第3讲 MATLAB数据挖掘算法（上）

卓金武
MathWorks 中国
steven.zhuo@mathworks.cn

课程介绍

- **第1讲：MATLAB快速入门**
- MATLAB快速入门实例
- MATLAB实用操作技巧
- MATLAB数据类型
- MATLAB程序结构
- MATLAB编程模式
- MATLAB学习理念

- **第2讲：MATLAB数据挖掘基础**
- MATLAB数据挖掘的过程
- 数据的可视化
- 数据的预处理
- 数据的探索
- 假设检验
- 数据回归

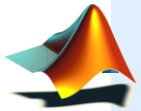
- **第3讲：MATLAB数据挖掘算法（上）**
- 回归算法
- 关联算法
- 聚类算法

- **第4讲：MATLAB数据挖掘算法（下）**
- 分类算法
- 预测算法
- 异常诊断算法

- **第5讲：MATLAB高级数据挖掘技术**
- MATLAB分类学习机
- 算法的高级使用方法
- 综合使用实例

- **第6讲：MATLAB数据挖掘项目实例**
- 故障诊断
- 生物信息学研究
- 量化投资

内容提要



回归算法

- 关联算法
- 聚类算法

回归方法

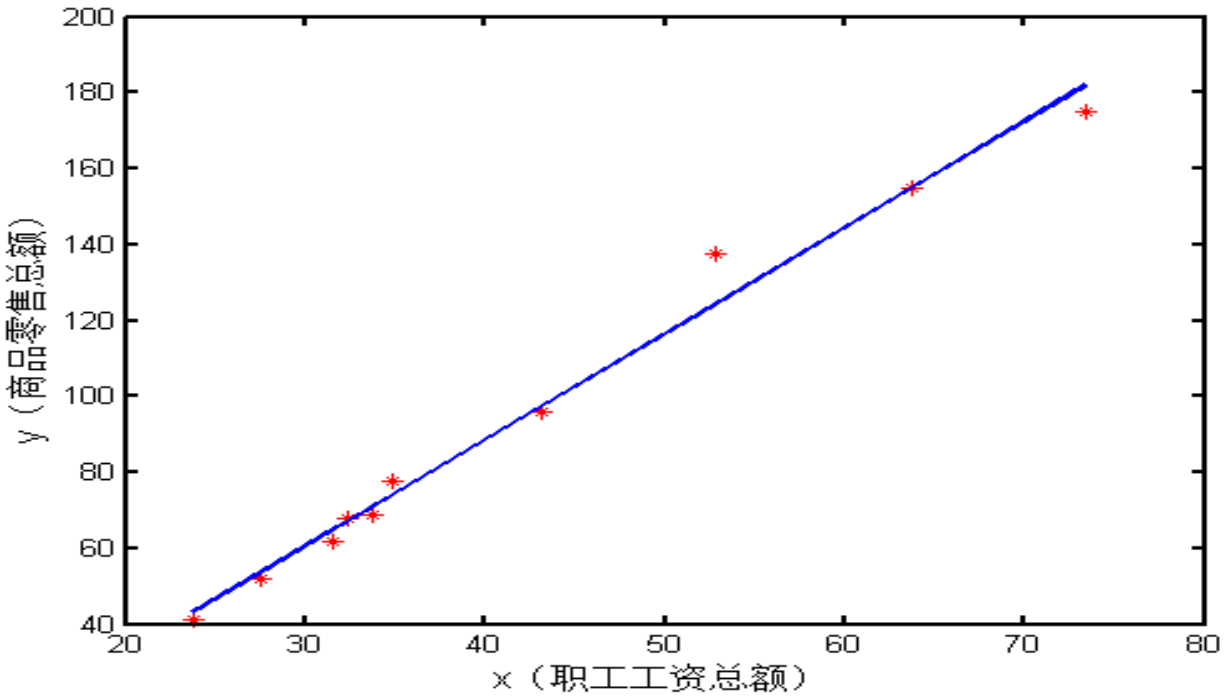
1. 一元回归
2. 多元回归
3. 逐步回归
4. **Logistic回归**

