓的特征空间，我们经过采样得到的数据可以看作是特征空间中的不同点。

在这种情况下，不同数据之间的距离就可以理解为在特征空间中不同点之间的距离

聚类分析

- 聚类分析的主要挑战

  1.数据的相似性体现在哪些方面?

  2.如果我们已知数据两两之间的相似性，如何对全部数据进行整体上的分类?

假定我们已有两个数据样本：$\boldsymbol{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})$, $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \dots, x_{jd})$，一种典型的相似度衡量标准是使用欧式距离，表述如下：

$$D(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{id} - x_{jd})^2}$$

聚类分析

- 相似度衡量

以距离为依据 $aff(\mathbf{x},\mathbf{y}) = \exp\left\{-(\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})/2\sigma_d^2\right\}$

以强度为依据 $aff(\mathbf{x},\mathbf{y}) = \exp\left\{-(I(\mathbf{x})-I(\mathbf{y}))^2/2\sigma_I^2\right\}$

以颜色为依据 $aff(\mathbf{x},\mathbf{y}) = \exp\left\{-(dist(c(\mathbf{x})-c(\mathbf{y}))^2/2\sigma_c^2\right\}$

以纹理为依据 $aff(\mathbf{x},\mathbf{y}) = \exp\left\{-(dist(f(\mathbf{x})-f(\mathbf{y}))^2/2\sigma_I^2\right\}$

聚类分析

- 如何聚类

 给定距离度量，如何grouping?

 -自上而下：K-means clustering。迭代式地将样本点归类到其最近的聚类中心。

 -自下而上：Hierarchical clustering。从样本开始不断“抱团”。

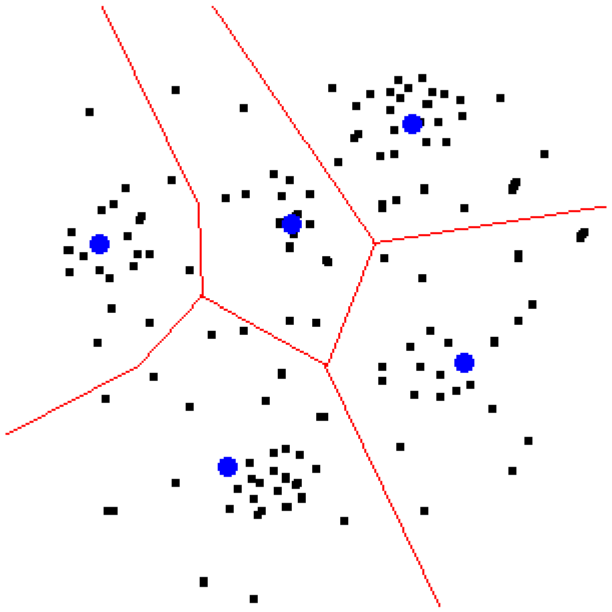
AI300学院

K-Means前辈

- 给定一组共n个采样数据: $H = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$，每个样本的特征维度为d。
- 将数据分成c个互不重叠的子集: $\{H_1, H_2, \dots, H_c\}$
- 目标：属于同一聚类的样本数据应当尽可能地“相似”，而属于不同聚类的样本数据应当尽可能地“不同”。

$$J_e \triangleq \sum_{i=1}^{c} \sum_{\mathbf{x} \in H_i} \|\mathbf{x} - \mathbf{m}_i\|^2$$

$\boldsymbol{m}_i$: 第i个聚类的中心点



K-Means前辈

- 基本思路

  1.确定一个初始分组

  2.将采样数据从某一分组分类到另一分组，目的是使得损失函数更小。

- 必要条件

$$J_e = \sum_{i=1}^{c} J_i \ \text{ where } \ J_i = \sum_{\mathbf{x} \in H_i} \|\mathbf{x} - \mathbf{m}_i\|^2 \ ; \quad \mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in H_i} \mathbf{x}$$

- 假定样本$\hat{x}$ 当前出现在H_i，并被移动到H_j，那么新的均值由如下公式更新：

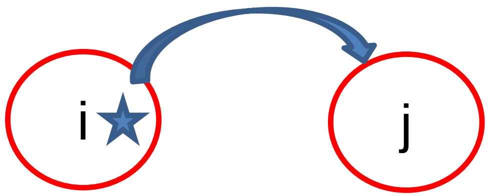
$$\begin{aligned} \mathbf{m}_j^{\star} &= \frac{n_j \mathbf{m}_j + \hat{\mathbf{x}}}{n_j + 1} = \mathbf{m}_j + \frac{n_j \mathbf{m}_j + \hat{\mathbf{x}}}{n_j + 1} - \mathbf{m}_j \\ &= \mathbf{m}_j + \frac{\hat{\mathbf{x}} - \mathbf{m}_j}{n_j + 1} \end{aligned}$$

$$\boxed{J_e \triangleq \sum_{i=1}^{c} \sum_{\mathbf{x} \in H_i} \|\mathbf{x} - \mathbf{m}_i\|^2}$$

AI300学院

K-Means前辈

- J_j按照如下公式更新



$$\mathbf{m}_j^\star = \mathbf{m}_j + \frac{\hat{\mathbf{x}} - \mathbf{m}_j}{n_j + 1}$$

$$\begin{aligned}
J_j^\star &= \sum_{\mathbf{x}\in H_j} \|\mathbf{x} - \mathbf{m}_j^\star\|^2 + \|\hat{\mathbf{x}} - \mathbf{m}_j^\star\|^2 \\
&= \sum_{\mathbf{x}\in H_j} \|\mathbf{x} - \mathbf{m}_j - \frac{\hat{\mathbf{x}} - \mathbf{m}_j}{n_j + 1}\|^2 + \|\frac{n_j}{n_j + 1}(\hat{\mathbf{x}} - \mathbf{m}_j)\|^2 \\
&= \sum_{\mathbf{x}\in H_j} \|\mathbf{x} - \mathbf{m}_j\|^2 - \frac{2}{n_j + 1} \sum_{\mathbf{x}\in H_j} (\hat{\mathbf{x}} - \mathbf{m}_j)^T (\mathbf{x} - \mathbf{m}_j) \\
&\quad + \frac{1}{(n_j + 1)^2} \sum_{\mathbf{x}\in H_j} \|\hat{\mathbf{x}} - \mathbf{m}_j\|^2 + \frac{n_j^2}{(n_j + 1)^2}\|\hat{\mathbf{x}} - \mathbf{m}_j\|^2 \\
&= J_j + \frac{n_j}{(n_j + 1)^2}\|\hat{\mathbf{x}} - \mathbf{m}_j\|^2 + \frac{n_j^2}{(n_j + 1)^2}\|\hat{\mathbf{x}} - \mathbf{m}_j\|^2
\end{aligned}$$

$$\Longleftrightarrow \quad J_j^\star = J_j + \frac{n_j}{(n_j + 1)}\|\hat{\mathbf{x}} - \mathbf{m}_j\|^2$$

- 相似地，J_i 更新为： $J_i^\star = J_i - \dfrac{n_i}{(n_i - 1)}\|\hat{\mathbf{x}} - \mathbf{m}_i\|^2$

K-Means前辈

Procedure: *Basic Minimum Squared Error*

1. Select an initial partition of the n samples into c clusters and compute $\mathbf{m}_1, \ldots, \mathbf{m}_c$, and J_e.
2. Select the next candidate sample $\hat{\mathbf{x}}$. Suppose that currently $\hat{\mathbf{x}} \in H_i$.
3. If $n_i = 1$, *goto* 6; otherwise compute

$$\rho_j = \begin{cases} \frac{n_j}{(n_j+1)} \|\hat{\mathbf{x}} - \mathbf{m}_j\|^2, & \forall j \neq i \\ \frac{n_i}{(n_i-1)} \|\hat{\mathbf{x}} - \mathbf{m}_i\|^2, & j = i \end{cases}$$

4. Transfer $\hat{\mathbf{x}}$ to H_k whose ρ_k is smallest.
5. Update $\mathbf{m}_i$, $\mathbf{m}_k$ and J_e using (3) – (6).
6. If J_e has not changed in n attempts, stop. Else, *goto* 2. 目标不再下降!

