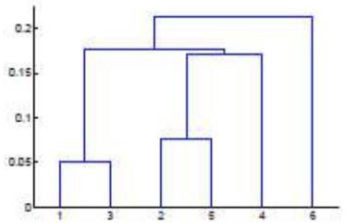
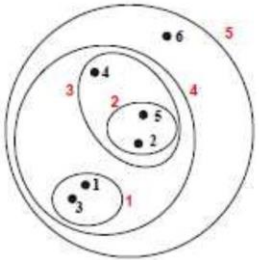
4.7 层次聚类 (Hierarchical Clustering)

CSDN学院

2017年11月





常用聚类算法

- 基于距离、相似度的聚类算法
  - *K*-means（*K*均值）及其变种（ *K*-centers 、 Mini Batch *K*-Means）
  - Mean shift
  - 吸引力传播（Affinity Propagation ，AP）
  - 层次聚类
    - 聚合聚类（Agglomerative Clustering）
- 基于密度的聚类算法
  - DBSCAN、DensityPeak（密度最大值聚类）
- 基于连接的聚类算法
  - 谱聚类

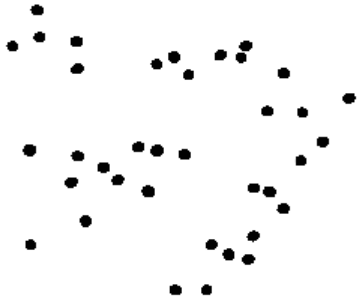
层次聚类 (Hierarchical Clustering)

- 层次聚类(Hierarchical clustering, 也称系统聚类法) 是最经典和常用的聚类方法之一
  - 能找到任意形状的类，而且不需指定类别数K
  - 需要度量样本点之间的距离以及类与类之间的联接 (linkage) 程度

- 系统聚类法包括两种
  - 聚合方法(agglomerative) / 自下而上:
    - 初始：每个样本作为单独的一类
    - 聚合：根据类间联接程度， 合并相近的类，直到所有的类合并成一个类
  - 分裂方法(divisive) / 自上而下
    - 初始：所有样本置于一个类
    - 分裂：一个类不断地分为更小的类，直到每个样本个体单独为一个类.

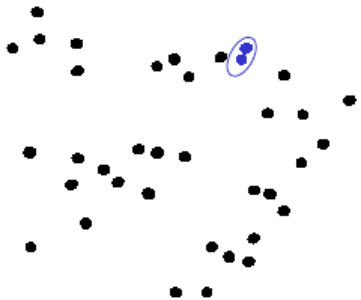
层次聚类

- 1、每个数据点为一类



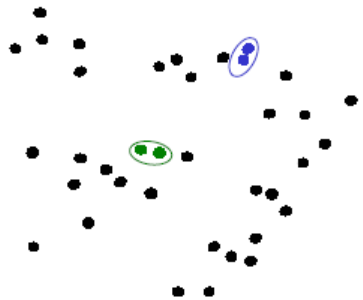
层次聚类

- 1、每个数据点为一类
- 2、找到两个最相似的类，并将其合并成 一个新的类



层次聚类

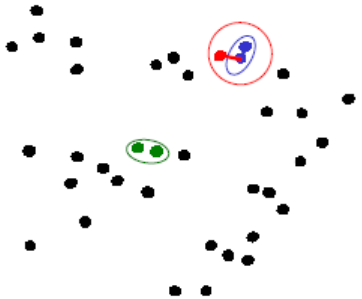
- 1、每个数据点为一类
- 2、找到两个最相似的类，并将其合并成 一个新的类
- 3、重复



