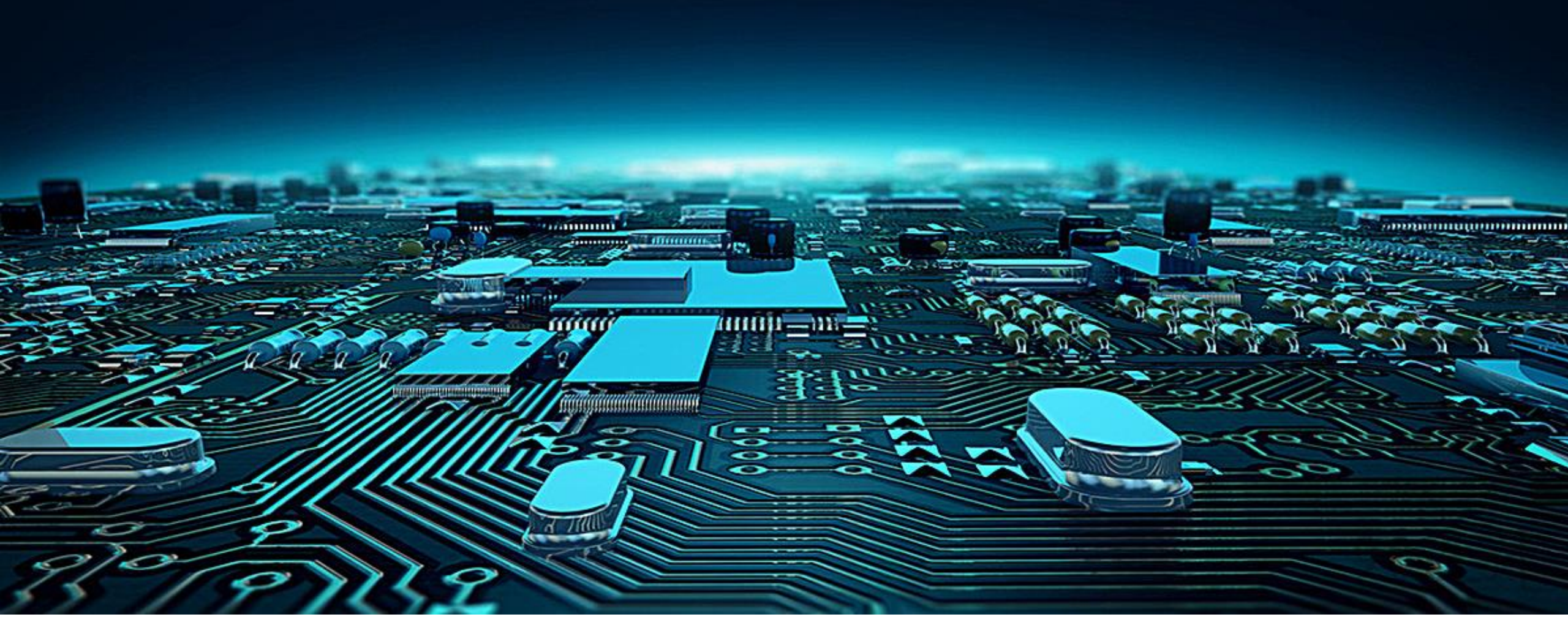
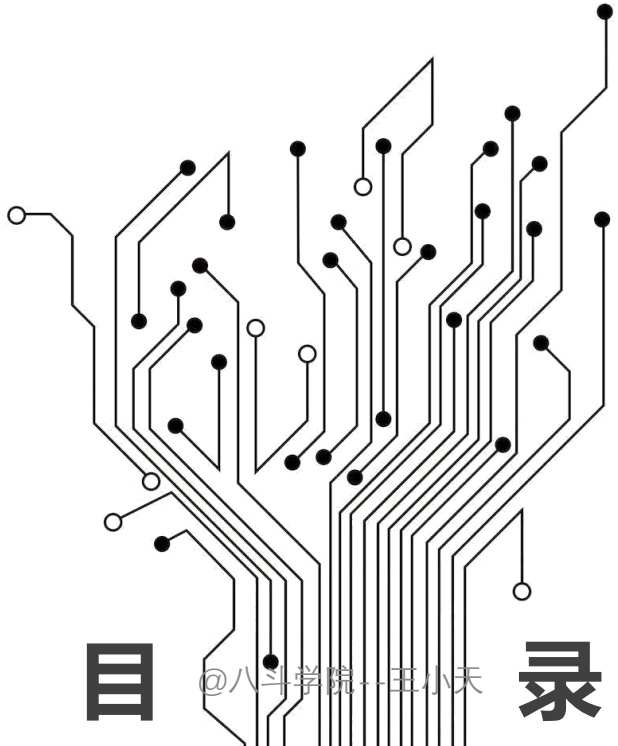


图像聚类算法

@八斗学院--王小天(Michael)

