第 20 章

3：

输出节点相互竞争以针对产生的特定评分函数的最佳值，最常采用的是如下距离评价指标。

闵可夫斯基距离

闵氏距离不是一种距离，而是一组距离的定义，是对多个距离度量公式的概括性的表述。

$$\mathrm{dist}_{\mathrm{mk}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_{u=1}^{n} |x_{iu} - x_{ju}|^p \right)^{\frac{1}{p}} .$$

其中 p 是一个变参数：

- 当 p=1 时，就是曼哈顿距离；
- 当 p=2 时，就是欧氏距离；
- 当 p→∞时，就是切比雪夫距离。

曼哈顿距离

曼哈顿距离也叫”曼哈顿街区距离”。想象你在曼哈顿街道上，从一个十字路口开车到另一个十字路口，驾驶距离就是这个“曼哈顿距离”。

(1)二维平面上两点 a(x1,y1)，b(x2,y2)之间的曼哈顿距离公式：

$$d_{ab} = |x_1 - x_2| + |y_1 - y_2|$$

(2) n 维空间上两点 a(x1,x2.....xn)，b(y1,y2.....yn)的曼哈顿距离公式：

$$d_{ab} = |x_1 - y_1| + |x_2 - y_2| + ... + |x_n - y_n|$$

欧式距离

欧式距离：也称欧几里得距离，在一个 N 维度的空间里，求两个点的距离，这个距离肯定是一个大于等于零的数字，那么这个距离需要用两个点在各自维度上的坐标相减，平方后加和再开方。

$$\mathrm{dist}_{\mathrm{ed}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2 = \sqrt{\sum_{u=1}^{n} |x_{iu} - x_{ju}|^2} .$$

即数学上两点之间的距离，差的平方然后开根号

切比雪夫距离

国际象棋中，国王可以直行、横行、斜行。国王走一步，可以移动到相邻的 8 个方格的任意一个。国王从格子（X1,Y1） 到格子（X2,Y2）最少需要多少步？这个距离就是切比雪夫距离，简单理解为就是各坐标数值差的最大值

(1)二维平面上两点 a(x1,y1)，b(x2,y2)之间的切比雪夫距离公式：

$$d_{ab} = max(|x_1 - x_2|, |y_1 - y_2|)$$

(2) n 维空间上两点 a(x1,x2.....xn)，b(y1,y2.....yn)的切比雪夫距离公式：

$$d_{ab} = max(|x_1 - y_1|, |x_2 - y_2|, \ldots |x_n - y_n|)$$

杰卡德相似系数

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

集合之间的交并比，主要用来计算集合之间的相似度

余弦相似度

余弦相似度是用向量空间中两个向量夹角的余弦值作为衡量两个个体间差异的大小的度量。如果两个向量的方向一致，即夹角接近零，那么这两个向量就越相近。要确定两个向量方向是否一致，要用到余弦定理计算向量的夹角。

$$\cos(\theta) = \frac{a^T b}{|a| \cdot |b|}$$

Pearson 相似系数

皮尔逊相关系数也可以看成是剔除了两个变量量纲影响，即将 X 和 Y 标准化后的协方差。用来刻画两组变量之间的相关性

$$\rho_{XY}=\frac{\operatorname{cov}(X,Y)}{\sigma_X\sigma_Y}=\frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X\sigma_Y}=\frac{\sum_{i=1}^{n}(X_i-\mu_X)(Y_i-\mu_Y)}{\sqrt{\sum_{i=1}^{n}(X_i-\mu_X)^2}\sqrt{\sum_{i=1}^{n}(Y_i-\mu_Y)^2}}$$

相对熵(K-L 距离)

在一定程度上，熵可以度量两个随机变量的距离，KL 距离是两个概率分布 P 和 Q 差别的非对称性的度量。

$$D(p\parallel q)=\sum_x p(x)\log\frac{p(x)}{q(x)}=E_{p(x)}\log\frac{p(x)}{q(x)}$$

原因：聚类就是对大量未知标注的数据集，按数据的内在相似性将数据集划分为多个类别，使类别内的数据相似度较大而类别间的数据相似度较小，在此类问题中，输入字段与连接权重中具有最小欧式距离的输出节点将被宣布为获胜者。

