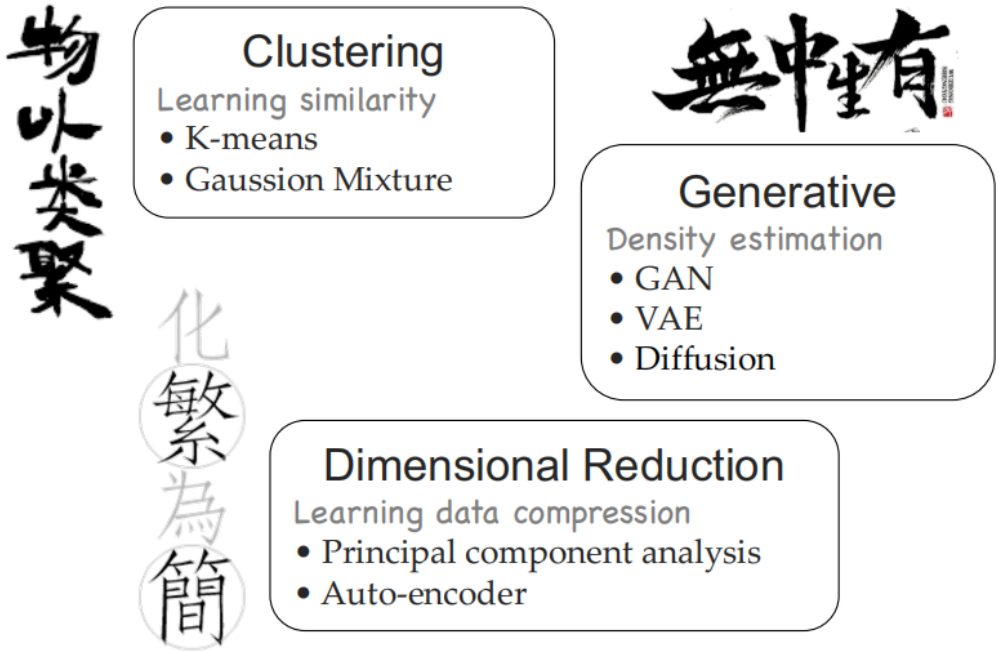
ch15: K-means (Clustering)

常见的无监督学习任务与分类（P5）



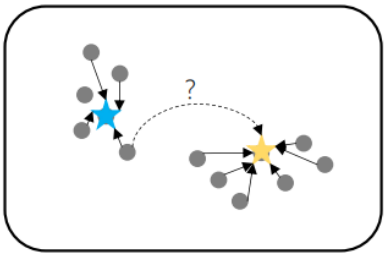
Clustering

Clustering定义（P7）

- **Clustering** = the organization of unlabeled data into similarity groups (called clusters).
- A cluster (簇) is a collection of data items which are "**similar**" between them, and "**dissimilar**" to data items in other clusters.

How to Represent Clusters（P8）

- Suppose we have figured out the clusters for a dataset (e.g., by eyes), how can we represent them?



– For each cluster, elect a "**center**" point which best represents the whole cluster.
– Each data instance is assigned with a "label" indicating its **membership** to each cluster.

数据点指向代表

"Similar" or "Dissimilar"?

相似：和本聚类中的其他物体尽量近，和其他聚类的尽可能远

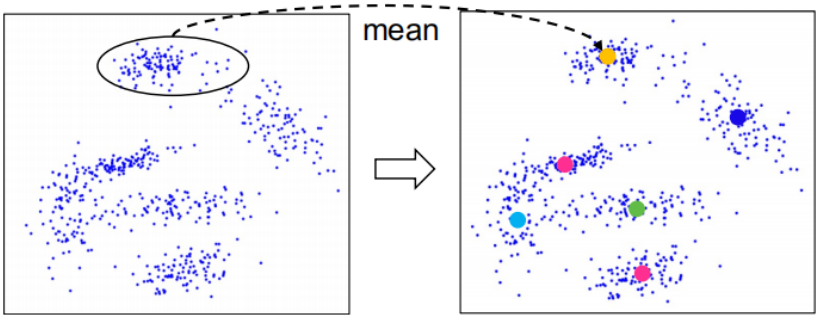
- The centroid must be more **similar** to instances in the represented cluster than other instances.
- Instances within a cluster are more **similar** to the centroid of the same cluster than centroids of other clusters.

K-means

overview（P10）

- The *k*-means algorithm partitions the given data into ***k*** clusters, where the **center** for each cluster is defined as the **mean** of all instances within the cluster, called **centroid**.
- Let $D = \{x_1,\ldots,x_N\}$ be a data set, where $x_i \in \mathbb{R}^d$ is a d dimensional vector. We want to partition D into C clusters. We assign each x_i with an indicating variable $m_i \in \{1,\ldots,C\}$.