(Other reasonable stopping criteria can also be used.)

- 可以保证收敛
  - ✓ 损失函数下有界
  - ✓ 每一步保证减少函数

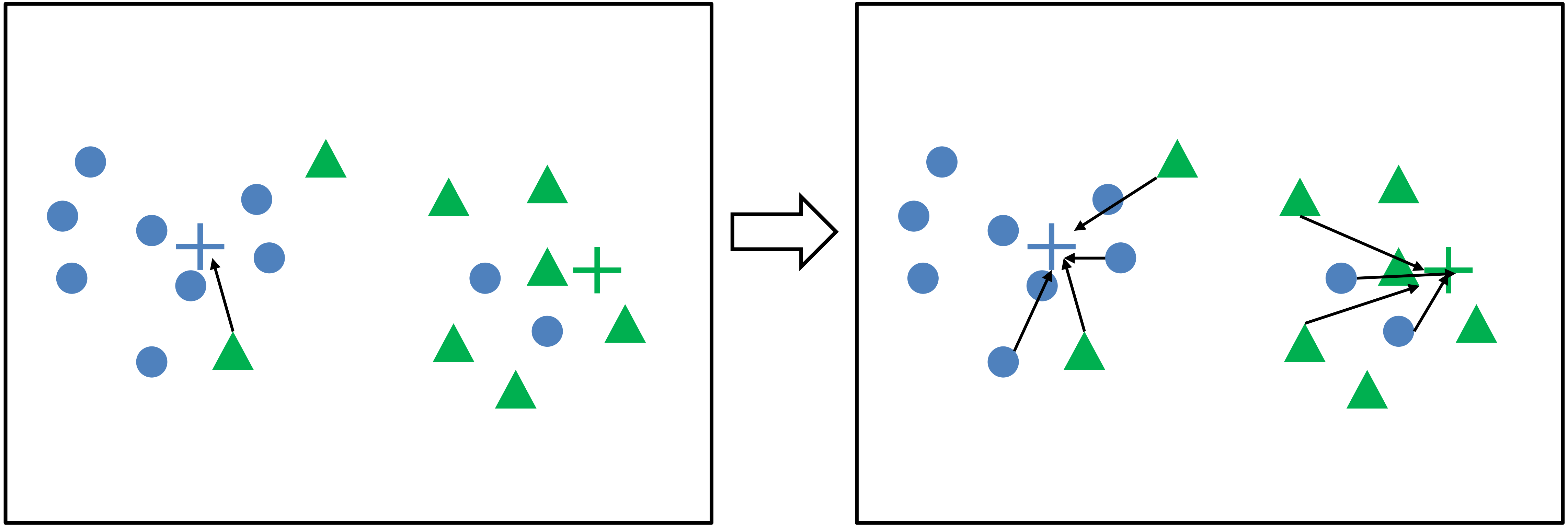
$$\boxed{J_e \triangleq \sum_{i=1}^{c} \sum_{\mathbf{x} \in H_i} \|\mathbf{x} - \mathbf{m}_i\|^2}$$

但是，每次迭代一个样本，收敛速度巨慢无比！

AI300学院

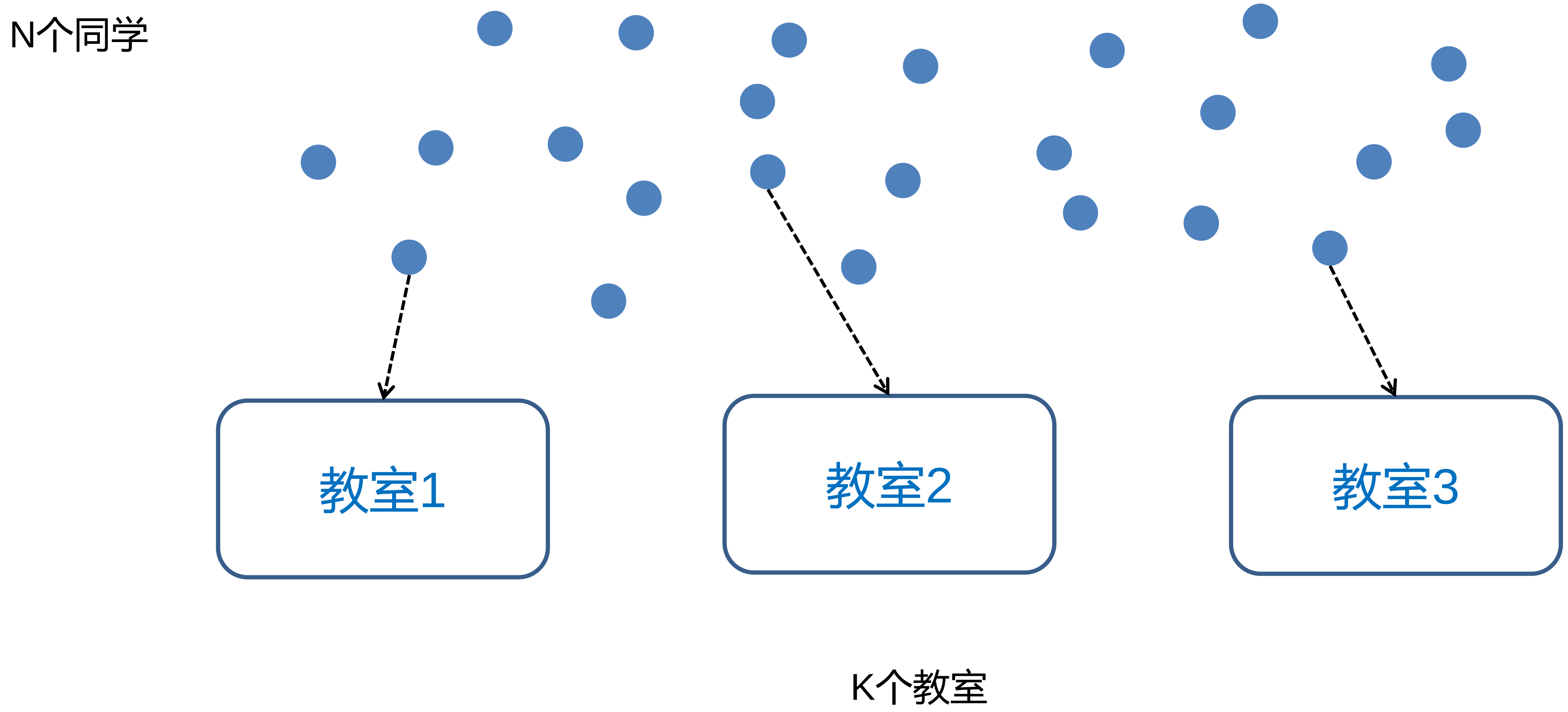
K-Means

- 改进方法



单样本迭代（sequential mode）　　　　批量迭代（batch mode）

K-Means



K-Means

- 定义赋值变量 $\{r_{ik}\}$, 比如, $r_{ik} = 1$ 表示第i个数据样本$\boldsymbol{x}_i$ 被归类于第 k个聚类，否则 为 0;且满足 $\sum_k r_{ik} = 1$

- K-Means算法流程:
  1. 初始化赋值变量 $\{r_{ik}\}$ 和聚类中心 $\{\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_c\}$
  2. 重复如下算法流程
     E-step：将聚类中心固定，将每一个采样点归类到距离其最近的聚类中心，比如，$r_{ik} = 1$，若 $k = \underset{l}{argmin}||\boldsymbol{x}_i - \boldsymbol{m}_l||$。
     M-step： 固定$\{r_{ik}\}$，重新计算聚类中心$\boldsymbol{m}_k = \frac{\sum_i x_i r_{ik}}{\sum_i r_{ik}}$
  3. 当模型收敛时算法停止。

