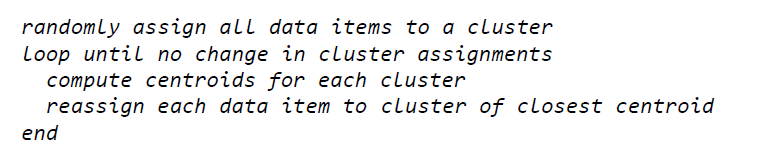
## K Means Algorithm

* K means算法是无监督算法
* K means需要预先知道聚簇的数目
* K means只能作用于数字类型的数据。



K-means算法是将样本聚类成k个簇（cluster），具体算法描述如下：

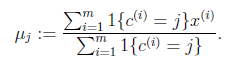
1、 随机选取k个聚类质心点（cluster centroids）为[clip_image008[6]](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104061601454064.png)。

2、 重复下面过程直到收敛 {

               对于每一个样例i，计算其应该属于的类

[clip_image009](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104061601464654.png)

               对于每一个类j，重新计算该类的质心

[](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104061601468308.png)

}

 K-means面对的第一个问题是如何保证收敛，前面的算法中强调结束条件就是收敛，可以证明的是K-means完全可以保证收敛性。下面我们定性的描述一下收敛性，我们定义畸变函数（distortion function）如下：

[clip_image016[6]](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/20110406160154496.png)

     J函数表示每个样本点到其质心的距离平方和。K-means是要将J调整到最小。

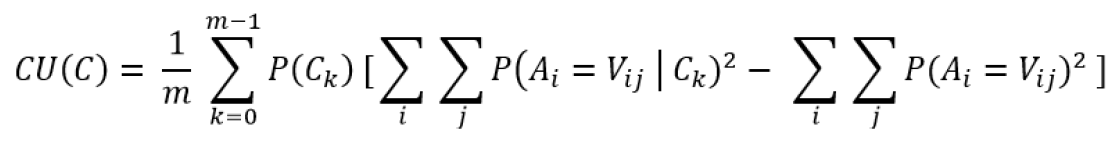
 由于畸变函数J是非凸函数，意味着我们不能保证取得的最小值是全局最小值，也就是说k-means对质心初始位置的选取比较感冒，但一般情况下k-means达到的局部最优已经满足需求。但如果你怕陷入局部最优，那么可以选取不同的初始值跑多遍k-means，然后取其中最小的J对应的[clip_image018[8]](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104061602007601.png)和c输出。

K means 需要数据归一化，因为不同的列数据差距不同。并且更新分组时，有时会产生空的分组。

## Categorical Data Clustering(CU)

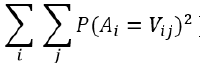
数据分组涉及到两个问题：1）找到评价分组好坏的标准，2）找到一种快速搜索所有分组的方法。

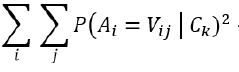
Category utility是用来衡量分类好坏的，值越高，分类越好。CU主要应用于枚举数据，也可以应用于数值数据（将数值分为指定的分组）。CU定义如下：



m:分组的个数

P:表示概率。 P(Ck)表示第k个分类的概率。

A:表示属性类型 V:表示属性值 。  该表达式无条件概率和。假设有N项数据，包含属性类型Ai等于该类型属性值Vij的个数为M，则P(Ai=Vij)=M/N。然后对加起来所有的项求和。表示猜中一个属性值的概率。

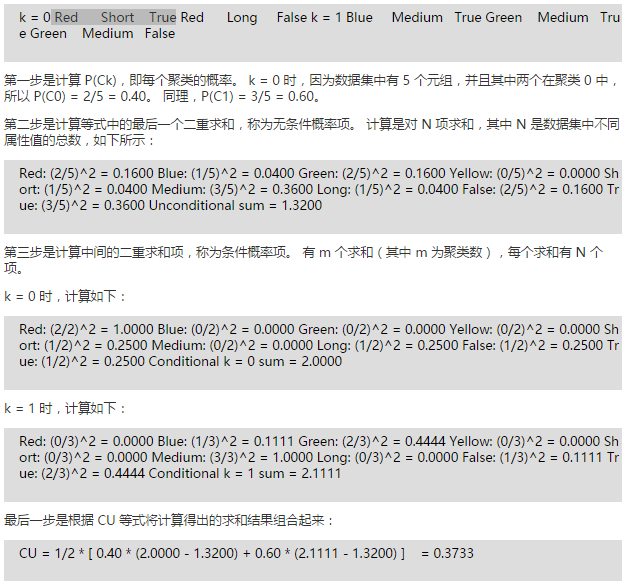
是条件概率。计算Ck分组条件下，各个项的概率，然后求和.表示在给定分组下猜中一个属性值的概率。因此差值越大，分组越好。