(指示变量)



Steps（P11）

1. Choose k (random) data points (seeds) to be the initial centroids, cluster centers.
2. Assign each data point to the closest centroid.

找离自己最近的中心点

$$L_2(x, \mu^k) = \|x - \mu^k\| = \sqrt{\sum_{m=1}^{d} (x_i - \mu_m^k)^2}$$

3. Re-compute the centroids using the current cluster memberships.

$$\mu^k = \frac{1}{C_k} \sum_{x \in C_k} x$$

和算质心一样

4. Repeat steps 2 and 3 until a convergence criterion is met.

Convergence Criterion（P18）

- No (or minimum) re-assignments of data points to different clusters, *or*
- No (or minimum) change of centroids, or
- Minimum decrease in the **sum of squared error** (SSE):

$$\min_{\{\mu^k\}_{k=1}^K} \sum_{k=1}^{K} \sum_{x \in C_k} L(x - \mu^k) \qquad \mu^k = \frac{1}{C_k} \sum_{x \in C_k} x$$

优化目标（点到中心点的距离尽可能小）的体现

Time Complexity（P22）

- Assume computing distance between two instances is $O(d)$ where d is the dimensionality of the vectors.
- **Reassigning clusters**: $O(knd)$ distance computations.
- **Computing centroids**: Each instance vector gets added once to some centroid: $O(nd)$.
- Assume these two steps are each done once for I iterations: $O(Iknd)$

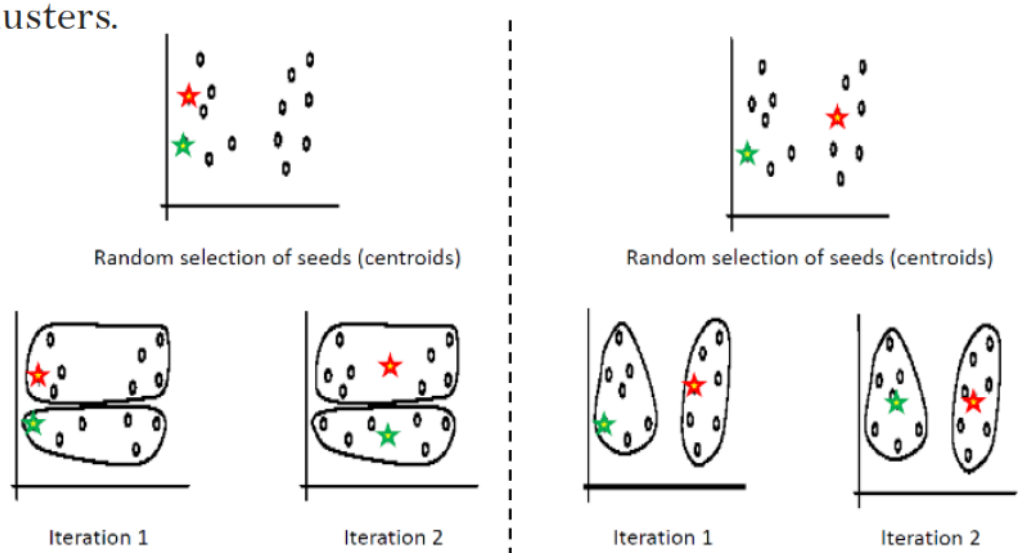
Kmeans的缺陷（P23-24）

- Finding the global optimum is NP-hard.
- The *k*-means algorithm is guaranteed to converge to a local optimum.

k-means 找不到全局最优解，只能找到局部最优解

Sensitivity to Initial Centroids (Seeds)

- Results can vary based on random seed selection.
- Some seeds can result in slow convergence or sub-optimal clusters.