损失函数：$J = \sum_i \sum_k \|\boldsymbol{x}_i - \boldsymbol{m}_k\| r_{ik}$

K-Means

- 模型可收敛性保证
  1. E-step进行之后有:

$$J_e(old) = \sum_i \sum_k r_{ik}^{(old)} \|\mathbf{x}_i - \mathbf{m}_k\| \geq \sum_i \sum_k r_{ik}^{(new)} \|\mathbf{x}_i - \mathbf{m}_k\| = J_e(new)$$

  2. M-step进行之后有:

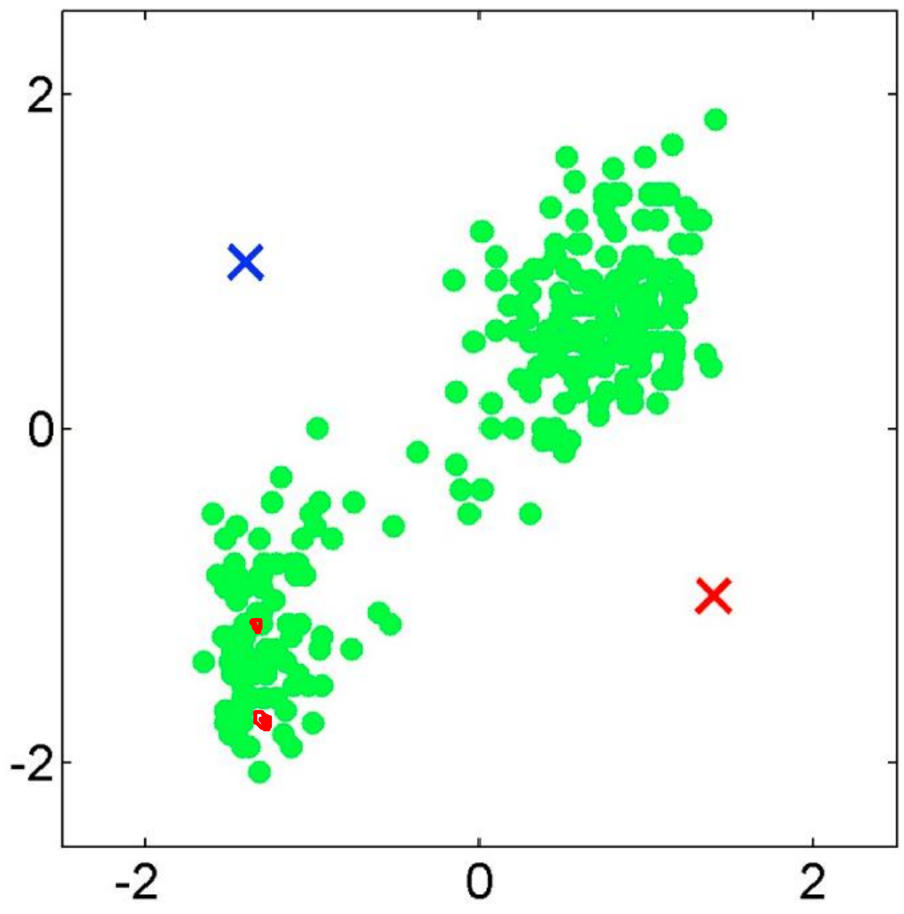
$$\mathbf{m_k}^{(new)} = \frac{\sum_{i \in H_k} \mathbf{x}_i}{|\{i \in H_k\}|}$$

$$\begin{aligned} J_e^k(old) &= \sum_{i \in H_k} \|\mathbf{x}_i - \mathbf{m_k}^{(old)}\|_2^2 \\ &= \sum_{i \in H_k} \|\mathbf{x}_i - \mathbf{m_k}^{(new)} + \mathbf{m_k}^{(new)} + \mathbf{m_k}^{(old)}\|_2^2 \\ &= \underbrace{\sum_{i \in H_k} \|\mathbf{x}_i - \mathbf{m_k}^{(new)}\|_2^2}_{J_e^k(new)} + \underbrace{\sum_{i \in H_k} \|\mathbf{m}_k^{(old)} - \mathbf{m_k}^{(new)}\|_2^2}_{\geq 0} + \underbrace{\sum_{i \in H_k} (\mathbf{x}_i - \mathbf{m_k}^{(new)})^T (\mathbf{m}_k^{(old)} - \mathbf{m_k}^{(new)})}_{= 0} \end{aligned}$$