层次聚类

如何度量两个类别之间相似性/距离？

- 1、每个数据点为一类
- 2、找到两个最相似的类，并将其合并成 一个新的类
- 3、重复

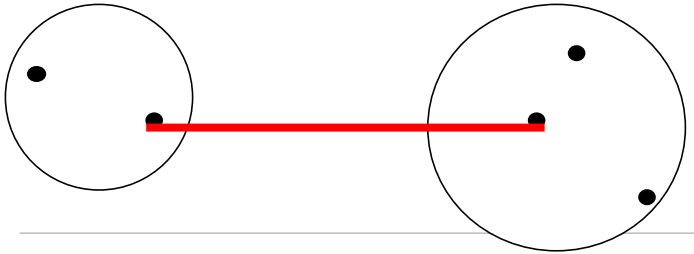


得到一个分类学的结果/dendrogram

类间联系程度度量 (Linkage criteria)

- 在合并两个最近的类别时，需要度量两个类别之间的距离。距离度量可选：
- 1.最小距离法（single linkage method）
  - 极小异常值在实际中不多出现，避免极大值的影响
  - 趋向于产生一个长链

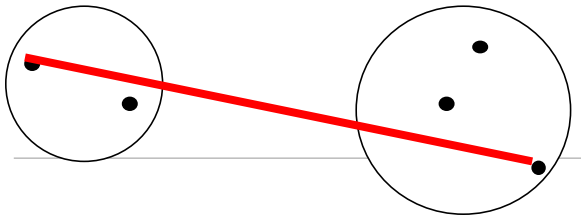
$$d_{\min}\left(C_k, C_{k'}\right) = \min_{x \in C_k, x' \in C_{k'}} \left\|\mathbf{x} - \mathbf{x}'\right\|$$



类间联系程度度量 (Linkage criteria)

- 在合并两个最近的类别时，需要度量两个类别之间的距离。距离度量可选：
- 2. 最大距离法（complete linkage method）
  - 可能被异常极大值扭曲，删除这些值之后再聚类
  - 趋向于产生产生一些紧致的类别

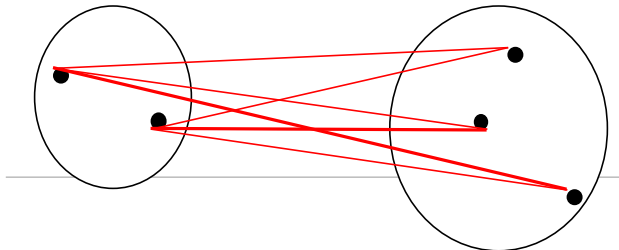
$$d_{\max}(C_k, C_{k'}) = \max_{\mathbf{x}\in C_k, \mathbf{x}'\in C_{k'}} \|\mathbf{x} - \mathbf{x}'\|$$



类间联系程度度量 (Linkage criteria)

- 在合并两个最近的类别时，需要度量两个类别之间的距离。距离度量可选：
- 3. 类平均距离法（ average linkage method ）
  - 类间所有样本点的平均距离
  - 利用了所有样本的信息，被认为是较好的系统聚类法

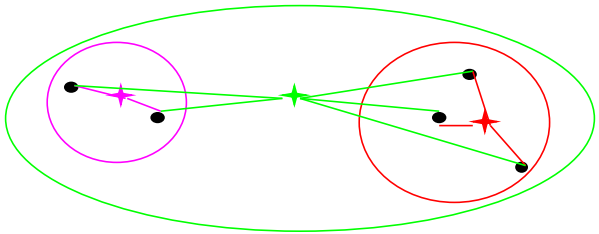
$$d_{avg}\left(C_k, C_{k'}\right) = \frac{1}{N_k N_{k'}} \sum_{\mathbf{x} \in C_k} \sum_{\mathbf{x}' \in C_{k'}} \left\| \mathbf{x} - \mathbf{x}' \right\|$$



类间联系程度度量 (Linkage criteria)

- 在合并两个最近的类别时，需要度量两个类别之间的距离。距离度量可选：
- 4. 离差平方和法（ward method）
  - 分别计算 类C_k和$C_{k'}$内各点到其均值的欧氏距离平方和W_k和$W_{k'}$；然后将两个类的点合并为一个类C_m并计算W_m