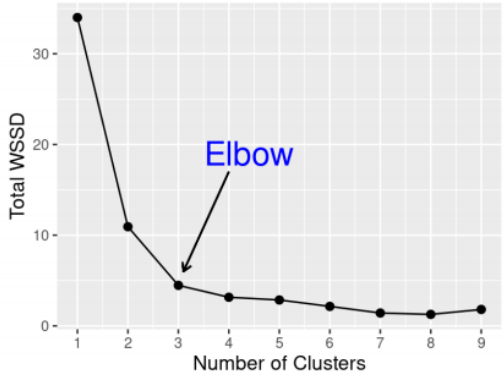


How to choose the value of K——Elbow（P25）

K的选取只能通过经验知尝试

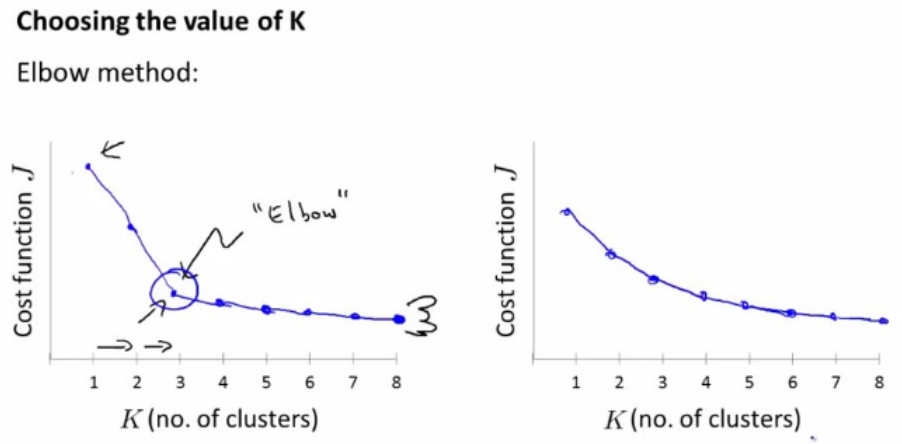
- The Elbow method is the best way to find K.
- For each K, the *within-sum-of-squares* (WSS) is defined as the sum of the squared distance between each member of the cluster and its centroid.
- K with the least amount of WSS is taken as the optimum.

$$WSS = \sum_{i=1}^{m} (x_i - c_i)^2$$



补充：

Elbow Method： Elbow意思是手肘，如下图左所示，此种方法适用于 K 值相对较小的情况，当选择的k值小于真正的时，k每增加1，cost值就会大幅的减小；当选择的k值大于真正的K时， k每增加1，cost值的变化就不会那么明显。这样，正确的k值就会在这个转折点，类似elbow的地方。 如下图：



通过画K与cost function的关系曲线图，如左图所示，肘部的值(cost function开始时下降很快，在肘部开始平缓了)做为K值，K=3。并不是所有的问题都可以通过画肘部图来解决，有的问题如右边的那个图，肘点位置不明显（肘点可以是3，4，5），这时就无法确定K值了。故肘部图是可以尝试的一种方法，但是并不是对所有的问题都能画出如左边那么好的图来确定K值。

Elbow Method公式：

$$D_k = \sum_{i=1}^{K} \sum dist(x, c_i)^2$$

补充：选取seed——Kmeans++

2.1 Kmeans++原理

$Kmeans++$在初始化簇中心时的方法总结成一句话就是：**逐个选取k个簇中心，且离其它簇中心越远的样本点越有可能被选为下一个簇中心。**其具体做法如下（其中引用英文部分论文原文）：

①从数据集$\mathcal{X}$中随机（均匀分布）选取一个样本点作为第一个初始聚类中心c_i；

> 1a. Take one center c_1, chosen uniformly at random from $\mathcal{X}$.

②接着计算每个样本与当前已有聚类中心之间的最短距离，用$D(x)$表示；然后计算每个样本点被选为下一个聚类中心的概率$P(x)$，最后选择最大概率值所对应的样本点作为下一个簇中心；

> 1b. Take a new center c_i, choosing $x \in \mathcal{X}$ with probability $P(x)$.

