# 聚类

算法的实现接收各种不同类型的矩阵作为输入。所有的方法接收[n\_samples, n\_features]这种类型的矩阵。这些可以从 [sklearn.feature\_extraction](http://scikit-learn.org/stable/modules/classes.html" \l "module-sklearn.feature_extraction" \o "sklearn.feature_extraction)模块中的类获得。对于[AffinityPropagation](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.AffinityPropagation.html" \l "sklearn.cluster.AffinityPropagation" \o "sklearn.cluster.AffinityPropagation)，[SpectralClustering](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html" \l "sklearn.cluster.SpectralClustering" \o "sklearn.cluster.SpectralClustering)和[DBSCAN](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html" \l "sklearn.cluster.DBSCAN" \o "sklearn.cluster.DBSCAN) 也可以输入形状[n\_samples，n\_samples]的相似矩阵。这些可以从 [sklearn.metrics.pairwise](http://scikit-learn.org/stable/modules/classes.html" \l "module-sklearn.metrics.pairwise" \o "sklearn.metrics.pairwise)模块中的函数获得。

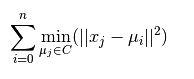
## K-means

<https://coolshell.cn/articles/7779.html>

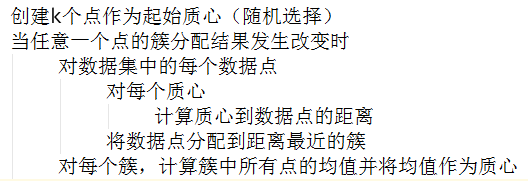
K-means通过把数据划到n个组（簇）、最小化inertia或簇内平方和(方差)来聚类，所以可以将KMeans视为具有每个分量相等协方差的高斯混合模型的特殊情况。

## 每个组用均值来描述。

所以目标就是：



算法描述：



在scikit-learn中使用k-means++来初始化质心。

## K-Means++

<https://en.wikipedia.org/wiki/K-means%2B%2B>

为了解决k-means算法中随机选择初始质心的弊端。

主要思想就是使初始的质心之间的相互距离要尽可能的远。

算法描述：

（1）从输入的数据点集合中随机选择一个点作为第一个聚类中心

（2）对于数据集中的每一个点x，计算它与最近聚类中心(指已选择的聚类中心)的距离D(x)

（3）选择一个新的数据点作为新的聚类中心，选择的原则是：D(x)较大的点。（被选取作为聚类中心的概率较大）

（4）重复步骤2和3直到k个聚类中心被选出来

（5）利用这k个初始的聚类中心来运行标准的k-means算法

注：较大点的选择：

<https://www.cnblogs.com/shelocks/archive/2012/12/20/2826787.html>

### **3、Mini Batch K-Means**

采用小批量的数据子集减小计算时间，同时仍试图优化目标函数。小批量是指每次训练算法时所随机抽取的数据子集。收敛时间更快，解决k-means在大数据量下出现的问题但聚类质量略差。

该算法的迭代步骤有两步：   
（1）从数据集中随机抽取一些数据形成小批量(mini batch)，把他们分别分配给最近的质心 。  
（2）通对于小批量中的每一个样本，取该样本和先前分配给该质心的样本的streaming average来更新质心。

1. 直到收敛或达到指定的迭代次数，停止计算。

## **4、Affinity Propagation（近邻传播）**

<http://blog.csdn.net/lixi__liu/article/details/48470173>

<http://blog.csdn.net/maoyaozong/article/details/40450067>

<http://www.mamicode.com/info-detail-1380807.html>

概述：近邻传播算法是一种基于代表点的聚类方法，将所有节点都看成潜在的聚类中心，然后通过节点之间的信息传递，去找出最合适的聚类中心，并将其他节点划分到这些中心下去，所以我们可以认为，AP算法所要做的事情就是去发现这些聚类中心。