CU解决了衡量分组好坏的情况。

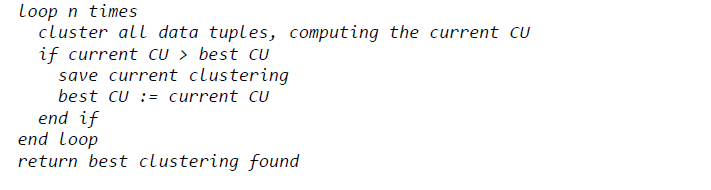
下一步是找到一种搜索所有可能分组的方法。GACUC是一种贪心算法。并不一定取得最优值。他需要事先指定分组数。具体思想是：每个分组先分配一个种子数据，对于剩下的数据项，确定分配到哪一个分组产生最大的CU,然后分配到该分组。

好的方法是选择尽可能互不相同的 m 个元组作为种子元组。 这里就又该是分类效用发挥作用了，可以计算所有可能的候选种子元组集的 CU，使用具有最佳 CU（最大值表示最不相同）的元组作为种子元组。 如上所述，一般不可能检查每个可能的种子元组集，因此我们重复选择 m 个随机元组，计算这些随机元组的 CU，并使用具有最佳 CU 的元组集作为种子。

greedy agglomerative category utility clustering (GACUC)需要事先指定分组的数目，但GACUC给出了一种判断分组好坏衡量的标准。



![C:\Users\cmlu\AppData\Roaming\Tencent\Users\975279703\QQ\WinTemp\RichOle\UBTAG(ZM](V]PQE)(4[3A9D.png](data:image/png;base64,)



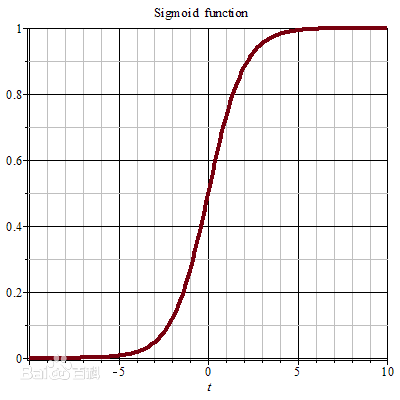
## 逻辑回归分类（Logistic Regression Classification）

对于输入特征是离散实数的特征向量[v1,v2,….vn]。

X = a0 + a1\*v1 + a2\*v2 + …… + an\*vn

http://d.hiphotos.baidu.com/baike/s%3D99/sign=a46bd6f1dd33c895a27e9472d01340df/0df3d7ca7bcb0a4659502a5f6f63f6246b60af62.jpg