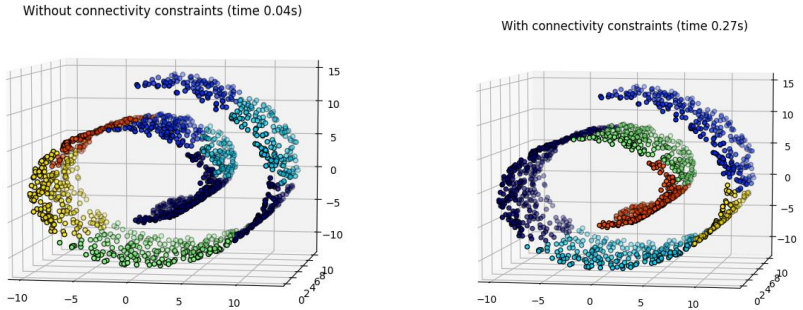
$$d_{ward} = W_{\mathrm{m}} - W_{C_k} - W_{C_{k'}}$$



  - 对异常值很敏感；
  - 对较大的类倾向产生较大的距离，从而不易合并，较符合实际需要

增加连接约束

- scikit learn实现的AgglomerativeClustering可以使用连接矩阵将连接约束添加到聚合层次聚类算法中
  - 只有相邻的聚类可以合并到一起
- 连接矩阵：为每个样本定义了邻域
  - 例如，在 swiss-roll 的例子中，连接约束禁止在不相邻的 swiss roll 上合并，从而防止形成在 roll 上 重复折叠的聚类



稀疏矩阵：一个样本只和少数样本相邻

Scikit learn中的AgglomerativeClustering

class sklearn.cluster.AgglomerativeClustering(*n_clusters=2, affinity='euclidean', memory=None, connectivity=None, compute_full_tree='auto', linkage='ward', pooling_func=<function mean>*)

Parameters:

n_clusters : int, default=2

The number of clusters to find.

类别数目

affinity : string or callable, default: "euclidean"

距离度量

Metric used to compute the linkage. Can be "euclidean", "l1", "l2", "manhattan", "cosine", or 'precomputed'. If linkage is "ward", only "euclidean" is accepted.

memory : None, str or object with the joblib.Memory interface, optional

Used to cache the output of the computation of the tree. By default, no caching is done. If a string is given, it is the path to the caching directory.

connectivity : array-like or callable, optional

样本连接矩阵约束

Connectivity matrix. Defines for each sample the neighboring samples following a given structure of the data. This can be a connectivity matrix itself or a callable that transforms the data into a connectivity matrix, such as derived from kneighbors_graph. Default is None, i.e, the hierarchical clustering algorithm is unstructured.

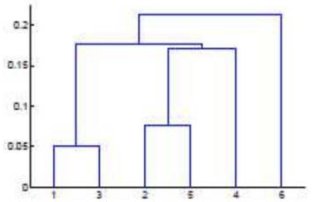
compute_full_tree : bool or 'auto' (optional)

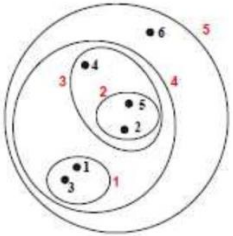
auto：n_clusters < max(100, .02 * n_samples)时计算完全树

Stop early the construction of the tree at n_clusters. This is useful to decrease computation time if the number of clusters is not small compared to the number of samples. This option is useful only when specifying a connectivity matrix. Note also that when varying the number of clusters and using caching, it may be advantageous to compute the full tree.

linkage : {"ward", "complete", "average"}, optional, default: "ward"

层次聚类





- 总结
  - 得到层次化的结果，而不是一堆无组织的类别
    - 一棵表示类别及其之间距离的树(称为*dendrogram*)
  - 层次聚类是一个确定的过程：没有参数
    - 如果只想得到*K*类，则剪掉*K*-1个最长的类
  - 但对层次聚类没有统计和信息基础
  - 运算量较大, 适用于不太大规模的数据
  - 一旦完成一个步骤(合并或分裂) , 就不能撤销或修正, 因此产生了改进的层次聚类方法, 如 BRICH, BURE, ROCK, Chameleon 等

CSDN
不 止 于 代 码

THANK YOU

AI100