2021/12/12



目　录

分类与聚类

分类

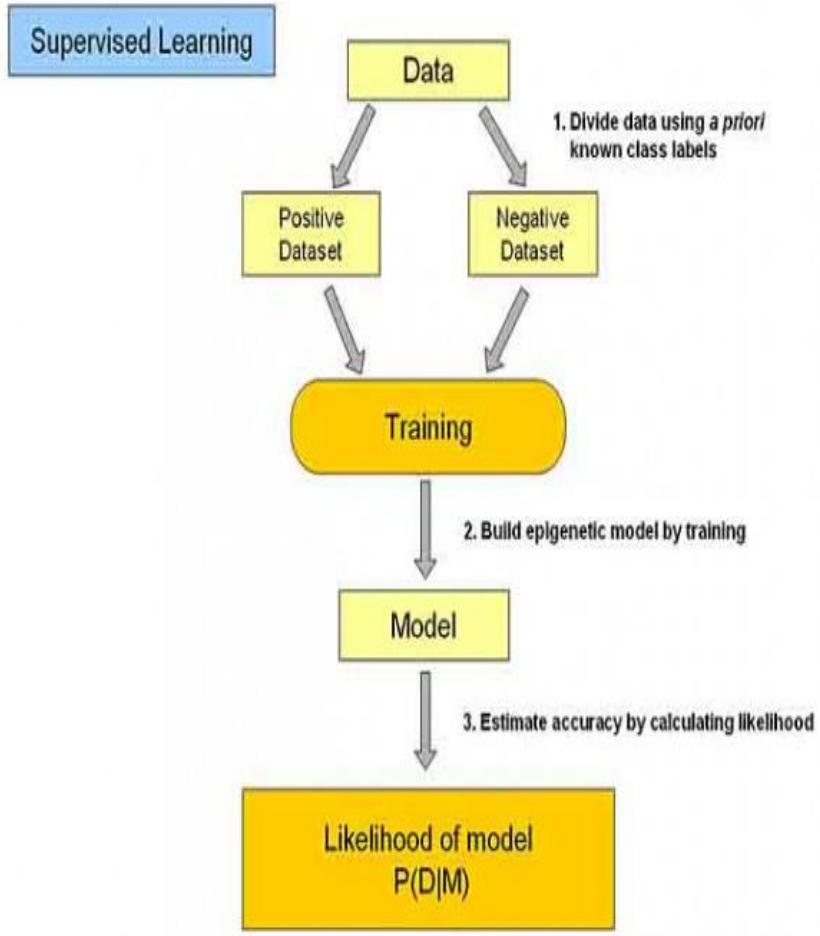
分类其实是从特定的数据中挖掘模式，作出判断的过程。

分类学习主要过程：

（1）训练数据集存在一个类标记号，判断它是正向数据集（起积极作用，不垃圾邮件），还是负向数据集（起抑制作用，垃圾邮件）；

（2）然后需要对数据集进行学习训练，并构建一个训练的模型；

（3）通过该模型对预测数据集进预测，并计算其结果的性能。



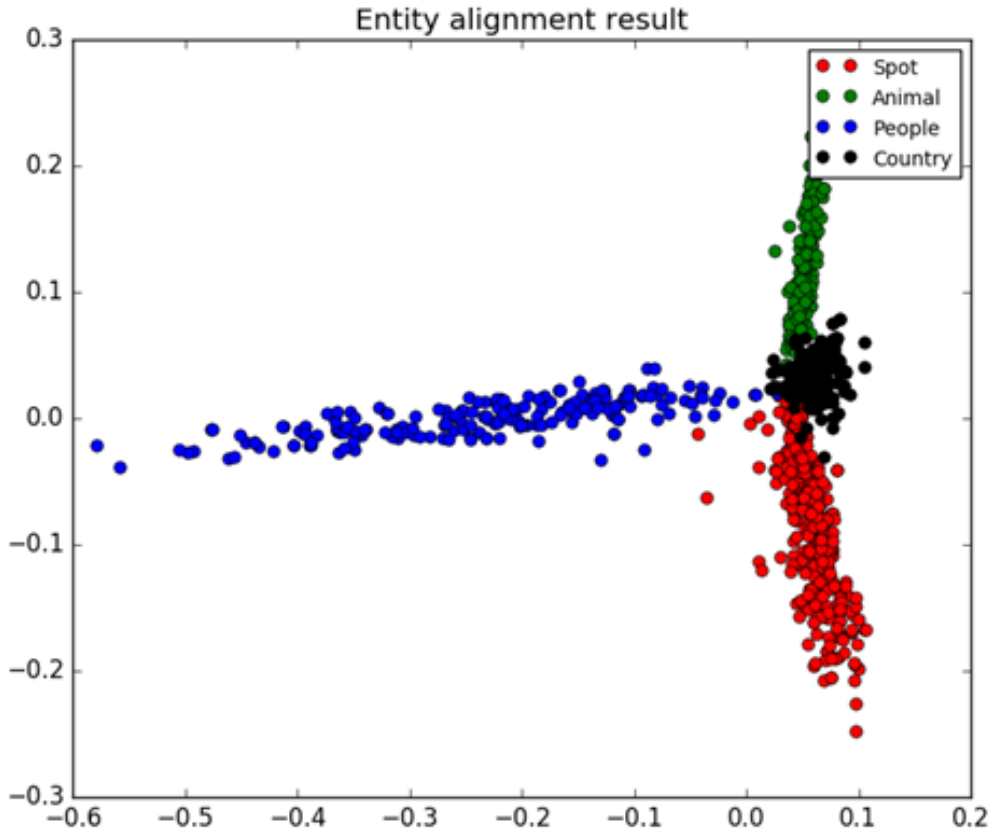
分类与聚类

聚类

从广义上说，聚类就是将数据集中在某些方面相似的数据成员放在一起。

一个聚类就是一些数据实例的集合，其中处于相同聚类中的数据元素彼此相似，但是处于不同聚类中的元素彼此不同。

由于在聚类中那些表示数据类别的分类或分组信息是没有的，**即这些数据是没有标签的**，所以聚类通常被归为无监督学习（Unsupervised Learning）。



分类与聚类

聚类的目的也是把数据分类，但是事先是不知道如何去分的，完全是算法自己来判断各条数据之间的相似性，相似的就放在一起。

在聚类的结论出来之前，完全不知道每一类有什么特点，一定要根据聚类的结果通过人的经验来分析，看看聚成的这一类大概有什么特点。