一元线性回归

一元线性回归模型的一般形式为：

$$Y = \beta_0 + \beta_1 x + \varepsilon$$

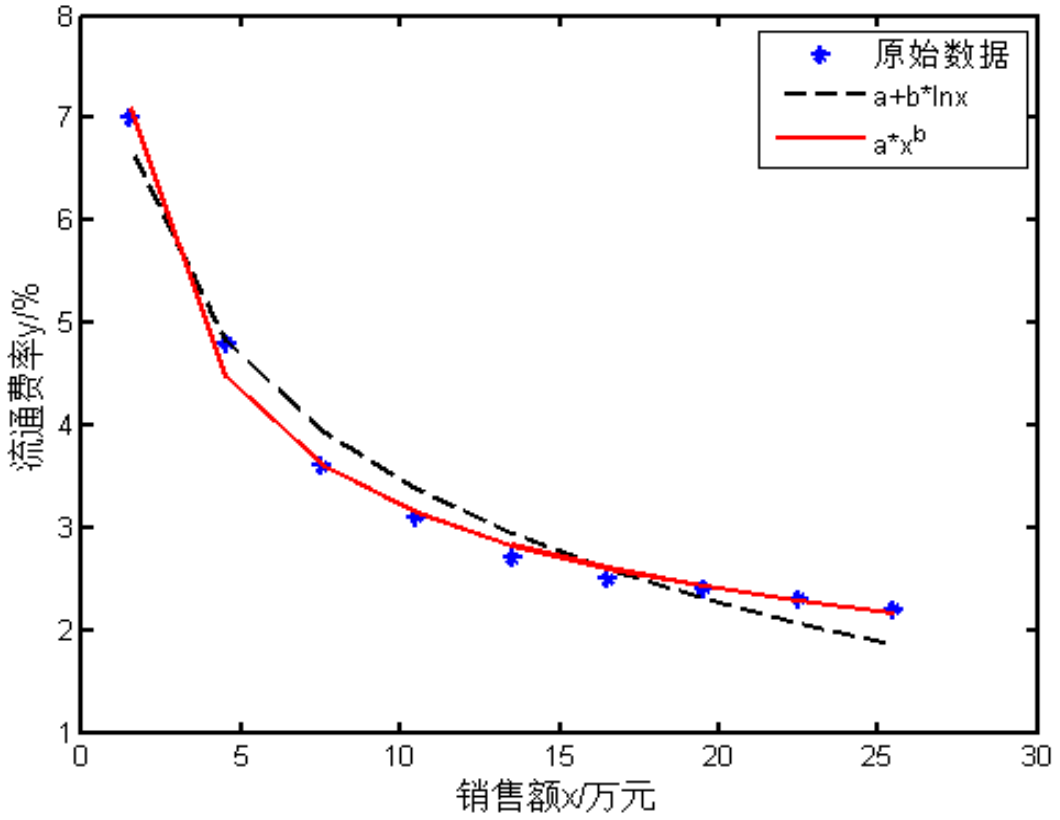
Matlab函数：
LinearModel.fit
regress



一元非线性回归

在一些实际问题中，变量间的关系并不都是线性的，那时就应该用曲线去进行拟合。

Matlab函数：
fitnlm



一元多项式回归

一元多项式回归模型的一般形式为：

$$y = \beta_0 + \beta_1 x + \cdots + \beta_m x^m + \varepsilon$$

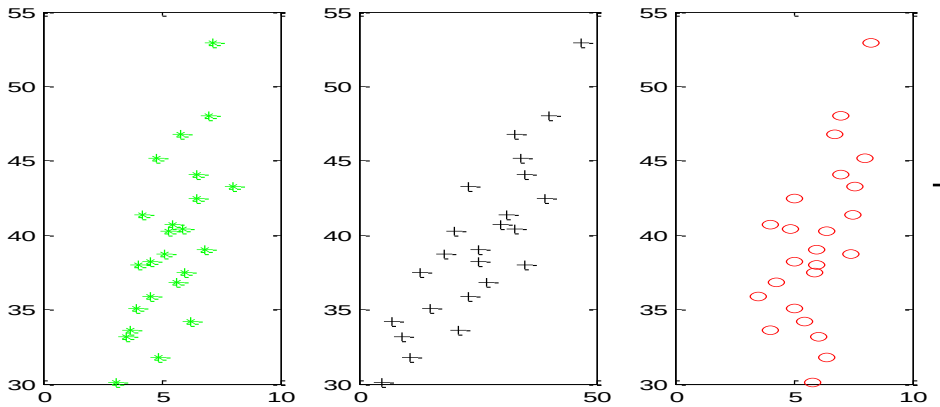
Matlab函数：
polyfit

多元线性回归

多元线性回归模型的一般形式为：

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots \beta_p X_p + \varepsilon$$