$$\Longleftrightarrow \quad J_e^k(old) \geq J_e^k(new)$$

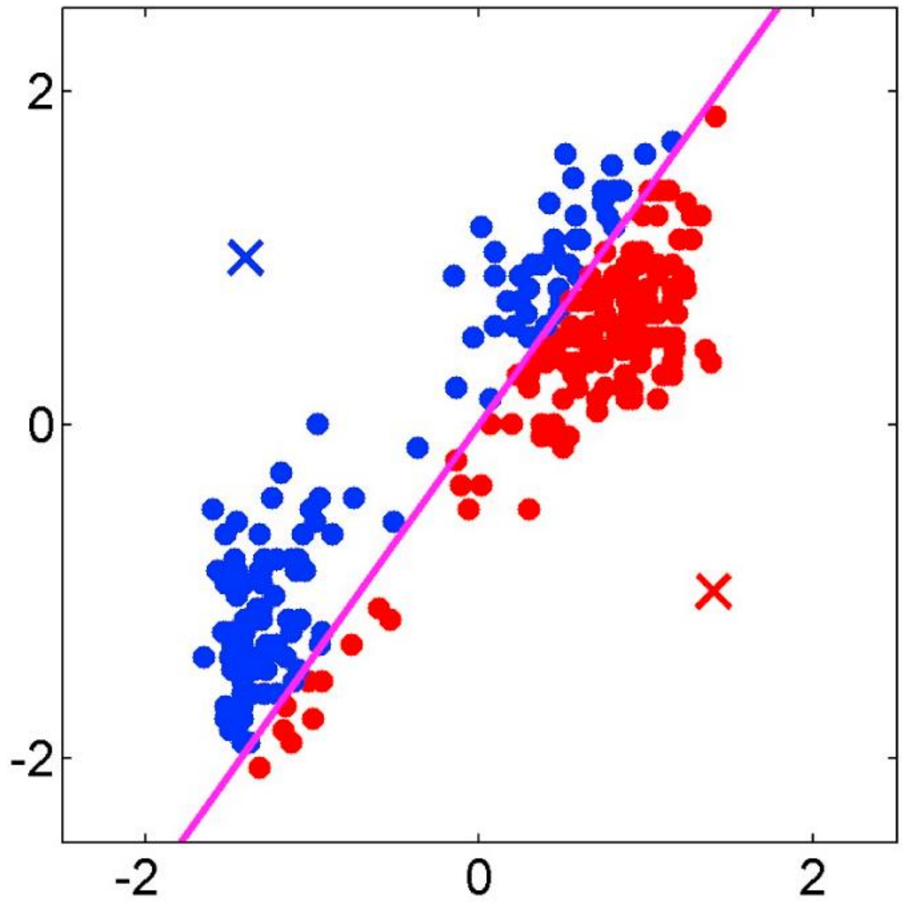
  3. 损失函数是有界的

K-Means



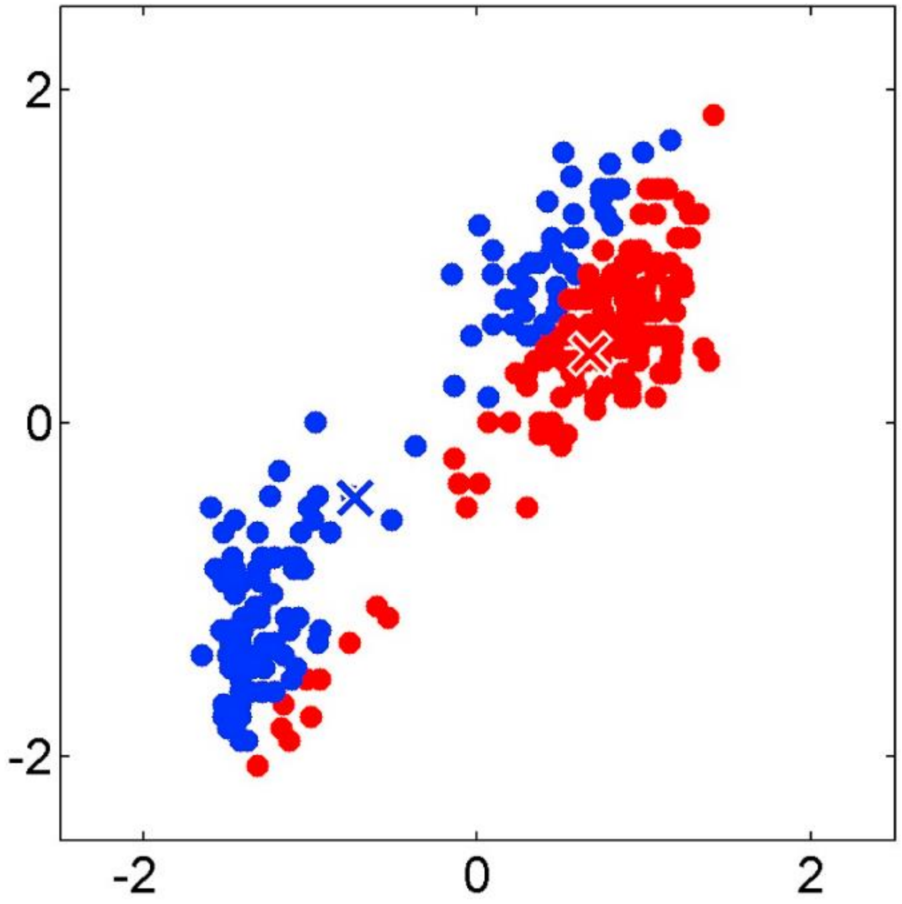
K-Means算法运行举例

K-Means



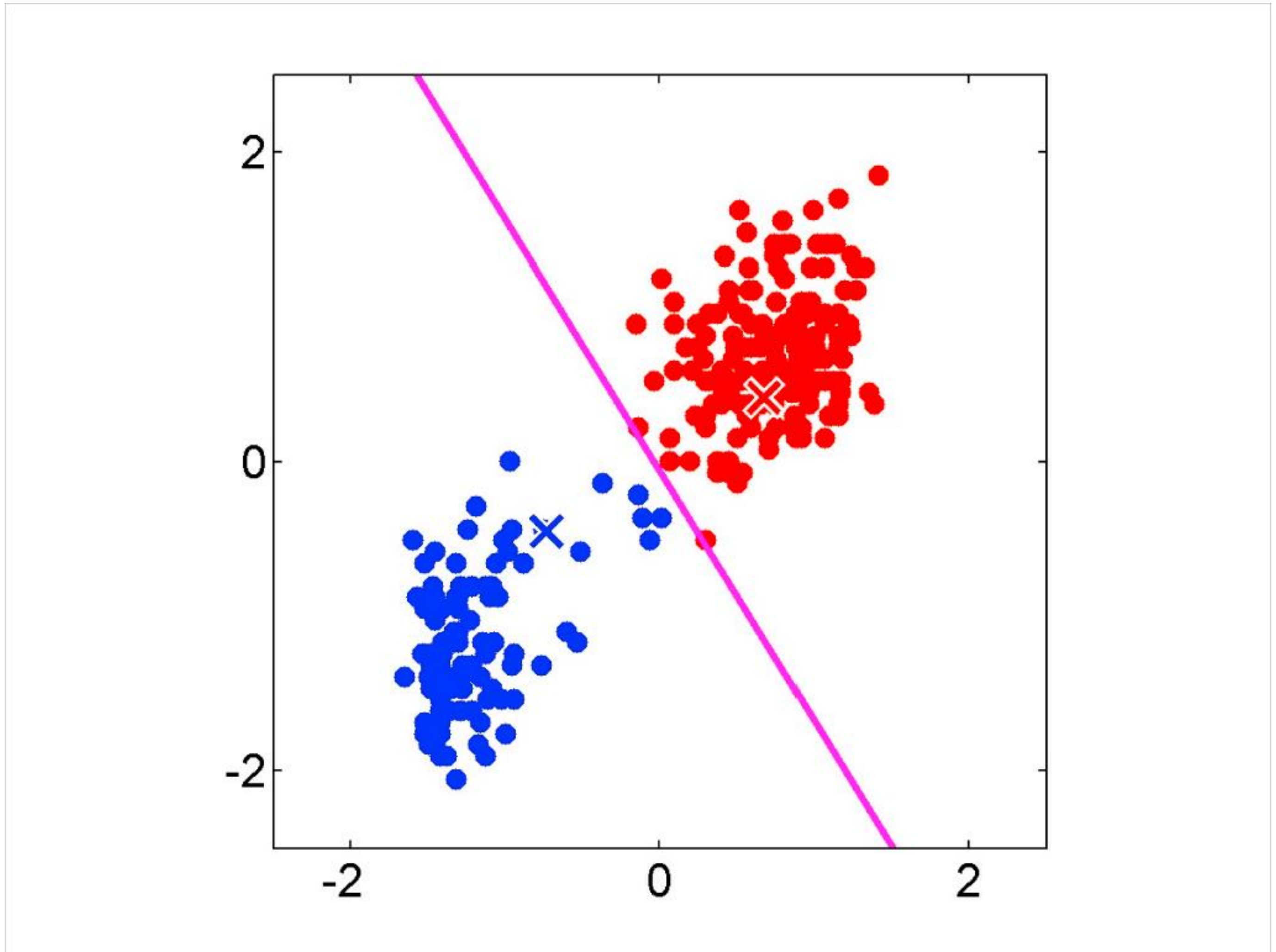
K-Means算法运行举例

K-Means