总之，聚类主要是"物以类聚"，通过相似性把相似元素聚集在一起，它没有标签；而分类通过标签来训练得到一个模型，对新数据集进行预测的过程，其数据存在标签。

聚类样本间的属性

1. 有序属性：西瓜的甜度：0.1,0.5,0.9

2. 无序属性：性别：男，女

聚类的常见算法

聚类算法分为三大类：

1. 原型聚类：
    - K均值聚类算法
2. 层次聚类
3. 密度聚类

K-Means聚类

K-Means聚类是最常用的聚类算法，最初起源于信号处理，其目标是将数据点划分为K个类簇。

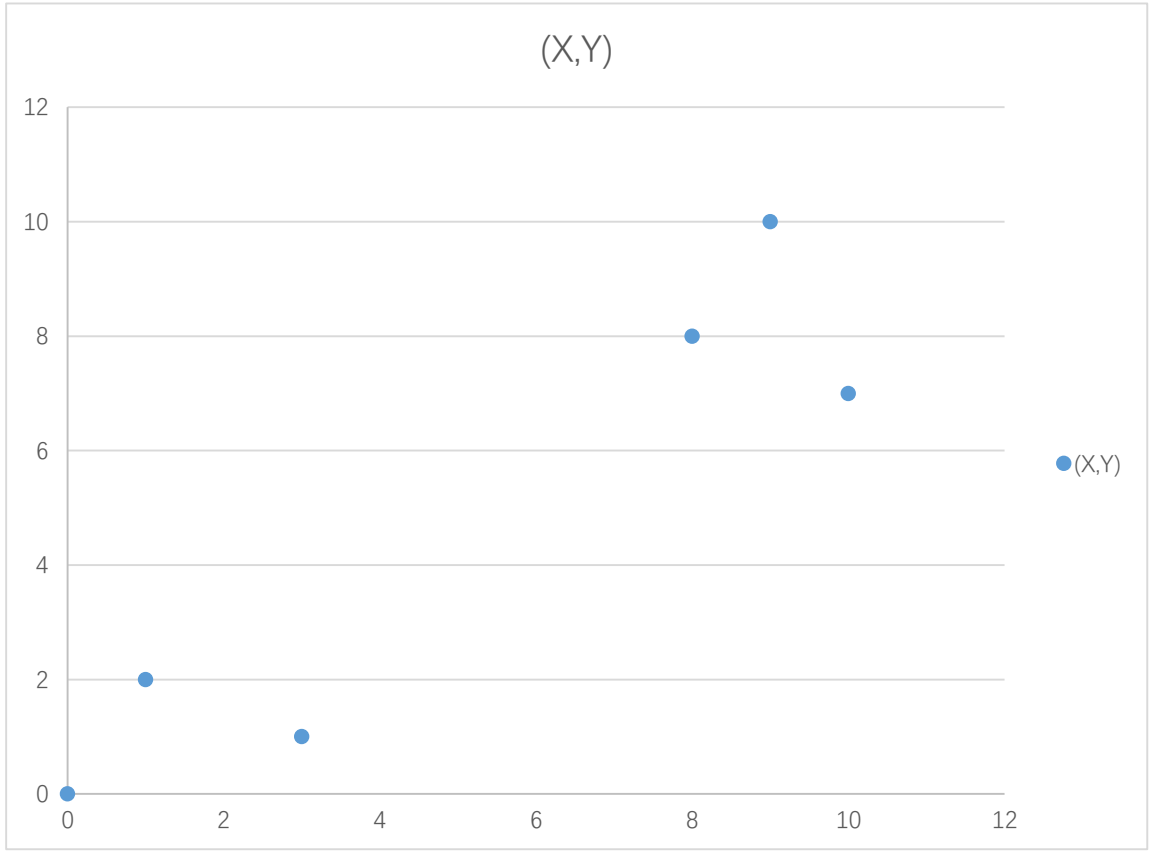
该算法的最大优点是简单、便于理解，运算速度较快，缺点是要在聚类前指定聚集的类簇数。

k-means算法是一种原型聚类算法。

k-means聚类算法的分析流程：

第一步，确定K值，即将数据集聚集成K个类簇或小组。
第二步，从数据集中随机选择K个数据点作为质心（Centroid）或数据中心。
第三步，分别计算每个点到每个质心之间的距离，并将每个点划分到离最近质心的小组。
第四步，当每个质心都聚集了一些点后，重新定义算法选出新的质心。（对于每个簇，计算其均值，即得到新的k个质心点）
第五步，迭代执行第三步到第四步，直到迭代终止条件满足为止（聚类结果不再变化）