S(x)是(0,1)范围的



归一化的目的是避免一个属性值比其它属性值的贡献要大太多，从而使其占主要作用。

归一化公式

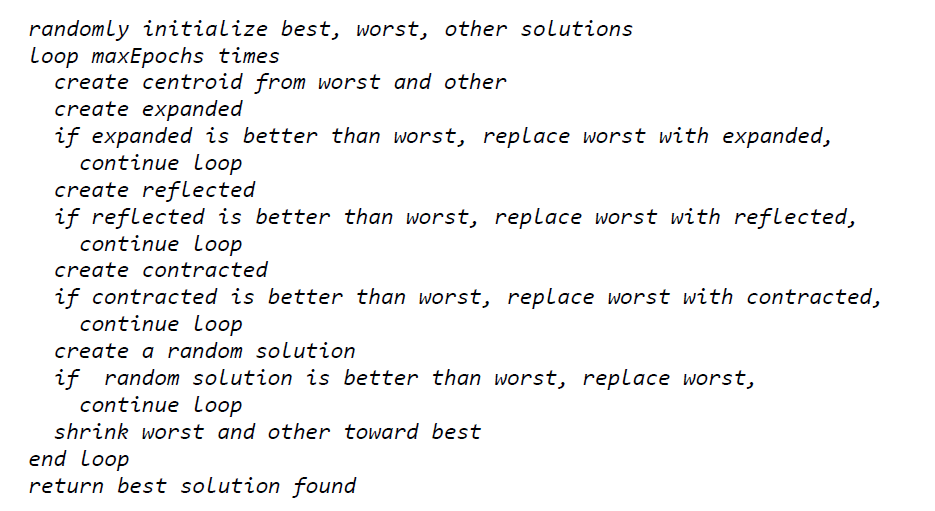
x = (x-μ)/ σ μ是平均值 σ是标准差

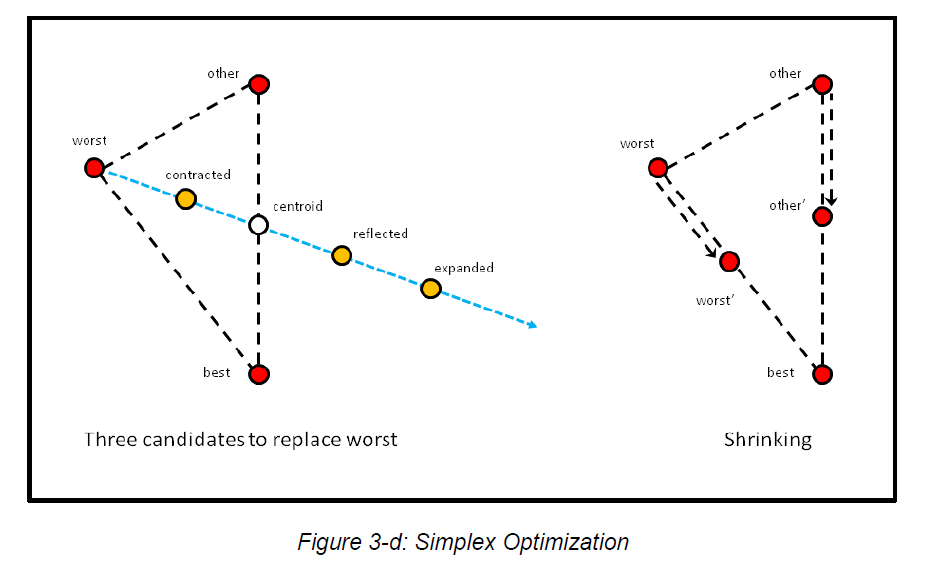
标准差公式：

http://c.hiphotos.baidu.com/baike/s%3D150/sign=49a3c31e8f1001e94a3c100a880f7b06/d058ccbf6c81800af3b703f9b33533fa838b47f3.jpg

训练数据的方法有：梯度下降  牛顿迭代法  反向传播  L-BFGS  simplex optimization(单纯形法) ， 这里使用单纯形法

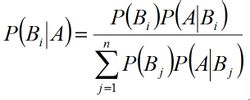
单纯形优化法：



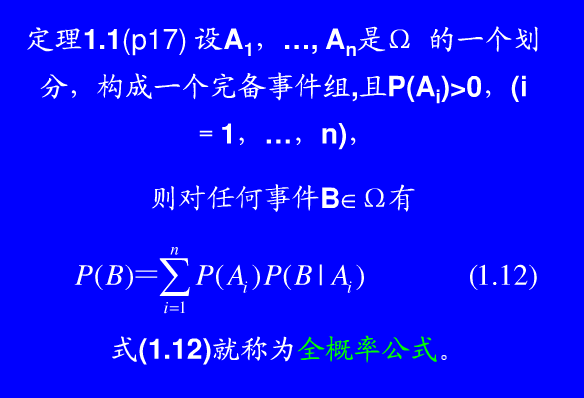


## 朴素贝叶斯分类 （Naïve Bayes Classification）

贝叶斯定理

[](http://baike.baidu.com/pic/%E8%B4%9D%E5%8F%B6%E6%96%AF%E5%AE%9A%E7%90%86/1185949/0/5fdf8db1cb134954a3ba2ede554e9258d0094ae0?fr=lemma&ct=single)

