聚类分析与判别分析

聚类分析

1.数据变换

设有 n 个样品，每个样品测得 p 项指标，原始数据阵为

$$A = (a_{ij})_{n\times p}$$

其中 a_{ij} 为第 i 个样品的第 j 个指标的观测数据。

设变换后的数据为 b_{ij}，

中心化处理

令

$$b_{ij} = a_{ij} - \mu_j, \quad i = 1,2,\dots,n;\ j = 1,2,\dots,p,$$

其中

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} a_{ij}$$

规格化变换

令

$$b_{ij} = \frac{a_{ij} - \underset{1\le i\le n}{min}(a_{ij})}{\underset{1\le i\le n}{max}(a_{ij}) - \underset{1\le i\le n}{min}(a_{ij})}, \quad i = 1,2,\dots,n;\ j = 1,2,\dots,p,$$

标准化变换

令

$$b_{ij} = \frac{a_{ij} - \mu_j}{s_j}, \quad i = 1,2,\dots,n;\ j = 1,2,\dots,p,$$

其中

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} a_{ij}, \quad s_j = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(a_{ij} - \mu_j)^2}.$$

变换后所得矩阵为

$$B = (b_{ij})_{n\times p}.$$

2.指标间亲疏程度计算

任何两个样品 ω_k 和 ω_m 之间的相似性，都可以通过矩阵 B 的第 k 行和第 m 行的相似程度来描述；任何两个变量 x_k 和 x_m 之间的相似性，都可以通过矩阵 B 的第 k 列和第 m 列的相似程度来描述。

2.1.距离

令 d_{ij} 表示样品 ω_i 和 ω_j 之间的距离，
Minkowski距离

$$d_q(\omega_i,\ \omega_j) = (\sum_{k=1}^{p}|b_{ik} - b_{jk}|^q)^{\frac{1}{q}}$$

当 $q = 1$ 时，有

$$d_1(\omega_i,\ \omega_j) = \sum_{k=1}^{p}|b_{ik} - b_{jk}|,$$

即为 L_1 范数，又称Manhattan距离；
当 $q = 2$ 时，有

$$d_2(\omega_i,\ \omega_j) = (\sum_{k=1}^{p}|b_{ik} - b_{jk}|^2)^{\frac{1}{2}},$$

即为 L_2 范数，又称欧几里得距离；
当 $q = \infty$ 时，有

$$d_\infty(\omega_i,\ \omega_j) = \underset{1\le k\le p}{max}|b_{ik} - b_{jk}|,$$

即为 L_∞ 范数，又称切比雪夫距离。

Mahalanobis距离

$$d(\omega_i,\ \omega_j) = \sqrt{(\omega_i - \omega_j)\Sigma^{-1}(\omega_i - \omega_j)^T},$$

其中 $\Sigma = (\sigma_{ij})_{p\times p}$ ，为观测变量之间的协方差矩阵，有

$$\sigma_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(b_{ki} - \mu_i)(b_{kj} - \mu_j),\quad i,j = 1,2,\dots,p,$$

式中

$$\mu_m = \frac{1}{n}\sum_{k=1}^{n} b_{km}.$$

2.2.相似系数

夹角余弦

$$cos\theta_{ij} = \frac{\sum_{k=1}^{p} b_{ik}b_{jk}}{\sqrt{\sum_{k=1}^{p} b_{ik}^2} \cdot \sqrt{\sum_{k=1}^{p} b_{jk}^2}}, \quad i,j = 1,2,\ldots,n,$$

皮尔逊相关系数

$$r_{ij} = \frac{\sum_{k=1}^{p}(b_{ik}-\bar{\mu}_i)(b_{jk}-\bar{\mu}_j)}{\sqrt{\sum_{k=1}^{p}(b_{ik}-\bar{\mu}_i)^2} \cdot \sqrt{\sum_{k=1}^{p}(b_{jk}-\bar{\mu}_j)^2}}, \quad i,j = 1,2,\ldots,n,$$