	X	Y
P1	0	0
P2	1	2
P3	3	1
P4	8	8
P5	9	10
P6	10	7



	X	Y
P1	0	0
P2	1	2
P3	3	1
P4	8	8
P5	9	10
P6	10	7

第一步，确定K值，即将数据集聚集成K个类簇或小组。

----这里我们选K=2

第二步，从数据集中随机选择K个数据点作为质心（Centroid）或数据中心。

----假设我们选择P1和P2作为初始的质心

第三步，分别计算每个点到每个质心之间的距离，并将每个点划分到离最近质心的小组。

----计算P3到P1的距离： √10 = 3.16;

----计算P3到P2的距离： √((3-1)^2+(1-2)^2 = √5 = 2.24;

----所以P3离P2更近，P3就加入P2的簇。同理，P4、P5、P6;

	X	Y
P1	0	0
P2	1	2
P3	3	1
P4	8	8
P5	9	10
P6	10	7

	P1	P2
P3	3.16	2.24
P4	11.3	9.22
P5	13.5	11.3
P6	12.2	10.3

P3到P6都跟P2更近，所以第一次分组的结果是：

- 组A：P1
- 组B：P2、P3、P4、P5、P6

	X	Y
P1	0	0
P2	1	2
P3	3	1
P4	8	8
P5	9	10
P6	10	7

第四步，当每个质心都聚集了一些点后，重新定义算法选出新的质心。（对于每个簇，计算其均值，即得到新的k个质心点）

----组A没啥可选的，还是P1自己

----组B有五个点，需要选新质心。这里要注意选择的方法是每个组X坐标的平均值和Y坐标的平均值组成的新的点，为新质心，也就是说这个质心是“虚拟的”。

----因此，B组选出新质心的坐标为：P哥（（1+3+8+9+10）/5，（2+1+8+10+7）/5）=（6.2，5.6）。

----综合两组，新质心为P1（0，0），P哥（6.2，5.6）。

----而P2-P6重新成为离散点。

	X	Y
P1	0	0
P2	1	2
P3	3	1
P4	8	8
P5	9	10
P6	10	7

	P1	P哥
P2	2.24	6.3246
P3	3.16	5.6036
P4	11.3	3
P5	13.5	5.2154
P6	12.2	4.0497

再次计算点到质心的距离：
这时可以看到P2、P3离P1更近，P4、P5、P6离P哥更近。

第二次分组的结果是：

- 组A：P1、P2、P3
- 组B：P4、P5、P6（虚拟质心这时候消失）

	X	Y
P1	0	0
P2	1	2
P3	3	1
P4	8	8
P5	9	10
P6	10	7

	P哥1	P哥2
P1	1.4	12
P2	0.6	10
P3	1.4	9.5
P4	47	1.1
P5	70	1.7
P6	56	1.7

按照上一次的方法选出两个新的虚拟质心：

---P哥1（1.33，1），P哥2（9，8.33）。

第三次计算点到质心的距离：

--- 这时可以看到P1、P2、P3离P哥1更近，P4、P5、P6离P哥2更近。

--- 所以第三次分组的结果是：

- 组A：P1、P2、P3
- 组B：P4、P5、P6

我们发现，这次分组的结果和上次没有任何变化了，说明已经收敛，聚类结束。

在图像处理中，通过K-Means聚类算法可以实现图像分割、图像聚类、图像识别等操作。

我们通过K-Means可以将这些像素点聚类成K个簇，然后使用每个簇内的质心点来替换簇内所有的像素点，这样就能实现**在不改变分辨率的情况下**量化压缩图像颜色，实现图像颜色层级分割。

优点:

1.是解决聚类问题的一种经典算法，简单、快速

2.对处理大数据集，该算法保持高效率

3.当结果簇是密集的，它的效果较好

缺点:

1.必须事先给出k（要生成的簇的数目）。

2.对躁声和孤立点数据敏感

层次聚类

层次聚类是一种很直观的算法。顾名思义就是要一层一层地进行聚类。

层次法（Hierarchicalmethods）先计算样本之间的距离。每次将距离最近的点合并到同一个类。然后，再计算类与类之间的距离，将距离最近的类合并为一个大类。不停的合并，直到合成了一个类。其中类与类的距离的计算方法有：最短距离法，最长距离法，中间距离法，类平均法等。比如最短距离法，将类与类的距离定义为类与类之间样本的最短距离。