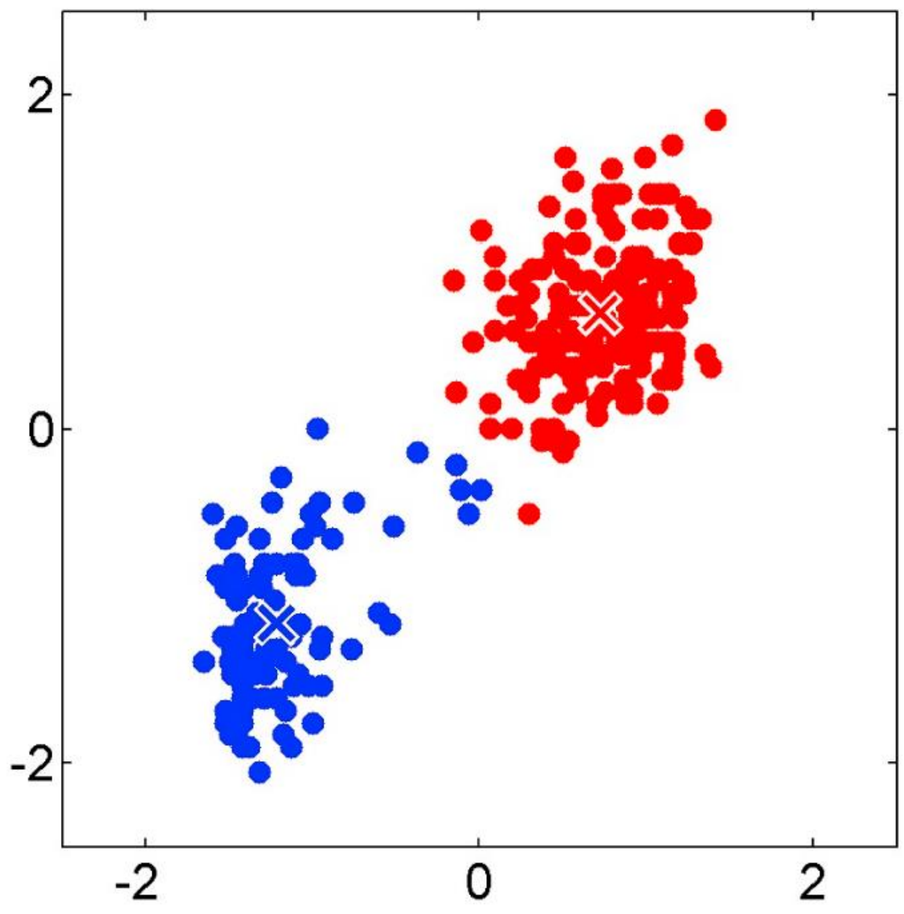
K-Means算法运行举例

K-Means



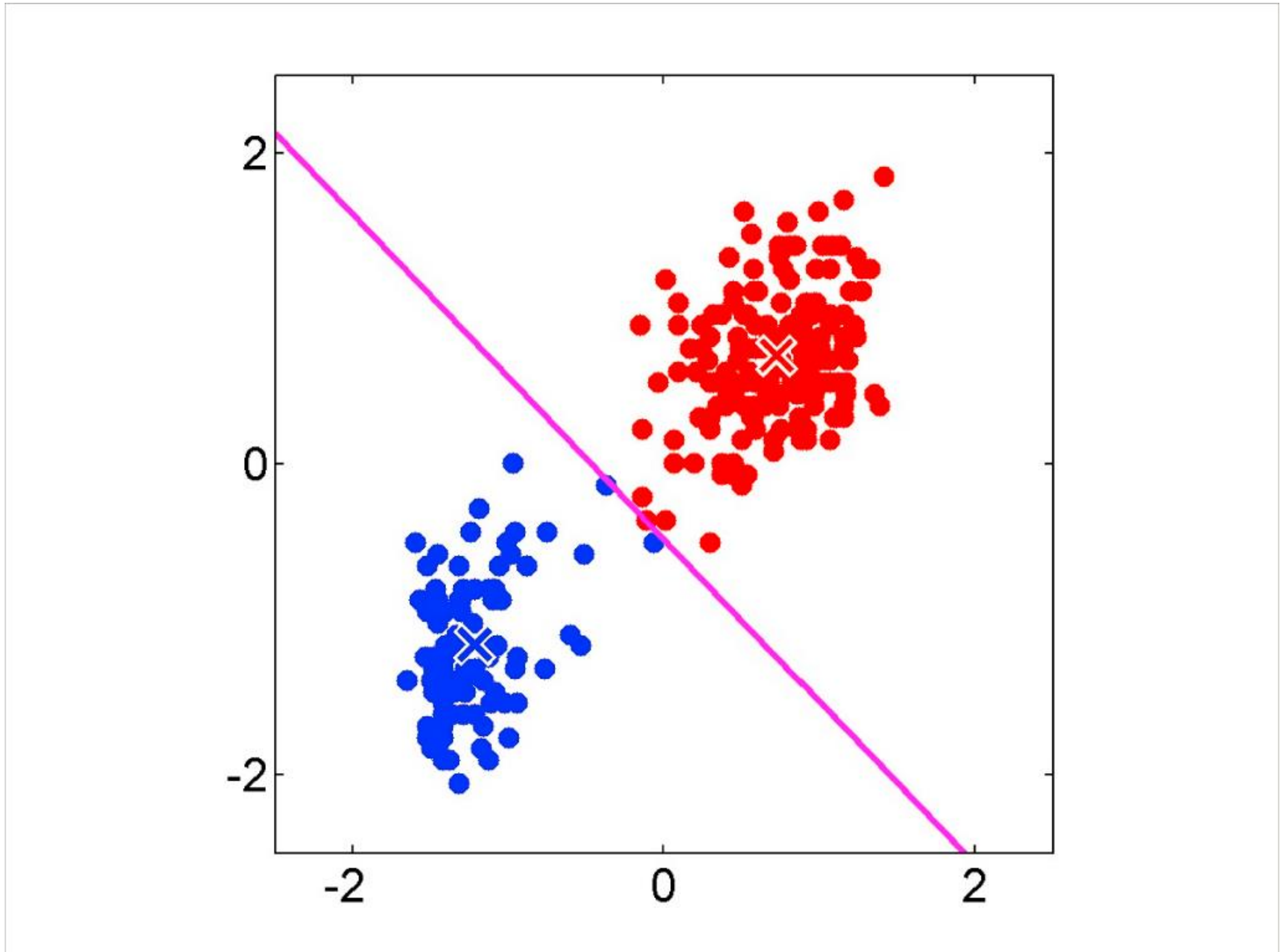
K-Means算法运行举例

K-Means



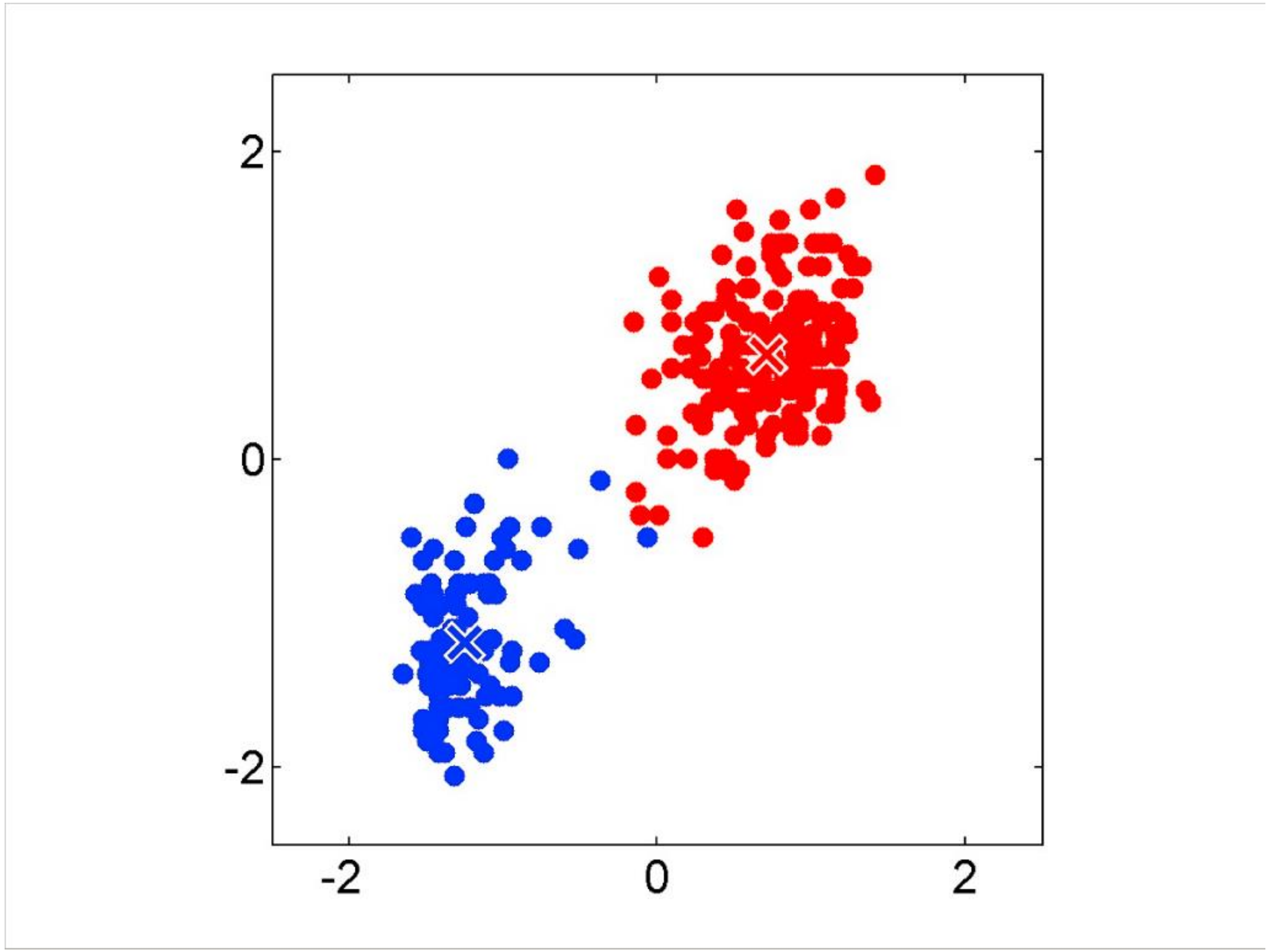
K-Means算法运行举例

K-Means



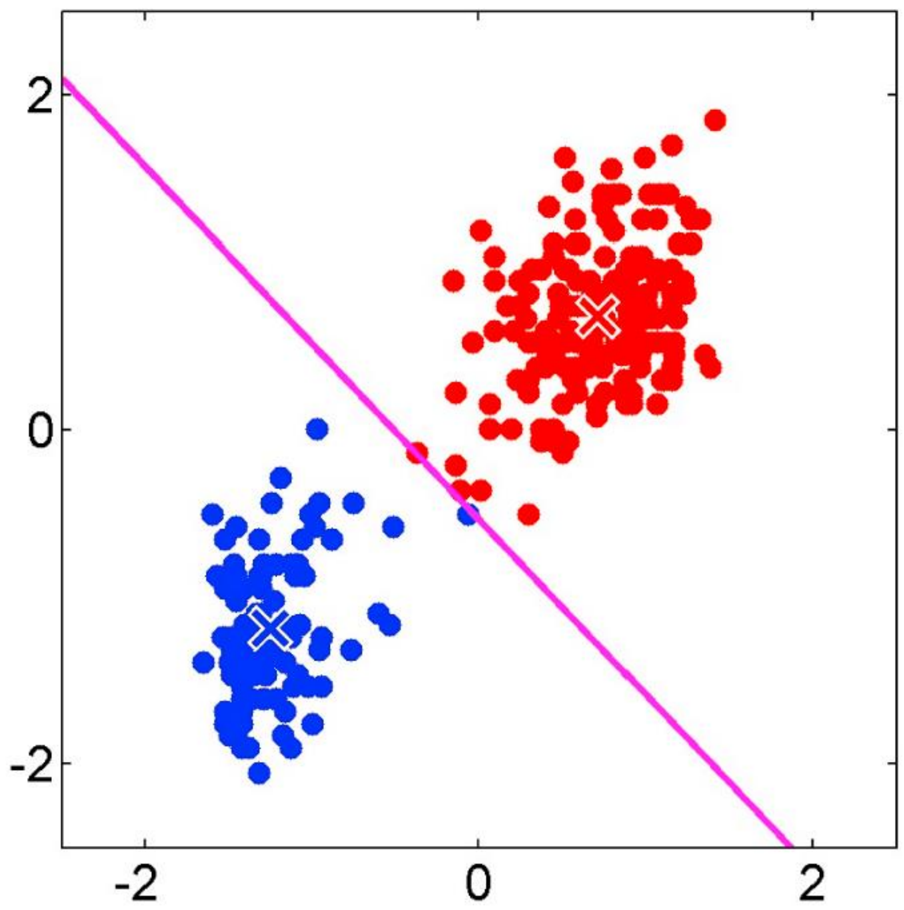
K-Means算法运行举例

K-Means



