# Clustering

1. **k-means**
2. 目标：

将N个点聚类为k类，希望得到的k个类的方差和最小，记为第i类中心，为第i类点的集合，即：within-cluster sum of squares (WCSS)

1. 算法：
2. 从N个点中选择k个点分别作为初始的k个类中心
3. 将剩余点按照欧式距离归到k个初始类中

【贪婪算法，固定中心，若改变分类，则WCSS严格减小】

1. 重新计算每个类的中心

【贪婪算法，固定分类，若改变中心，则WCSS严格减小】

【若改为，

则变为求（各分量）中位数，由此产生的算法称为**k-medians**】

1. 重复B)和C)直至中心的变化小于指定阈值

【由于分类的有限性，和B,C的严格减小性，算法必定收敛】

1. 算法复杂度：

，其中t为迭代次数，N为聚类点的个数，d为数据的维数

1. 优点：

时间复杂度为线性，适合挖掘大规模数据集

1. 缺点：
2. k的选取难以估计
3. 初始聚类中心对结果影响较大

如：

①1,3作为初始点可以得到1,2一类3,4一类

②1,2作为初始点可以得到1,3一类2,4一类

且显然②比①更好的极小化目标函数WCSS

1. 优化：
2. k值的选取：

**ISODATA算法**

（Iterative Self Organizing Data Analysis Techniques Algorithm）

迭代自组织数据分析方法：（详见相关理论D）

给定数据，选择适当的期望聚类数K，初始聚类数k，每个聚类中最少需要的点N，标准偏差S，合并参数C，每次迭代允许合并的类数L，最大迭代次数I

①随机选k个点作为初始k类的中心

②将剩余点按照欧式距离归到k个初始类中，如果有一类点个数<N，则k=k-1，并将这类中的点按照欧式距离归到另外的k-1个类中去

③重新计算中心

④计算类内点到其中心的平均距离以及所有样本到其中心的平均距离

case A：若迭代到最后一次，结束算法

case B：k<=K/2，即分类数太少，需要分裂

case C：偶数次迭代或k>=2K，合并

case D：otherwise，分裂

分裂：若最大分量标准差

且(i) k<=K/2 或 (ii)且>2(N+1)

新的两个中心为：旧中心(max)分量±0.5

若分裂成功，跳过合并回到步骤②，否则考虑合并

合并：计算聚类中心间的距离，将的从小到大排列，然后开始合并，

一次最多不超过L对，注意从第2对开始要放弃涉及合并过的类，

重新计算聚类中心，然后回到步骤②

* 具体实例参见：http://www.cnblogs.com/huadongw/p/4101422.html

1. 简单初始聚类中心的选择：

Forgy method：随机选取初始聚类中心

Random Partition method：将点随机分为k类中某类然后从算法步骤C开始

1. 避免缺点B中离最优解可以无限远（长方形长>>宽）的初始聚类中心的选择：

想法：尽量让k个初始中心散开比较好（详见相关理论E）

**k-means++算法**

①从所有点中随机选一个点作为第一个中心

②对任一数据点，计算到最近中心的距离

③按照正比于的概率选取下一个中心

【编程技巧：将点编号，产生一个0~的随机数，从开始每次减去第一次小于等于0的下标对应点即是新产生的中心点】

④重复2,3直至k个中心都选好

1. 由于迭代一定步数后很少有点会跳出原来的聚类，所以大部分工作可以用三角不等式放缩不予以考虑来加速算法（详见相关理论C）：

Lemma1：若则

用法：设x为数据点，为中心，

找上界，若，

则数据点x应归到为中心的一类中，无需再算x到其他中心的距离

Lemma2：

用法：设x为数据点，分别为本次和上次的第j类中心，

如果能找到比较好的下界，则当中心只移动一点

也会是一个比较好的下界

Lemma1+Lemma2：设是x和（x属于的类新计算出得到的中心）的上界，是x和另外中心距离的下界，如果，那么就没必要计算【注意：一定不用计算，但是可能因为有别的而需要计算】

**（Elkan）改进的算法**

选择初始中心，初始化，然后对每个点x计算他的最近中心点，每次被计算时，令,【】

循环下列过程直至收敛：

①计算，再计算出

②找出所有满足的x【这样的x应仍属于c为中心的类】

③对所有剩下的点x，对中心c做循环：

(a) 如果r(x)=true：计算，令，r(x)=false

如果r(x)=false：【只计算一遍】

(b) 如果：

【不满足任一条件的x应仍属于c为中心的类】

计算，如果，令

【否则三角不等式没有帮助，直接计算比较，然后更新】

④对所有中心，重新计算新中心

⑤更新【下界如Lemma2用法中所述】

⑥更新，重置r(x)=true

【上界由三角不等式给出】

⑦用代替进入下一次迭代

1. 对异常值敏感，不robust；

对数字特征（身高/体重）取平均值可以，对类别特征（性别/种族）不很适用：

**k-medoids算法**

其中衡量不相似程度，常见的可以是取成不相似程度矩阵，

必须是某个数据点而不再是空间中随便一个点

①固定将各数据点分配给最近的中心（同k-means）

②外层循环：当前中心点（k个）内层循环：非当前中心点（N-1个）

设中心点变为内层循环的点，计算cost function，取最小的作为新中心点

注：这样会增加计算复杂度到，所以只适合数据量较小时使用

1. 相关理论：
2. 2011，Andrea Vattani，即使平面上k-means也可能需要2N迭代次数

http://cseweb.ucsd.edu/~avattani/papers/kmeans-journal.pdf

1. 2009，David Arthur，k-means在smoothed Complexity意义下（对每个原数据点做均值为0的Gaussian扰动）可以期望有多项式迭代次数（与扰动方差成反比，扰动方差越大迭代次数越少）

http://xueshu.baidu.com/s?wd=paperuri%3A%28d57859ad4ca7edfba7ebd04faa8208c3%29&filter=sc\_long\_sign&tn=SE\_xueshusource\_2kduw22v&sc\_vurl=http%3A%2F%2Farxiv.org%2Fpdf%2F0904.1113&ie=utf-8