层次聚类算法根据层次分解的顺序分为：自下底向上和自上向下，即**凝聚的层次聚类算法**和**分裂的层次聚类算法（agglomerative和divisive）**，也可以理解为自下而上法（bottom-up）和自上而下法（top-down）。

凝聚层次聚类的流程

凝聚型层次聚类的策略是先将每个对象作为一个簇，然后合并这些原子簇为越来越大的簇，直到所有对象都在一个簇中，或者某个终结条件被满足。绝大多数层次聚类属于凝聚型层次聚类，它们只是在簇间相似度的定义上有所不同。 这里给出采用最小距离的凝聚层次聚类算法流程:

(1) 将每个对象看作一类，计算两两之间的最小距离;

(2) 将距离最小的两个类合并成一个新类;

(3) 重新计算新类与所有类之间的距离;

(4) 重复(2)、(3)，直到所有类最后合并成一类。

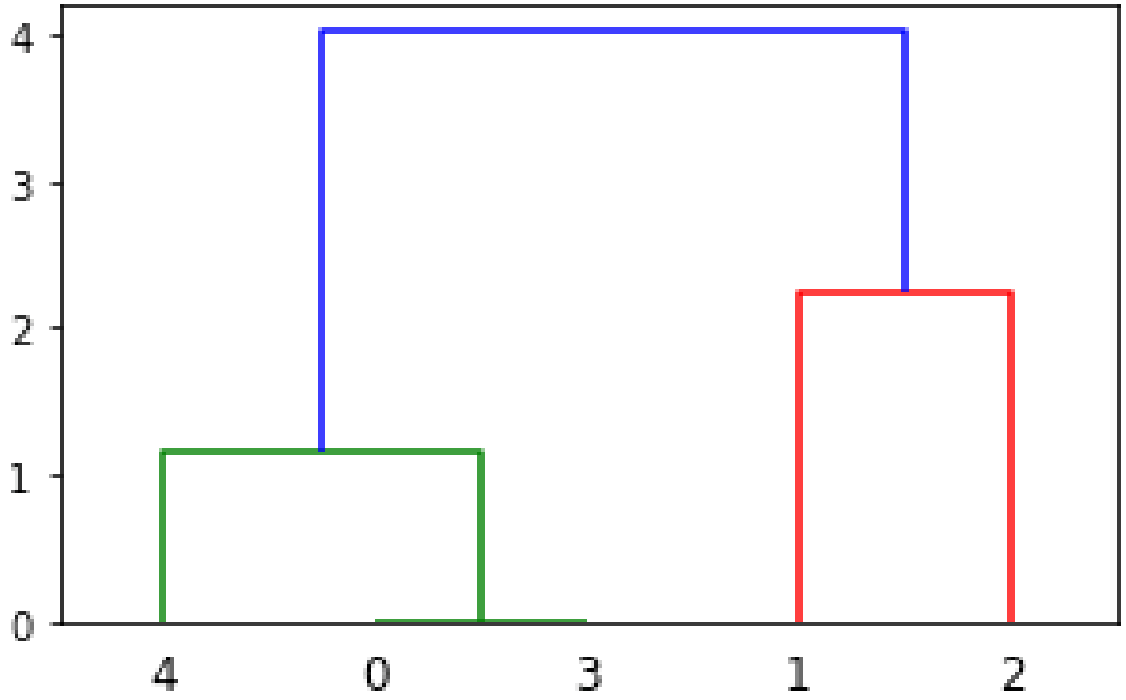
特点:

- 凝聚的层次聚类并没有类似K均值的全局目标函数，没有局部极小问题或是很难选择初始点的问题。
- 合并的操作往往是最终的，一旦合并两个簇之后就不会撤销。
- 当然其计算存储的代价是昂贵的。

层次聚类的优缺点

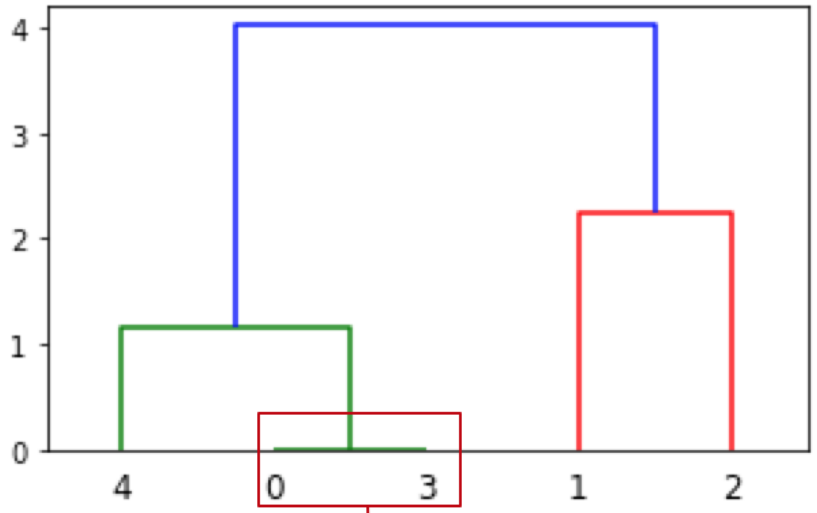
优点：1，距离和规则的相似度容易定义，限制少；
2，不需要预先制定聚类数；
3，可以发现类的层次关系；
4，可以聚类成其它形状