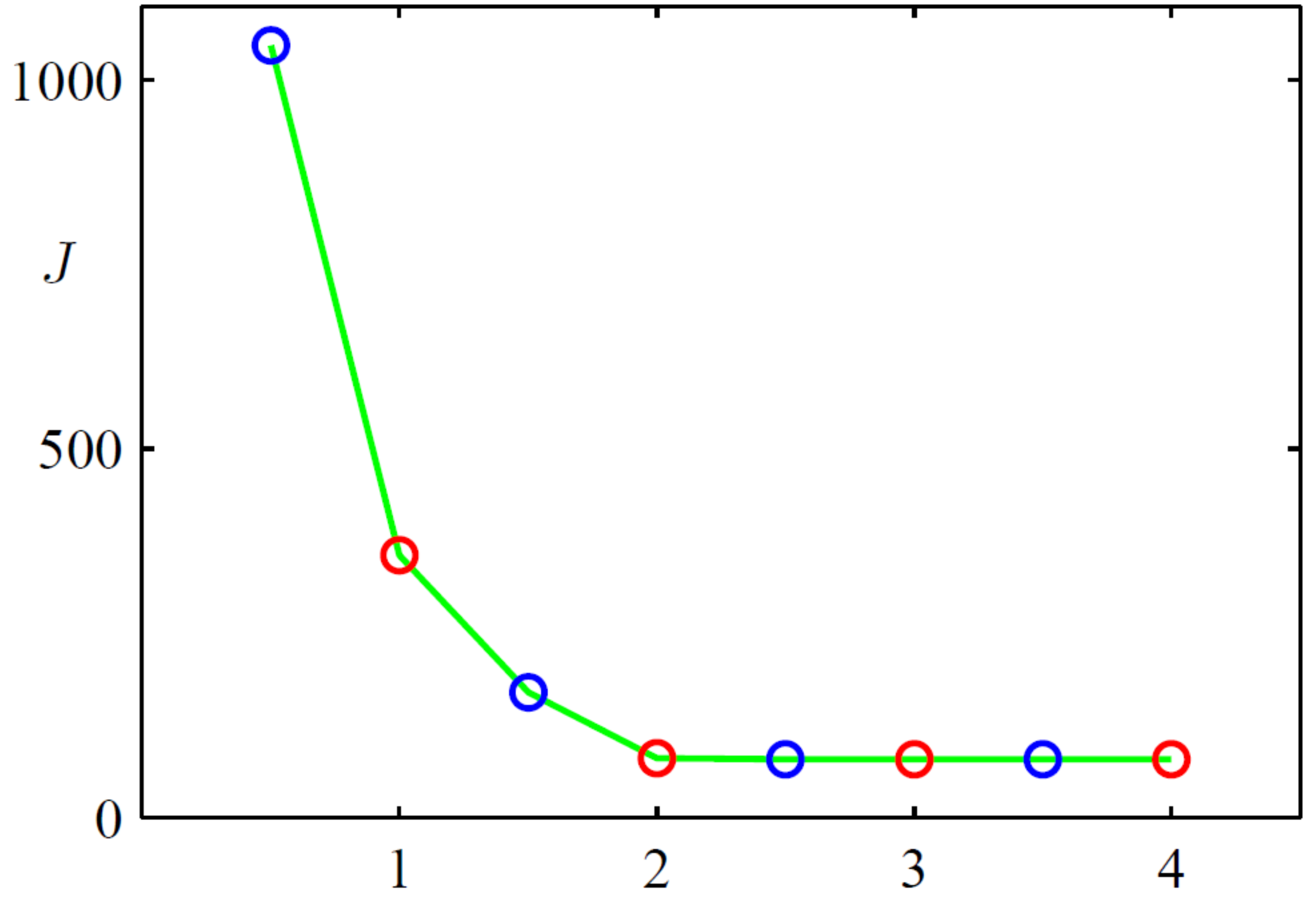
K-Means算法运行举例

K-Means



K-Means算法运行举例

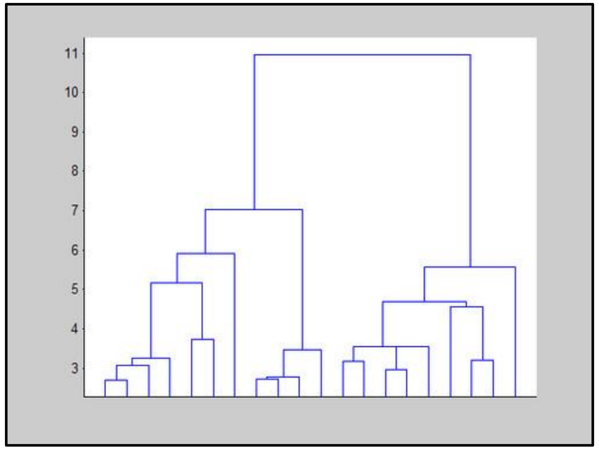
K-Means



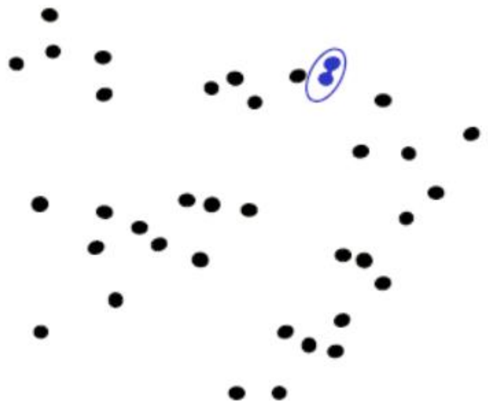
收敛曲线

K-Means

- K-means算法的缺陷
  1. Generally是非常棒的算法：简单有效，实现方便
  2. 如何选择合适的K受经验因素影像 (*Hierarchical K-means*)