$$P(x) = \frac{D(x)^2}{\sum_{x \in \mathcal{X}} D(x)^2} \tag{1}$$

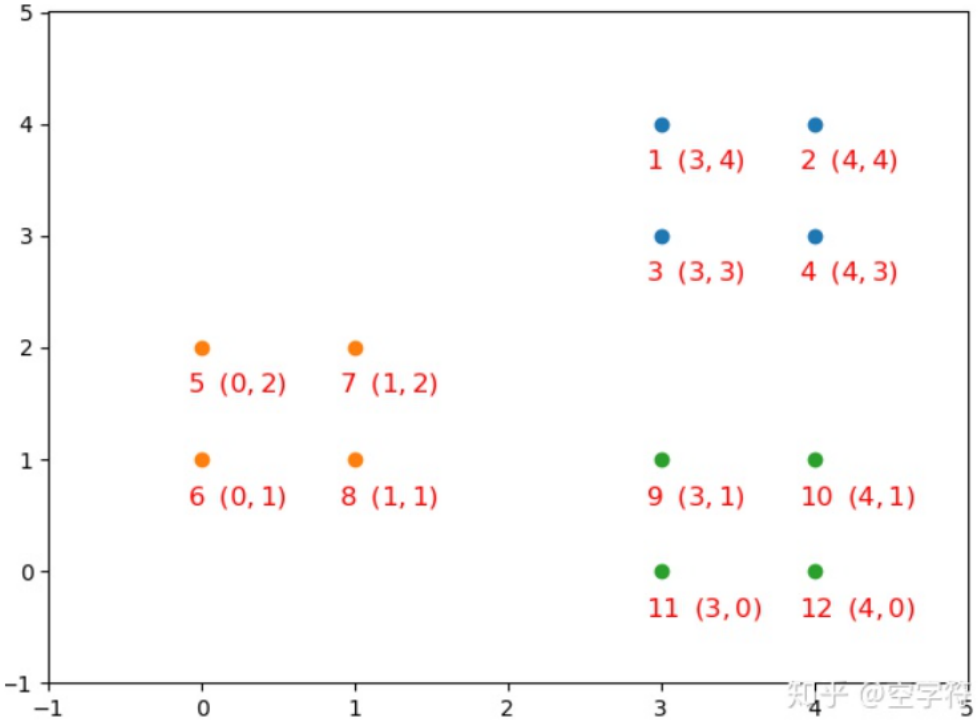
③重复第②步，直到选择出k个聚类中心；

> 1c. Repeat Step 1b. until we have taken k centers altogether.

从公式(1)也可以看成，距离现有簇中心越远的样本点，越可能被选为下一个簇中心。

2.2 计算示例

在上面的内容中，我们已经介绍了$Kmeans++$聚类算法在初始化簇中心时的具体步骤，不过仅仅只是列出公式显然不是本系列文章的风格。下面，我们就通过一个例子来实际计算一下簇中心的选择过程。



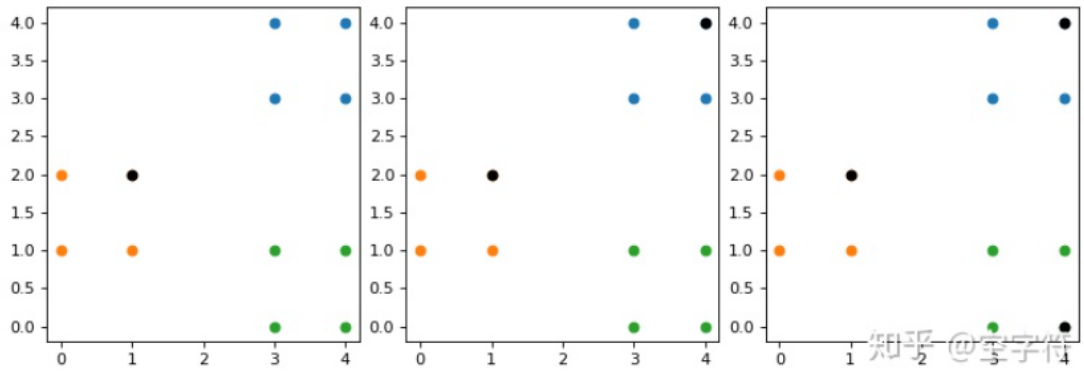
如图所示为所有的样本点，且很明显的就能看成一共包含有3个簇，这也就意味着我们需要找到3个簇中心。我们假设第一步选择的是将7号样本点$(1,2)$作为第一个初始聚类中心，那么在进行第二个簇中心的查找时，我们就需要计算所有样本点到7号样本点的距离，然后进行一个归一化。由此我们就能得到如下表格：

编号	1	2	3	4	5	6	7	8	9	10	11	12
$D(x)^2$	8	13	5	10	1	2	0	1	5	10	8	13

从表中可以看出，离7号样本点最远的是2号和12号样本点（其实从图中也可以看出），因此$Kmeans++$就会选择2号样本点为下一个聚类中心。接着，我们再次重复步骤二又能得到下面这个表格：

编号	1	2	3	4	5	6	7	8	9	10	11	12
$D(x)^2$	1	0	2	1	1	2	0	1	5	9	8	13

从表中可以看出，离2号和7号样本点最远的是12号样本点，所以下一个簇中心就会是12号样本点。进一步，我们可以得到如下可视化结果：



Application（P27-28）

- **Image Segmentation**
- **Image Compression**

Pros and Cons（P29-30）

Advantages:

- **Simple**: easy to understand and implement,
- **Efficient**: k-means is considered a linear algorithm.

Disadvantages:

- The algorithm is only applicable if the **mean** is defined.
- The user needs to specify *k*.
- The algorithm is sensitive to **outliers**.

求平均可能比较困难

比较适用于向量化的数据

假设簇为球形