（1）



→ 有比较好的线性关系，可以采用线性回归。

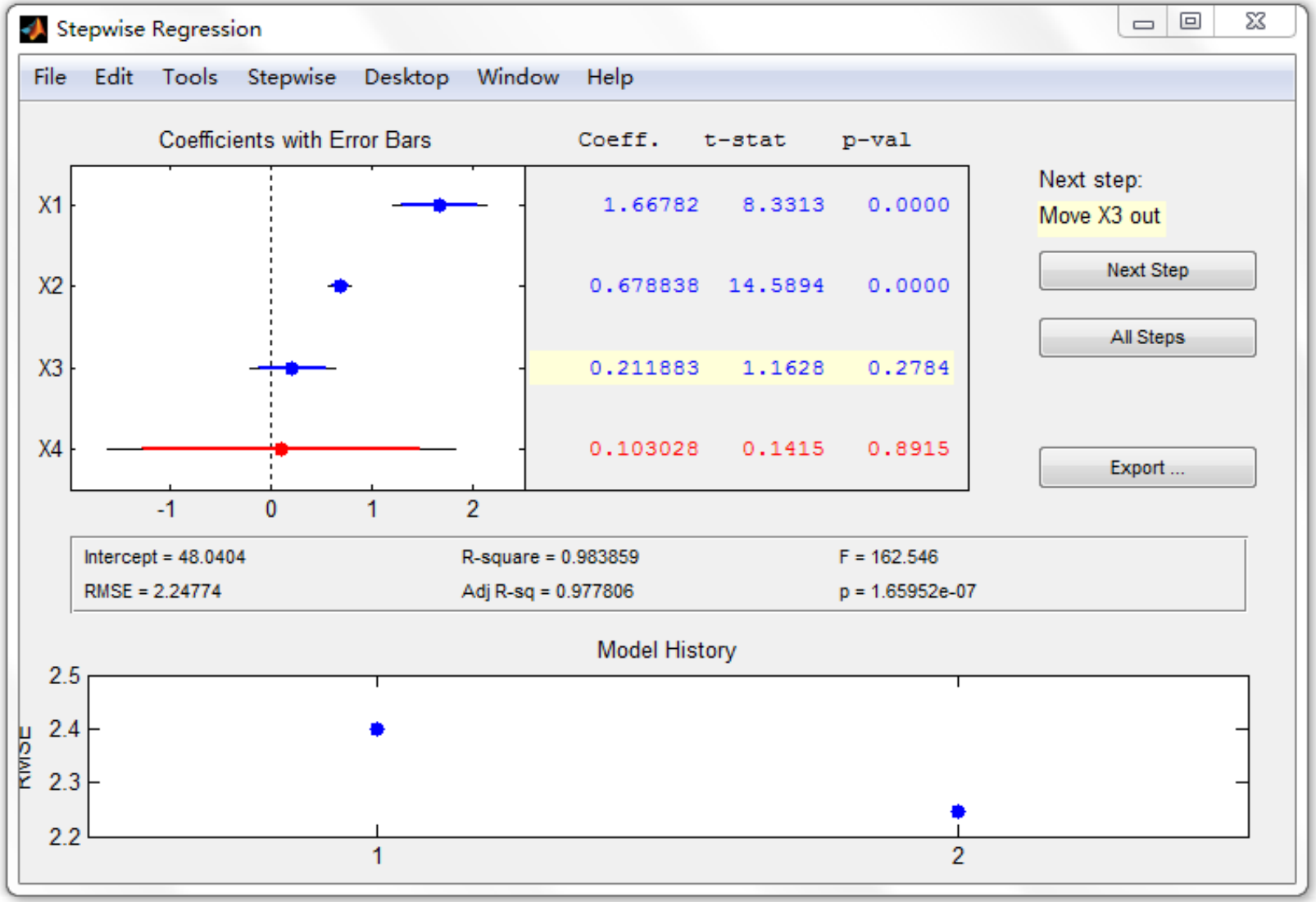
（2）

regress函数 →

回归系数	回归系数的估计值	回归系数的置信区间
β_0	18.0157	[13.9052 22.1262]
β_1	1.0817	[0.3900 1.7733]
β_2	0.3212	[0.2440 0.3984]
β_3	1.2835	[0.6691 1.8979]
R^2 =0.9106 F=67.9195 p<0.0001 s^2= 3.0719		