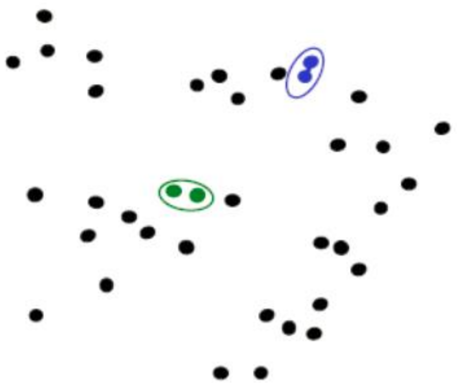
Agglomerative Clustering



1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster

Slide credit: Andrew Moore

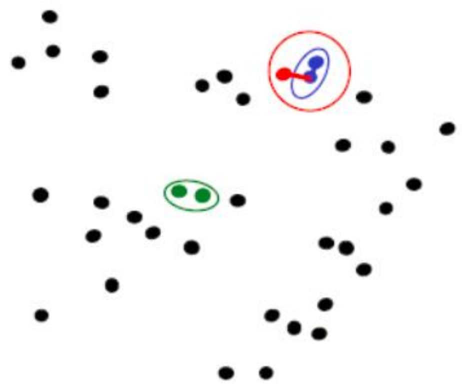
Agglomerative Clustering



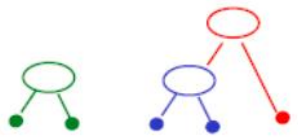
1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat

Slide credit: Andrew Moore

Agglomerative Clustering



1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat

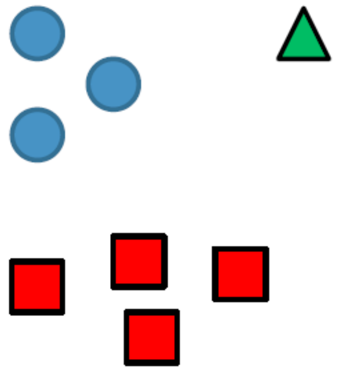


Slide credit: Andrew Moore

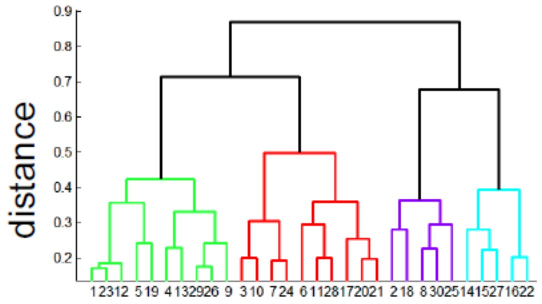
AI300学院

Agglomerative Clustering

- 如何定义类间相似性?
  1. 平均距离
  2. 最大距离
  3. 最小距离
  4. 中心距离



- 如何选择中心数量
  1. 树状结构
  2. 预定义聚类中心数量
  3. 预定义聚类中心距离



K-means聚类实战

K-means scikit-learn的python实现

【对scikit-learn中K-means概述】

K-Means算法是一个重复移动类中心点的过程，把类的中心点，也称重心（centroids），移动到其包含成员的平均位置，然后重新划分其内部成员。

参数分析：

- n_cluster：类别的个数
- max_iter：迭代的次数

属性分析：

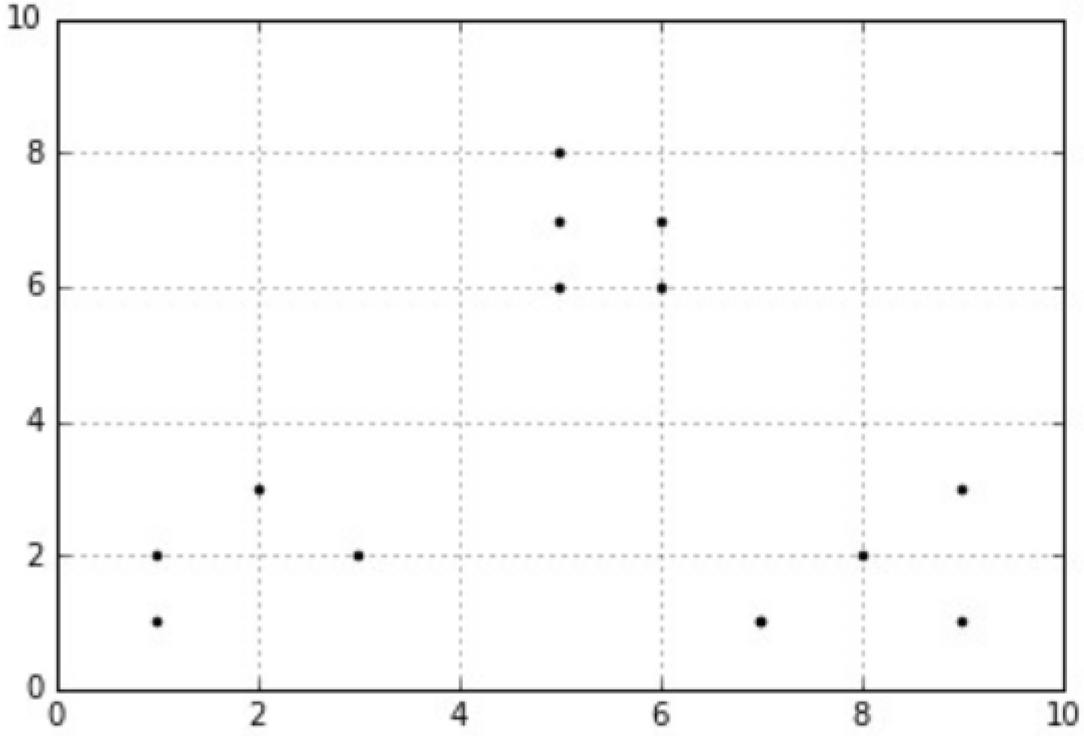
- cluster_centers_：向量，[n_clusters, n_features]，每个簇中心的坐标
- Labels_：每个点的分类
- inertia_：float，每个点到其簇的质心的距离之和

K-means聚类实战

K-means scikit-learn的python实现

【K-means 数据构成】

```
import numpy as np
from sklearn.cluster import KMeans
from sklearn import metrics
plt.figure(figsize=(8, 10))
plt.subplot(3, 2, 1)
x1 = np.array([1, 2, 3, 1, 5, 6, 5, 5, 6, 7, 8, 9, 7, 9])
x2 = np.array([1, 3, 2, 2, 8, 6, 7, 6, 7, 1, 2, 1, 1, 3])
X = np.array(list(zip(x1, x2))).reshape(len(x1), 2)
plt.xlim([0, 10])
plt.ylim([0, 10])
plt.title('样本',fontproperties=font)
plt.scatter(x1, x2)
```