4：

降低算法效率，算法效率非常关键，时间和空间上的复杂度之间可以相互补充。

6：

不适合，k 均值聚类仅考虑类与类距离上的关系，并无应用到簇

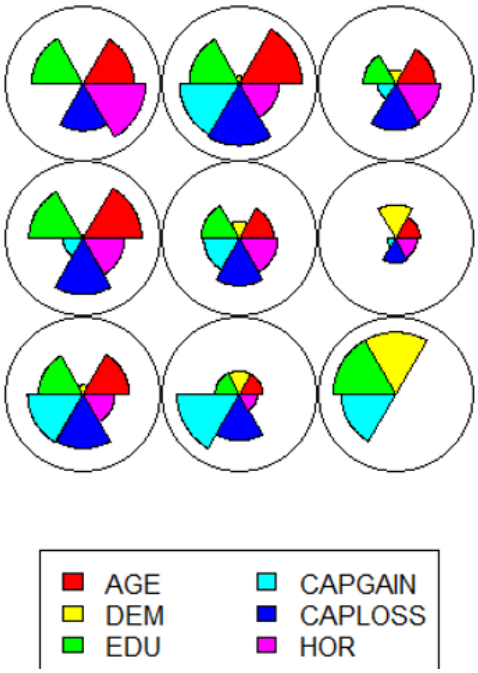
成员关系。在簇的平均值可被定义的情况下才能使用，可能不适用于某些应用，必须事先给出 k(要生成的簇的数目)，而且对初值敏感，对于不同的初始值，可能会导致不同结果。不适合于发现非凸形状的簇或者大小差别很大的簇，对躁声和孤立点数据敏感。

K-means 算法的思想很简单，简单来讲就是对于给定的样本集，按照样本之间的距离大小，将样本集划分为 K 个簇。让簇内的点尽量紧密的连在一起，而让簇间的距离尽量的大，两个对象之间的距离越近，相似性越高。聚类的结果就是使类内部的同质性高，而类之间的异质性高。

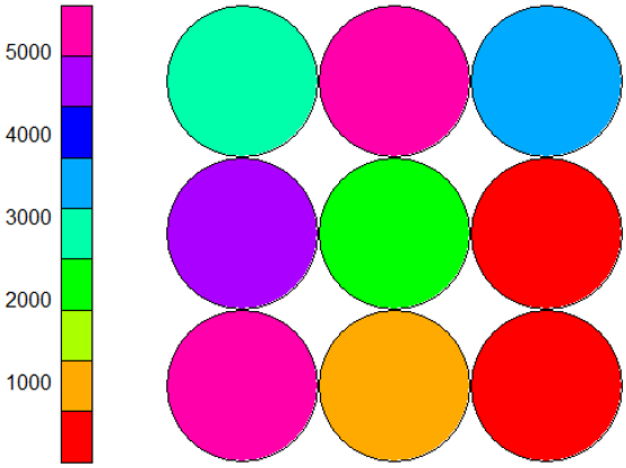
7，8:

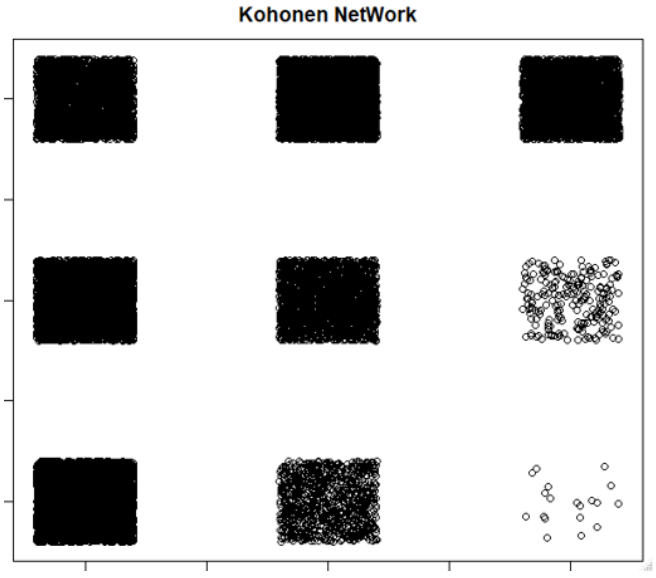
```
#打开kohonen包，读入数据并且准备数据
library(kohonen)
a<-read.csv("e:/bigdata/1pdXM5yKe9BJXTqsektX5UqUHgif/Clem3Training")
AGE<-(a$age-mean(a$age)/sd(a$age))
DEM<-(a$demogweight-mean(a$demogweight)/sd(a$demogweight))
EDU<-(a$education.num-mean(a$education.num)/sd(a$education.num))
CAPGAIN<-(a$capital.gain-mean(a$capital.gain)/sd(a$capital.gain))
CAPLOSS<-(a$capital.loss-mean(a$capital.loss)/sd(a$capital.loss))
HOR<-(a$hours.per.week-mean(a$hours.per.week)/sd(a$hours.per.week))
#运行kohonen算法，得到一个3X3的网格
data<-t(rbind(AGE,DEM,EDU,CAPGAIN,CAPLOSS,HOR))
som.6<-som(data,grid=somgrid(3,3),rlen=170,alpha=c(0.3,0.00),radius=2)
#画出每个簇进行观察
plot(som.6,type=c("codes"),palette.name = rainbow,main="Cluster counts")
#画出每个簇中的计数
plot(som.6,type=c("counts"),palette.name = rainbow,main="Cluster counts")
som.6$unit.classif#画出每个簇
som.6$grid$pts#描绘位置
```

Cluster content



Cluster counts




Kohonen NetWork