缺点：1，计算复杂度太高；
2，奇异值也能产生很大影响；
3，算法很可能聚类成链状




```
[[0.          3.          0.          2.        ]
 [4.          5.          1.15470054  3.        ]
 [1.          2.          2.23606798  2.        ]
 [6.          7.          4.00832467  5.        ]]
```

层次聚类

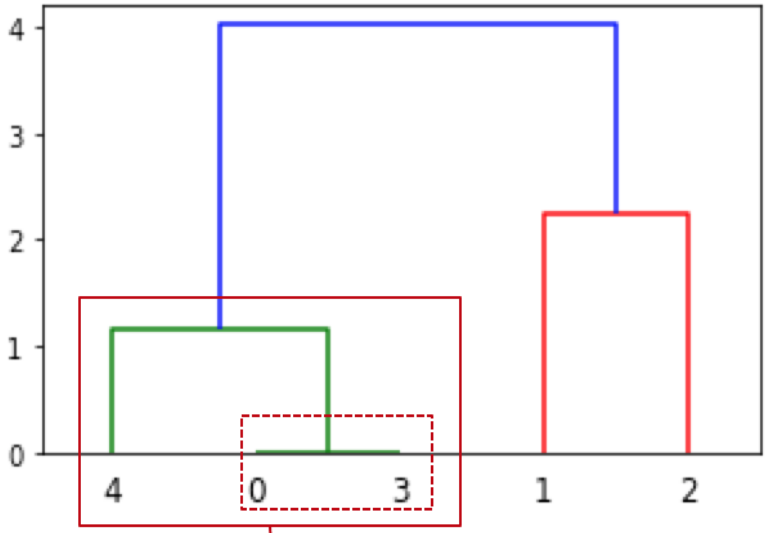



```
array([[0.        , 3.        , 0.        , 2.        ],
       [4.        , 5.        , 1.15470054, 3.        ],
       [1.        , 2.        , 2.23606798, 2.        ],
       [6.        , 7.        , 4.00832467, 5.        ]])
```

0、3两类聚为一类，为类别5

Z的第一行：[0, 3]意思是类别0和类别3距离最近，首先聚成一类，并自动定义类别为5(=len(X)-1+1)

层次聚类

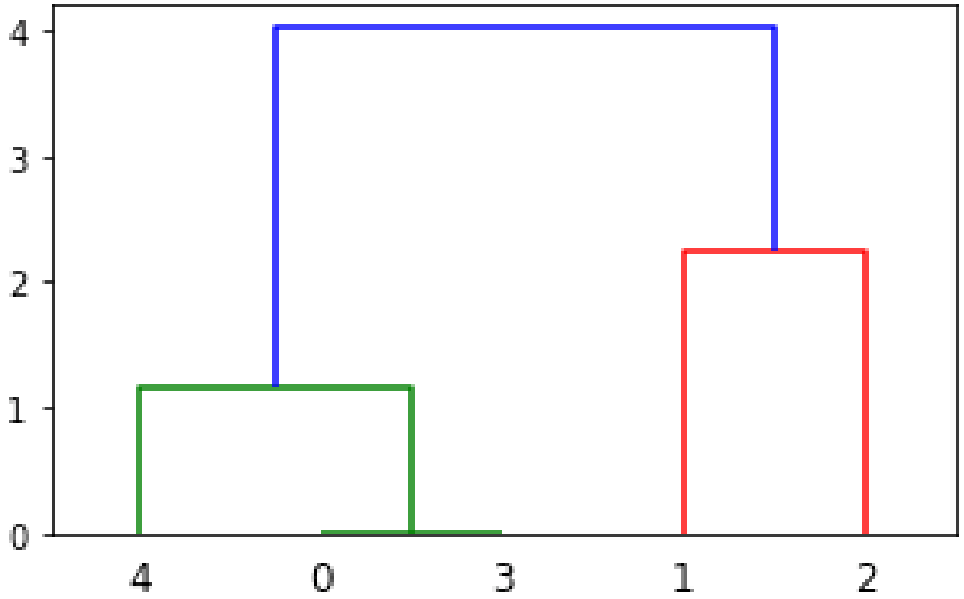



```
array([[0.        , 3.        , 0.        , 2.        ],
       [4.        , 5.        , 1.15470054, 3.        ],
       [1.        , 2.        , 2.23606798, 2.        ],
       [6.        , 7.        , 4.00832467, 5.        ]])
```

类别4与(第一次类别0和类别3聚成的)类别5进行聚类，生成类别6

Z的第二行：[4, 5]意思是类别4和上面聚类的新类别5距离为第二近，4、5聚成一类，类别为6(=len(X)-1+2)

层次聚类




```
[[0.         3.         0.         2.        ]
 [4.         5.         1.15470054 3.        ]
 [1.         2.         2.23606798 2.        ]
 [6.         7.         4.00832467 5.        ]]
```