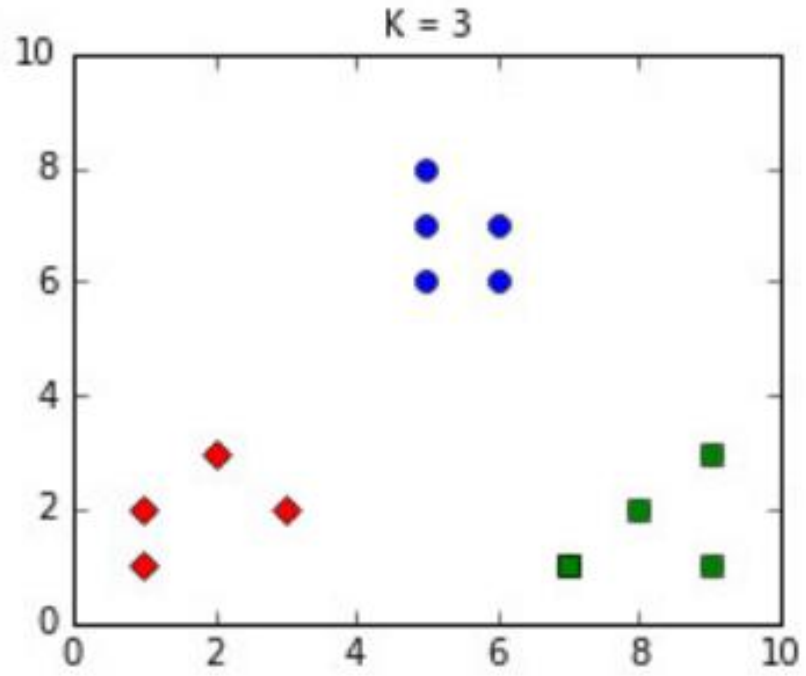


K-means聚类实战

K-means scikit-learn的python实现

【K-means 聚类（3类）】

```
plt.scatter(x1, x2)
colors = ['b', 'g', 'r']
markers = ['o', 's', 'D']
t=3
kmeans_model = KMeans(n_clusters=3).fit(X)
for i, l in enumerate(kmeans_model.labels_):
    plt.plot(x1[i], x2[i],
color=colors[l],marker=markers[l],ls='None')
    plt.xlim([0, 10])
    plt.ylim([0, 10])
    plt.title('K = %s' %(t),fontproperties = font)
```



局部最优解：K-Means的初始重心位置是随机选择的。有时，如果运气不好，随机选择的重心会导致K-Means陷入局部最优解。这些类可能没有实际意义，为了避免局部最优解，K-Means通常初始时要重复运行十几次甚至上百次。每次重复时，它会随机的从不同的位置开始初始化。最后把最小的成本函数对应的重心位置作为初始位置。

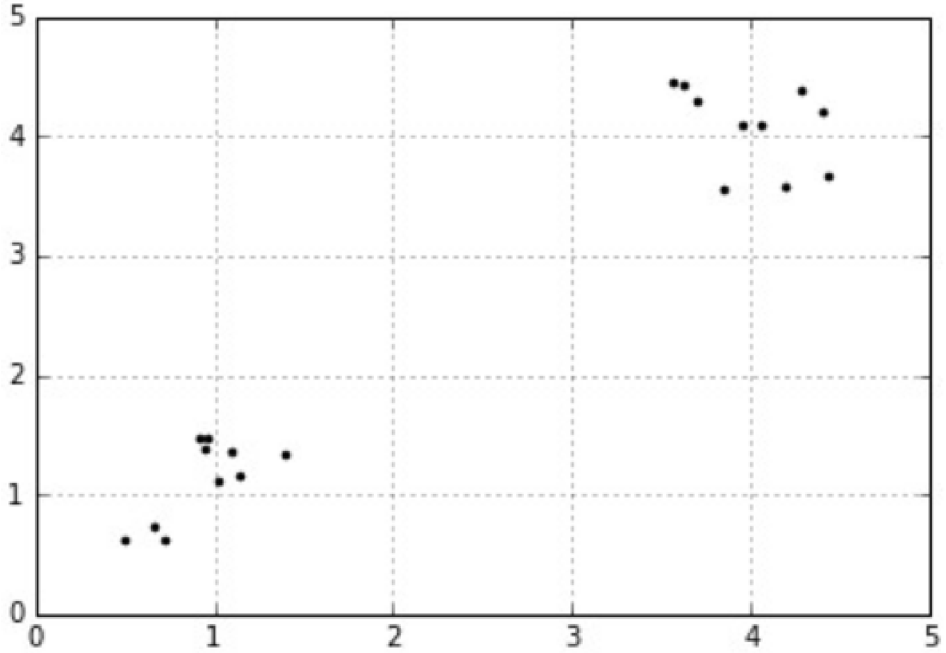
K-means聚类实战

K-means scikit-learn的python实现

【K值确定】

肘部法则：如果问题中没有指定K的值，可以通过肘部法则这一技术来估计聚类数量。肘部法则会把不同值的成本函数值画出来。随着 值的增大，平均畸变程度会减小；每个类包含的样本数会减少，于是样本离其重心会更近。但是，随着K值继续增大，平均畸变程度的改善效果会不断减低。 值增大过程中，畸变程度的改善效果下降幅度最大的位置对应的值就是肘部。

```
import numpy as np
cluster1 = np.random.uniform(0.5, 1.5, (2, 10))
cluster2 = np.random.uniform(3.5, 4.5, (2, 10))
X = np.hstack((cluster1, cluster2)).T
plt.figure()
plt.axis([0, 5, 0, 5])
plt.grid(True)
plt.plot(X[:,0],X[:,1],'k.');
```

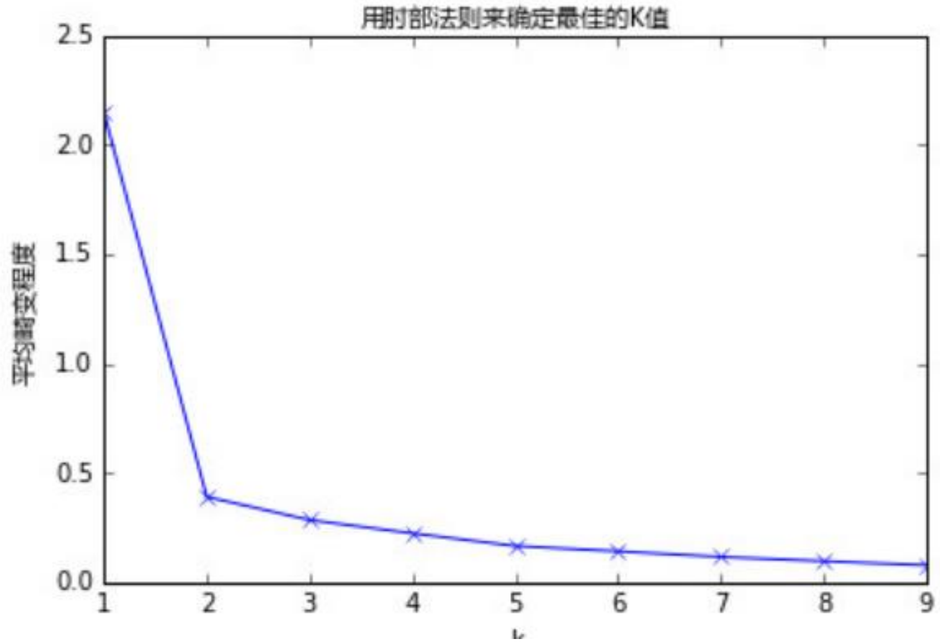


K-means聚类实战

K-means scikit-learn的python实现

【肘部法则】

```
from sklearn.cluster import KMeans
from scipy.spatial.distance import cdist
K = range(1, 10)
meandistortions = []
for k in K:
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(X)
    meandistortions.append(sum(np.min(cdist(X,
kmeans.cluster_centers_, 'euclidean'), axis=1)) / X.shape[0])
plt.plot(K, meandistortions, 'bx-')
plt.xlabel('k')
plt.ylabel('平均畸变程度',fontproperties=font)
plt.title(‘用肘部法则来确定最佳的K值’，fontproperties = font)
```



K 值从1到2时，平均畸变程度变化最大。超过2以后，平均畸变程度变化显著降低。因此肘部就是K=2 。

AI300学院