其中

$$\bar{\mu}_m = \frac{1}{p}\sum_{k=1}^{p} b_{mk}.$$

3.Python模块的系统聚类

使用 `scipy.cluster.hierarchy` 模块，

```
import scipy.cluster.hierarchy as sch
# 使用由method指定的算法生成聚类树
Z = sch.linkage(y, method='single', metric='euclidean')

# method取值
# 'single' 最短距离; 'average' 平均距离; 'complete' 最大距离

# metric取值
# 'euclidean' 欧几里得距离(缺省); 'seuclidean' 标准欧几里得距离;
# 'cityblock' Manhattan距离; 'minkowski' Monkowski距离;
# 'chebyshev' 切比雪夫距离; 'mahalanobis' Mahalanobis距离;
# 'cosine' 1-两个向量夹角余弦; 'correlation' 1-样本相关系数.


# 从linkage的输出Z，根据给定的类数k创建聚类
```

```
T = sch.fcluster(Z, t=k, criterion='maxclust')

# 由linkage产生的数据矩阵Z画聚类树状图，p为节点数
H = sch.dendrogram(Z. p=30)
```

4.基于类间距离的系统聚类

系统聚类的基本思想为：距离相近的样品先聚为一类，距离远的后聚成类，依次下去，每个样品总能聚到合适的类中。

步骤：
1）将每个样品独自聚成一类，构造 n 个类；
2）根据所确定的样品距离公式，计算 n 个样品两两之间的距离，构造距离矩阵，记为 D_0 ；
3）将距离最近的两类归为一新类，其他样品仍然聚成一类，共聚成 $n-1$ 类；
4）计算新类与当前各类的距离，将距离最近的两个类进一步聚成一类，共聚成 $n-2$ 类；
5）重复上述步骤，最后将所有样品聚为一类。

类间距离定义
设类 G_i 与 G_j 之间的距离为 D_{ij} ，
最短距离：

$$D_{ij} = \min_{\omega_s \in G_i, \omega_t \in G_j} d(\omega_s, \omega_t)$$

最大距离：

$$D_{ij} = \max_{\omega_s \in G_i, \omega_t \in G_j} d(\omega_s, \omega_t)$$

重心距离：

$$D_{ij} = d(\bar{x}^{(i)}, \bar{x}^{(j)}),$$

其中 $\bar{x}^{(i)} = \frac{1}{n_i} \sum_{\omega_k \in G_i} \omega_k$ 为 G_i 的重心，n_i 为 G_i 中样本点个数；

类平均距离：

$$D_{ij} = \frac{1}{n_i n_j} \sum_{\omega_s \in G_i} \sum_{\omega_t \in G_j} d(\omega_s, \omega_t)$$

其中 n_i，n_j 分别为 G_i，G_j 中的样本点个数；

5.动态聚类法

K均值聚类算法最后将总样本集 G 划分为 C 个子集：$G_1, G_2, \ldots, G_C$，满足下面的条件：

$$(1)\ G_1 \cup G_2 \cup \cdots \cup G_C = G;$$
$$(2)\ G_i \cap G_j = \varnothing\ (1 \le i \le j \le C);$$
$$(3)\ G_i \ne \varnothing, G_i \ne G\ (1 \le i \le C).$$

记 $m_i (i = 1, 2, \ldots, C)$ 为 C 个聚类中心，定义

$$J_e = \sum_{i=1}^{C} \sum_{\omega \in G_I} ||\omega - m_i||^2$$

使 J_e 最小的聚类是误差平方和准则下的最优结果。

算法描述如下：
1）初始化。设总样本集 $G = \{\omega_j,\ j = 1, 2, \ldots, n\}$ 是 n 个样品组成的集合，聚类数为 $C(2 \le C \le n)$，将样本集 G 任意划分为 C 类，记为 $G_1, G_2, \ldots, G_C$，计算对应的 C 个初始聚类中心，记为 $m_1, m_2, \ldots, m_C$，并计算 J_e；
2）$G_i = \varnothing (i = 1, 2, \ldots, C)$，按照最小距离原则将样品 $\omega_j (j = 1, 2, \ldots, n)$ 重新进行聚类，即
若 $d(\omega_j,\ G_i) = \min\limits_{1 \le k \le C} d(\omega_j,\ m_k)$，则 $\omega_j \in G_i, G_i = G_i \cup \{\omega_j\}, j = 1, 2, \ldots, n$，聚类完成后，再计算新的聚类中心

$$m_i = \frac{1}{n_i} \sum_{\omega_j \in G_i} \omega_j,\ i = 1, 2, \ldots, C; j = 1, 2, \ldots, n,$$