概念：

Exemplar：聚类中心

Similarity：数据点i和点j的相似度记为s(i, j)。表示j作为i的聚类中心的合适程度。

Preference：数据点i的参考度称为p(i)或s(i,i)。表示点i作为聚类中心的参考度。由于AP的输入是一个节点间的相似度矩阵S，S矩阵的对角线上的数值即为s(i,i)。

Responsibility: r(i,k)用来描述点k适合作为数据点i的聚类中心的程度。

Availability:a(i,k)用来描述点i选择点k作为其聚类中心的适合程度。

所以：AP算法通过迭代过程不断更新每一个点的吸引度（Responsibility）和归属度（Availability）值，直到产生m个高质量的exemplar,同时将其余的数据点分配到相应的聚类中。

缺点：复杂度。时间复杂度,N是样本数量，T是迭代次数。如果使用的是稠密相似度矩阵，那么空间复杂度，如果是稀疏相似度矩阵，那么空间复杂度降低。所以AP算法只适合小中型数据集。

算法描述：

（1）初始化参数Damping factor与maxIterNum，并读取数据。

（2）先计算n个点之间的相似度值，将值放在s矩阵中，再选取p值(一般取s的中值)。设置一个最大迭代次数maxits (文中设默认值为1000)，迭代过程开始。   
（3）迭代的过程主要更新两个矩阵，代表(Responsibility)矩阵r=[r(i,k)],(n\*n)和适选(Availabilities)矩阵a=[a(i,k)],(n\*n)。这两个矩阵初始化为0，n是所有样本的数目。r(i,k)表示第k个样本适合作为第i个样本的类代表点的代表程度，a(i,k)表示第i个样本选择第k个样本作为类代表样本的适合程度。

（4）所有的样本的所属经过连续的convits次迭代都不再变化，或者迭代超过了maxits次。

# **5、均值漂移（Meanshift）算法**

<http://blog.csdn.net/jinshengtao/article/details/30258833>

1. 设想在一个有N个样本点的特征空间
2. 初始确定一个中心点center，计算在设置的半径为D的圆形空间内所有的点（xi）与中心点center的向量
3. 计算整个圆形空间内所有向量的平均值，得到一个偏移均值
4. 将中心点center移动到偏移均值位置，重复移动，直到满足一定条件结束

# **Hierarchical clustering（层次聚类）**

由于层次聚类需要连接小簇，所以需要使用不同的连接准则：（1）Ward：最小化连接的两个簇的方差

1. Maximum or complete linkage：最小化maximum distance between observations of pairs of clusters.
2. Average linkage：minimizes the average of the distances between all observations of pairs of clusters.

The guidelines for choosing a metric is to use one that maximizes the distance between samples in different classes, and minimizes that within each class.

# **DBSCAN**

DBSCAN聚类出的簇是核心对象的集合。那些在eps范围内，但不是核心对象的点被称为离群点。

# **BIRCH**

<http://www.cnblogs.com/pinard/p/6179132.html>

分两个阶段： 先聚类，再改善

1、BIRCH扫描数据库，建立一棵存放于内存的初始CF-树，它可以被看做数据的多层压缩，试图保留数据的内在聚类结构。（动态的构造CF树）---关键

2、BIRCH采用某个聚类算法对CF树的叶结点进行聚类，把稀疏的簇当作离群点删除，而把稠密的簇合并为更大的簇。（改进聚类质量）

　　 将第一步建立的CF Tree进行筛选，去除一些异常CF节点，这些节点一般里面的样本点很少。对于一些超球体距离非常近的元组进行合并

　　　　 利用其它的一些聚类算法比如K-Means对所有的CF元组进行聚类，得到一颗比较好的CF Tree.这一步的主要目的是消除由于样本读入顺序导致的不合理的树结构，以及一些由于节点CF个数限制导致的树结构分裂。

　　　 利用第三步生成的CF Tree的所有CF节点的质心，作为初始质心点，对所有的样本点按距离远近进行聚类。这样进一步减少了由于CF Tree的一些限制导致的聚类不合理的情况。