逐步回归的MATLAB方法

Matlab函数：
stepwise



Logistic模型

Logistic回归模型的基本形式为：

$$P(Y = 1 \mid x_1, x_2, \cdots, x_k) = \frac{\exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)}$$

该式表示当变量为 $x_1, x_2, \cdots, x_k$ 时，自变量 p 为1的概率。对该式进行对数变换，可得：

$$\ln \frac{p}{1 - p} = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

令 $\pi = P(Y = 1 \mid x_1, x_2, \cdots, x_k)$，$0 < \pi < 1$， Logistic模型就可变形为：

$$\ln \frac{\pi}{1 - \pi} = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

然后只需要对原始数据进行合理的映射处理，就可以用线性回归方法得到回归系数。

Logistic回归实例

（1）确定概率函数π和评价结果p之间的映射关系

对于$p=0$，$\pi=(0+0.5)/2=0.25$

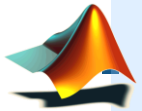
对于$p=1$，$\pi=(0.5+1)/2=0.75$

（2）利用MATLAB进行求解（ P6_1_logisctic_ex1.m ），得到该问题的logistic回归模型为

$$\begin{cases}\pi=\dfrac{\exp(-0.63656+0.004127x_1+0.016292x_2\cdots+0.53305x_3)}{1+\exp(-0.63656+0.004127x_1+0.016292x_2\cdots+0.53305x_3)}\\ P=\begin{cases}0, & \pi\le 0.5\\ 1, & \pi>0.5\end{cases}\end{cases}$$

内容提要

- 回归算法



关联算法

- 聚类算法

关联方法

1. Apriori算法
2. FP-Growth算法
3. 相关系数法

Apriori算法步骤

①扫描全部数据，产生候选1-项集的集合C1；

②根据最小支持度，由候选1-项集的集合C1产生频繁1-项集的集合L1；

③对k>1，重复执行步骤④、⑤、⑥；

④由Lk执行连接和剪枝操作，产生候选（k+l）-项集的集合Ck+1；

⑤根据最小支持度，由候选（k+l）-项集的集合Ck+1，产生频繁（k+1）-项集的集合Lk+1；

⑥若L≠Φ，则k=k+1，跳往步骤④；否则，跳往步骤⑦；

⑦根据最小置信度，由频繁项集产生强关联规则，结束。

Apriori算法实例

第一次扫描

C1

项集	支持度计数
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

→

L1

项集	支持度计数
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

第二次扫描

C2

项集	支持度计数
{I1，I2}	4
{I1，I3}	4
{I1，I4}	1
{I1，I5}	2
{I2，I3}	4
{I2，I4}	2
{I2，I5}	2
{I3，I4}	0
{I3，I5}	1
{I4，I5}	0

→

L2

项集	支持度计数
{I1，I2}	4
{I1，I3}	4
{I1，I5}	2
{I2，I3}	4
{I2，I4}	2
{I2，I5}	2

第三次扫描

C3

项集	支持度计数
{I1，I2，I3}	2
{I1，I2，I5}	2

→

L3

项集	支持度计数
{I1，I2，I3}	2
{I1，I2，I5}	2

第四次扫描

C4= Φ ⟶ 算法终止

Apriori算法程序实现

事务列表 —映射→

	I1	I2	I3	I4	I5
T100	1	1	0	0	1
T200	0	1	0	1	0
T300	0	1	1	0	0
T400	1	1	0	1	0
T500	1	0	1	0	0
T600	0	1	1	0	0
T700	1	0	1	0	0
T800	1	1	1	0	1
T900	1	1	1	0	0

事务矩阵

↓ 程序P6-1

```
     1     0     0     0     0     6
     0     1     0     0     0     7
     0     0     1     0     0     6
     0     0     0     1     0     2
     0     0     0     0     1     2
     1     1     0     0     0     4
     1     0     1     0     0     4
     1     0     0     0     1     2
     0     1     1     0     0     4
     0     1     0     1     0     2
     0     1     0     0     1     2
     1     1     1     0     0     2
     1     1     0     0     1     2
```