其中 n_i 为当前 G_i 类中的样本数目，并重新计算 J_e；
3）若连续两次迭代 J_e 不变，则算法终止，否则转 2）。

python调用：

```
from sklearn.cluster import KMean

# k为聚类数，a为数据集
md = KMean(k).fit(a)
labels = md.labels_  # 提取聚类标签
centers = md.cluster_centers_  # 提取聚类中心
```

判别分析

1.距离判别法

设有 k 个总体 $G_1,\ G_2,\ \ldots,\ G_k$ ，它们的均值和协方差矩阵分别为 $\mu^{(i)},\ \Sigma^{(i)},\ i=1,2,\ldots,k$ ，
取样品 $X=[x_1,\ x_2,\ \ldots,\ x_p]^T$ ，则
1）当协方差矩阵相同时
判别函数为

$$W_{ij}(X)=(X-\frac{\mu^{(i)}+\mu^{(j)}}{2})^T\Sigma^{-1}(\mu^{(i)}-\mu^{(j)}),\quad i,j=1,2,\ldots,k,$$

判定准则为

$$\begin{cases}X\in G_i, & W_{ij}(X)>0,\forall\ j\neq i,\\ \text{待判}, & \exists\ i,j,\ s.t.\ W_{ij}(X)=0.\end{cases}$$

2）当协方差矩阵不相同时
判别函数为

$$V_{ij}=(X-\mu^{(i)})(\Sigma^{(i)})^{-1}(X-\mu^{(i)})-(X-\mu^{(j)})(\Sigma^{(j)})^{-1}(X-\mu^{(j)}),\quad i,j=1,2,\ldots,k,$$

判别准则为

$$\begin{cases}X\in G_i, & V_{ij}(X)<0,\forall\ j\neq i,\\ \text{待判}, & \exists\ i,j,\ s.t.\ W_{ij}(X)=0.\end{cases}$$

2.Fisher判别

设两个 p 维总体为 G_1,G_2 ，它们的均值向量分别为 μ_1,μ_2 ，且有公共的协方差矩阵 $\Sigma(\Sigma>0)$，取样品 $X=[x_1,\ x_2,\ \ldots,\ x_p]^T$ ，令

$$K=\frac{1}{2}(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1+\mu_2),$$

则判别函数为

$$W(X)=(\mu_1-\mu_2)^T\Sigma^{-1}X-K,$$

判别准则为

$$\begin{cases}X\in G_1, & W(X)\geq 0,\\ X\in G_2, & W(X)<0.\end{cases}$$

3.判别准则评价

采用交叉误判率估计，
算法：
1）从总体 G_1 的样品开始，提出其中一个样品，剩余的 $m-1$ 个样品与 G_2 的全部样品建

立判别函数;
2）用建立的判别函数对提出的样品进行判别;
3）重复步骤 1）和 2），直到 G_1 中的全部样品依次被删除又进行判别，其误判的样品个数记为 N_1^* ;
4）对 G_2 的样品类似地重复 1）、2）和 3），直到 G_2 中的全部样品依次被删除又进行判别，其误判的个数记为 N_2^* ;
5）最终得到交叉误判率为

$$\hat{p}^* = \frac{N_1^* + N_2^*}{m+n}.$$

Python调用如下:

```
from sklearn.model_selection import cross_val_score

# model为建立的模型;x0为已知样本点的数据;
# y0为已知样本的标号值;cv=k表示已知样本被分为k组
r = cross_val_score(model, x0, y0, cv=k)
```