第三行、第四行以此类推，

因为类别5有两个样本，加上类别4形成类别6，有3个样本；

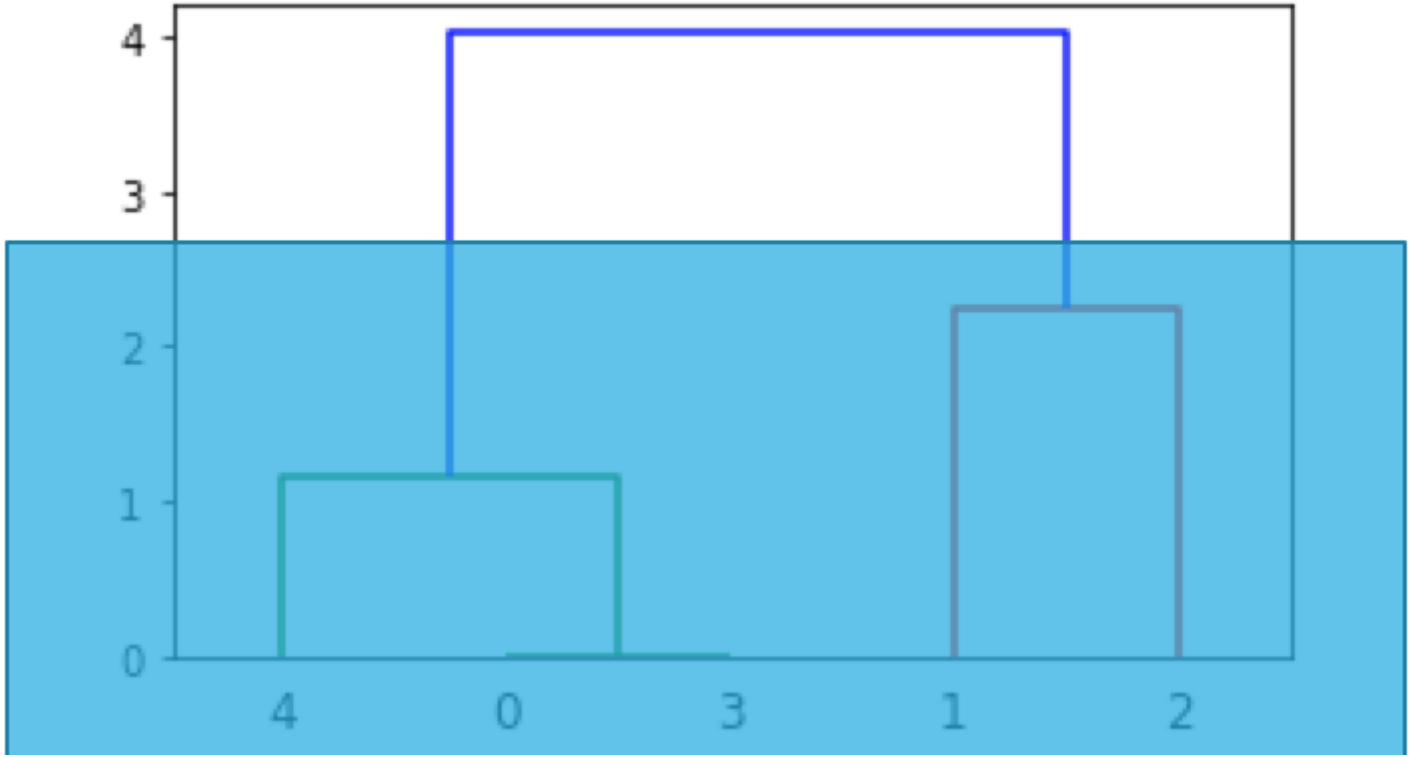
类别7是类别1、2聚类形成，有两个样本；

类别6、7聚成一类后，类别8有5个样本，这样X全部样本参与聚类，聚类完成。

Z第四列中有样本的个数，当最下面一行中的样本数达到样本总数时，聚类就完成了。

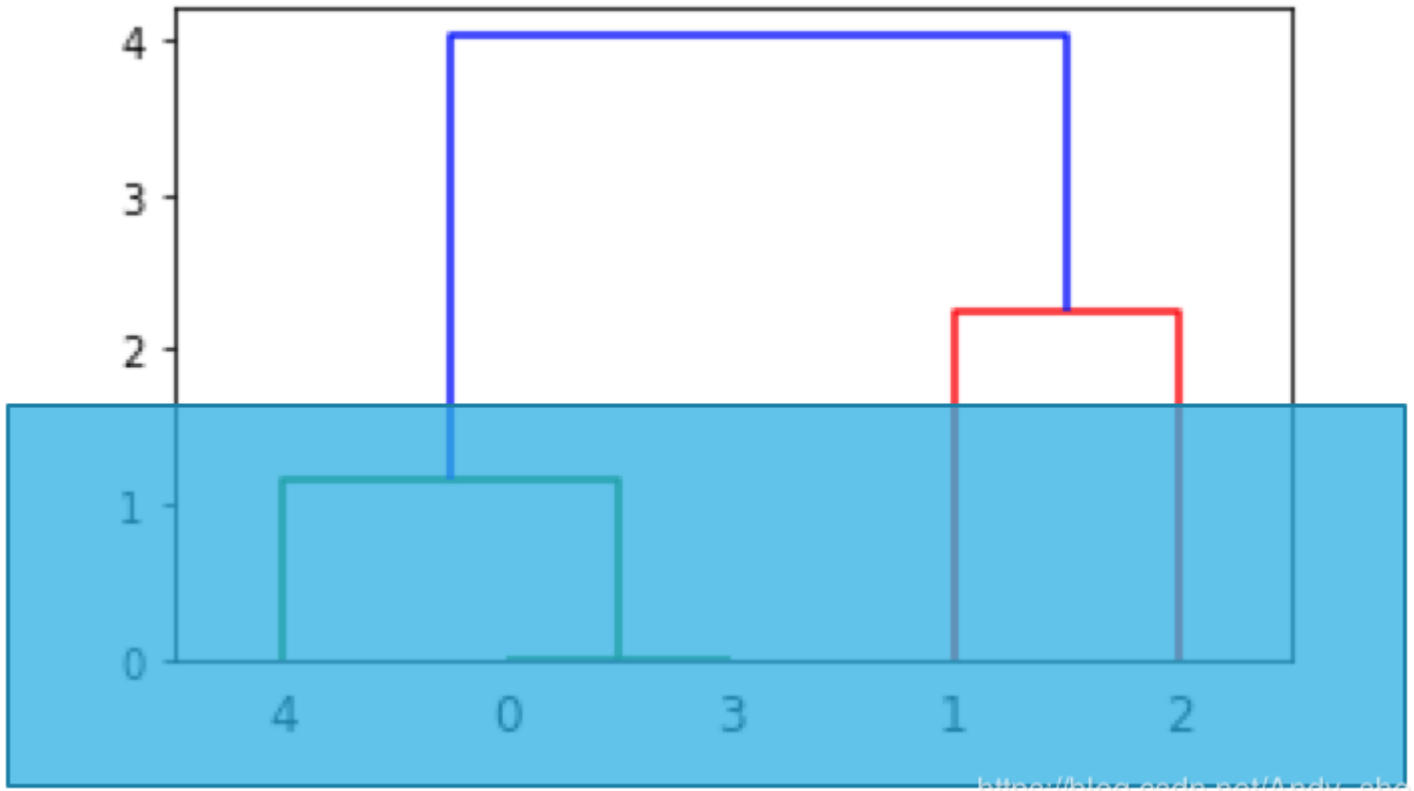
树状图分类判断

想分两类时，就从上往下数有两根竖线时进行切割，那么所对应的竖线下面所连接的为一类



树状图分类判断

想分三类时，就从上往下数有三根竖线时进行切割，那么所对应的竖线下面所连接的为一类



每一种聚类方法都有其特定的数据结构，对于服从高斯分布的数据用K-Means来进行聚类效果会比较好。

而对于类别之间存在层结构的数据，用层次聚类会比较好。

密度聚类DBSCAN

算法:

需要两个参数：ε (eps) 和形成高密度区域所需要的最少点数 (minPts)

- 它由一个任意未被访问的点开始，然后探索这个点的 ε-邻域，如果 ε-邻域里有足够的点，则建立一个新的聚类，否则这个点被标签为杂音。
- 注意，这个杂音点之后可能被发现在其它点的 ε-邻域里，而该 ε-邻域可能有足够的点，届时这个点会被加入该聚类中。