1. 2003，Charles Elkan，利用三角不等式加速k-means算法

http://cseweb.ucsd.edu/~elkan/kmeansicml03.pdf

1. ISODATA算法的C++实现

http://blog.csdn.net/acdreamers/article/details/44663975

1. 2007，David Arthur，k-means++得到的WCSS期望意义下\*最小值

http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf

1. **Hierarchical Clustering Algorithms**
2. 构造N\*N对称矩阵，对角线上元素设为0，非对角线上元素为i与j的相似程度（越小说明越相似）
3. 找整个矩阵中非零元素最小的，设其行列为i,j (不妨设i>j)，去掉第j行和列，更新第i行和列为与原i,j相似度较大的那个（数值上即取小）
4. 循环②直至变为1个类
5. 整个过程可以画出一个树状图，即我们所需要的

* 具体实例参见：

http://home.deib.polimi.it/matteucc/Clustering/tutorial\_html/hierarchical.html

1. **Fuzzy C-Means Clustering**
2. 目标：极小化 subject to
3. 记号：是一个>1的实数，为点属于第类的概率，为第类的中心
4. 算法：（贪婪算法）

①初始化

②【固定极小化目标函数，求导可得】

③

【固定极小化带约束目标函数，用Lagrange乘子法求导简化后可得】

④循环直至变化的最大值小于给定阈值

1. 优势：允许一个点属于多个不同的类
2. **Gaussian Mixture Model(GMM)**
3. 假设数据符合混合高斯分布，

密度函数：

注：为了使其是个密度函数，必须满足

(i)中心极限定理说明假设合理 (ii)高斯分布在计算上比较方便

1. 用极大似然估计：

最大化对数似然函数

【取对数为了防止多个很小的数相乘发生下溢！】

① 对对数似然函数关于求导令为0：

令，

② 对对数似然函数关于求导令为0：

这里用到【难证！】，以及下面的证明

首先说明两个简单性质：

A.

B.

③ 对对数似然函数加约束的Lagrange函数关于求导令为0：

&&

第一式乘以对k求和，交换求和次序，将第二式代入：

代回第一式

1. 注意，上面的推导并不是显示解，因为每个等式右边都与自身仍相关，

所以事实上用的是迭代算法来完成求解：

①初始化，计算对数似然函数的值

②计算

③计算

④计算

⑤计算

⑥计算对数似然函数的值

⑦重复②~⑤直至对数似然函数值的改变量小于给定阈值

1. MATLAB code：http://blog.pluskid.org/?p=39
2. 意义：可以看做点属于第k类的概率
3. 改进：

①由于运算量较大，可以先用k-means做几步，然后用各类均值作为本方法的，点的比例作为本方法的，类内方差作为本方法的

②Gaussian函数中需要对求逆，若更新时奇异：

1. 观察什么时候奇异：

由r(AB)<=min{r(A),r(B)}以及r(A+B)<=r(A)+r(B)可以知道，如果N<d，肯定奇异，事实上，这是因为数据点太少，维数很高出现了overfitting的问题

1. 解决方法：降低模型复杂度

1）暴力：限制协方差矩阵为对角阵甚至纯量阵，这样就把d维问题转为 d 个1维问题，除非某个分量对所有点都相等，否则不会奇异

2）soft：在估计出来的协方差矩阵上加上一个很小的使之非奇异（半正 定矩阵加上一个正定矩阵一定正定），称为正则化系数，数值上 可以由向后误差分析说明解不会发生很大变化；意义上可以从 MAP (Maximum a posteriori)来解释，

其中为Conjugate Prior，为似然函数， 为归一化常数使之成为一个概率密度函数

取参数后验分布的最大值点作为估计，详细见：

http://freemind.pluskid.org/machine-learning/

regularized-gaussian-covariance-estimation/

1. **Spectral Clustering**
2. 算法：

①根据数据构造一个无向图，每个顶点代表一个数据点，每条带权边表示两个数据 点之间的相似度，形成对称的邻接矩阵

②将每一列元素(由于对称也即每一行)加起来得到n个数，把它们放在对角 线上构成一个n\*n对角阵记为，然后令

③求的最小的k个非0特征值对应特征向量

【有性质：半正定且0特征值对应线性无关特征向量只有(1,...,1)T】

④将特征向量排成一个n\*k的矩阵，每一行看成一个点做k-means（可以与上 面取最小k个特征值的k不一样），聚类的结果中每一行所属的类别就是原 来对应n个数据点分别所属的类别

1. 常用相似度选择方法：

①高斯核：

②欧氏距离：

1. 原理：

①Min cut：

极小化

其中，则

所以将离散问题松弛为连续问题，即在的条件下min，即求L的最 小特征值对应特征向量，不需要0特征值对应的全1特征向量，因为这种分法并

没有意义

②Ratio cut：

极小化

设，则，

+

由L和W定义，第一个等式对任意成立，无论如何定义

其中为类S中点的个数，为类T中点的个数

所以将离散问题松弛为连续问题，即在和的条件下min，由 L的性质（只有1重0特征值且对应特征向量为1，不同特征值对应的特征向量 必正交），即求L的最小非0特征值对应特征向量

③Normalized cut：

极小化

其中，有效防止孤立点出现

设，

则

=

=

所以将离散问题松弛为连续问题，即在和的条件下min， 由L的性质（）所以只要求的最小特征值对应特征向量即可