贝叶斯定理



朴素贝叶斯分类器公式：

假设个体有n个特征F1,F2,……,Fn。 有m个分类，分别为C1,C2,…..,Cm。贝叶斯分类是计算概率最大的那个分类。

P(Ci | F1F2…Fn) = P(Ci F1F2…Fn)/P(F1F2….Fn) = P(Ci)\*P(F1F2…Fn |Ci)/P(F1F2…Fn)

对于所有的类别，P(F1F2…Fn)都是一样的。只要求P(Ci)\*P(F1F2…Fn |Ci)的最大值就可以了

假设所有的特征F1,F2…Fn都是独立的（实际上是不可能的）。则:

P(Ci)\*P(F1F2…Fn |Ci) = P(Ci)\*P(F1|Ci)\*P(F2|Ci)\*…\*P(Fn|Ci)

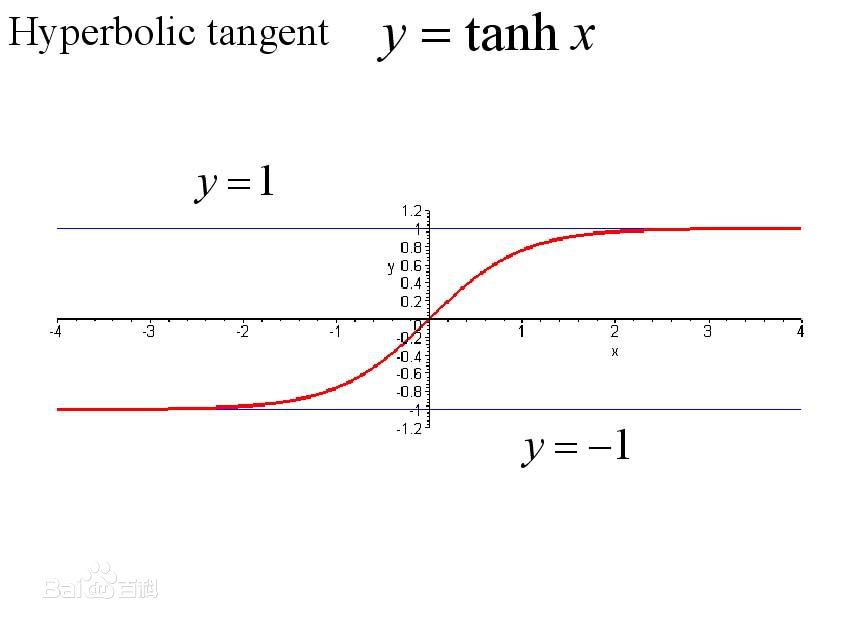
贝叶斯分类应用于枚举数据:

计算max(P(Ci | F1F2…Fn))转换成 计算 max(P(Ci)\*P(F1|Ci)\*P(F2|Ci)\*…\*P(Fn|Ci) ),观察到如果P(Fi|C)为0，整个结果为0，我们计算时做平滑处理。将count(Fi C)加1。

## Neural Network Classification(神经网络分类)

双曲正切函数 值域范围(-1,1)

http://f.hiphotos.baidu.com/baike/s%3D177/sign=3e3a856a520fd9f9a417516e122cd42b/77094b36acaf2edd760ee3ff8e1001e93901932d.jpg



Softmax 函数

https://pic4.zhimg.com/75938cc54604077d2ed193e97a5302bb_b.jpg

本书使用了粒子群优化算法，另一种可行的算法是反向传播（本书未实现）

高斯归一化和 Min-Max归一化

Min-Max归一化计算公司： （x-min）/（max-min）

高斯归一化： (x-mean)/stdDev mean表示均值 stdDev表示标准差

标准差 = Sqrt(方差) = sqrt ∑ (xi-mean)\*(xi-mean)/length