优点：

1. 对噪声不敏感；
2. 能发现任意形状的聚类。

缺点：

1. 但是聚类的结果与参数有很大的关系；
2. 用固定参数识别聚类，但当聚类的稀疏程度不同时，相同的判定标准可能会破坏聚类的自然结构，即较稀的聚类会被划分为多个类或密度较大且离得较近的类会被合并成一个聚类。

扩展--谱聚类

1. 根据数据构造一个 图结构（Graph），Graph 的每一个节点对应一个数据点，将相似的点连接起来，并且边的权重用于表示数据之间的相似度。把这个 Graph 用邻接矩阵的形式表示出来，记为 W 。
2. 把 W 的每一列元素加起来得到 N 个数，把它们放在对角线上（其他地方都是零），组成一个 N * N的矩阵，记为 D 。并令 L = D-W 。
3. 求出 L 的前 k 个特征值，以及对应的特征向量。
4. 把这 k 个特征（列）向量排列在一起组成一个 N * k 的矩阵，将其中每一行看作 k 维空间中的一个向量，并使用 K-means 算法进行聚类。聚类的结果中每一行所属的类别就是原来 Graph 中的节点亦即最初的N 个数据点分别所属的类别。

扩展---谱聚类

简单抽象谱聚类过程，主要有两步：

1. 构图，将采样点数据构造成一张网图。
2. 切图，即将第一步构造出来的按照一定的切边准则，切分成不同的图，而不同的子图，即我们对应的聚类结果。