频繁项集的0-1映射矩阵

FP-Growth算法步骤

第一步：按以下步骤构造FP-tree：

(a) 扫描事务数据库D 一次。收集频繁项的集合F和它们的支持度。对F按支持度降序排序，结果为频繁项表L。

(b) 创建FP-树的根结点，以“null”标记它。对于D中每个事务Trans，执行：选择Trans中的频繁项，并按L中的次序排序。设排序后的频繁项表为[p|P]，其中，p是第一个元素，而P是剩余元素的表。调用insert_tree([p|P], T)。该过程执行情况如下。如果T有子女N使得N.item-name = p.item-name，则N的计数增加1；否则创建一个新结点N，将其计数设置为1，链接到它的父结点T，并且通过结点链结构将其链接到具有相同item-name的结点。如果P非空，递归地调用insert_tree(P, N)。

第二步：根据FP-tree挖掘频繁项集，该过程实现如下：

(1) if Tree 含单个路径P then
(2) for 路径 P 中结点的每个组合（记作β）
(3) 产生模式β∪α，其支持度support = β中结点的最小支持度
(4) else for each a i 在 Tree 的头部 {
(5) 产生一个模式β = a i ∪α，其支持度support = a i .support
(6) 构造β的条件模式基，然后构造β的条件FP-树Treeβ
(7) if Treeβ ≠∅ then
(8) 调用 FP_growth (Treeβ, β)；}

FP-Growth算法实例（1）

第一步：构造FP-tree

I1	I2	I3	I4	I5
6	7	6	2	2

排序

I2	I1	I3	I4	I5
7	6	6	2	2

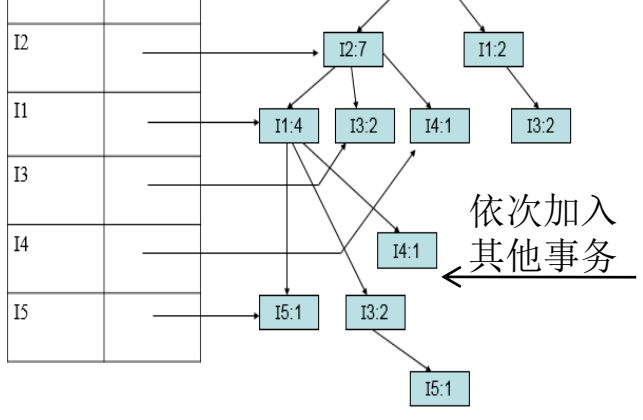
重新调整事务数据库

Tid	Items
1	I2, I1,I5
2	I2,I4
3	I2,I3
4	I2, I1,I4
5	I1,I3
6	I2,I3
7	I1,I3
8	I2, I1,I3,I5
9	I2, I1,I3

创建根结点和频繁项目表

Item-name	Node-head
I2	Null
I1	Null
I3	Null
I4	Null
I5	Null

加入第一个事务（I2，I1，I5）

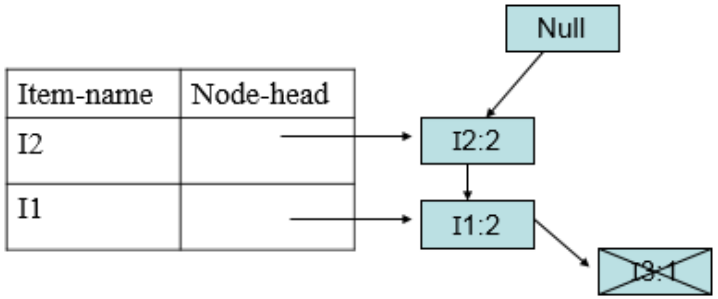


依次加入其他事务

FP-Growth算法实例（2）

第二步：根据FP-tree挖掘频繁项集

（1）首先考虑I5，得到条件模式基：<(I2,I1:1)>、<I2,I1,I3:1>, 并构造条件FP-tree：



得到I5频繁项集：{{I2,I5:2},{I1,I5:2},{I2,I1,I5:2}}。

（2）同理，依次考虑I4，I3，I1，可以得到以下频繁项集：
I4频繁项集：{{I2,I4:2}}；
I3频繁项集：{{I2,I3:4},{I1,I3:4},{I2,I1,I3:2}}；
I1频繁项集：{{I2,I1:4}。

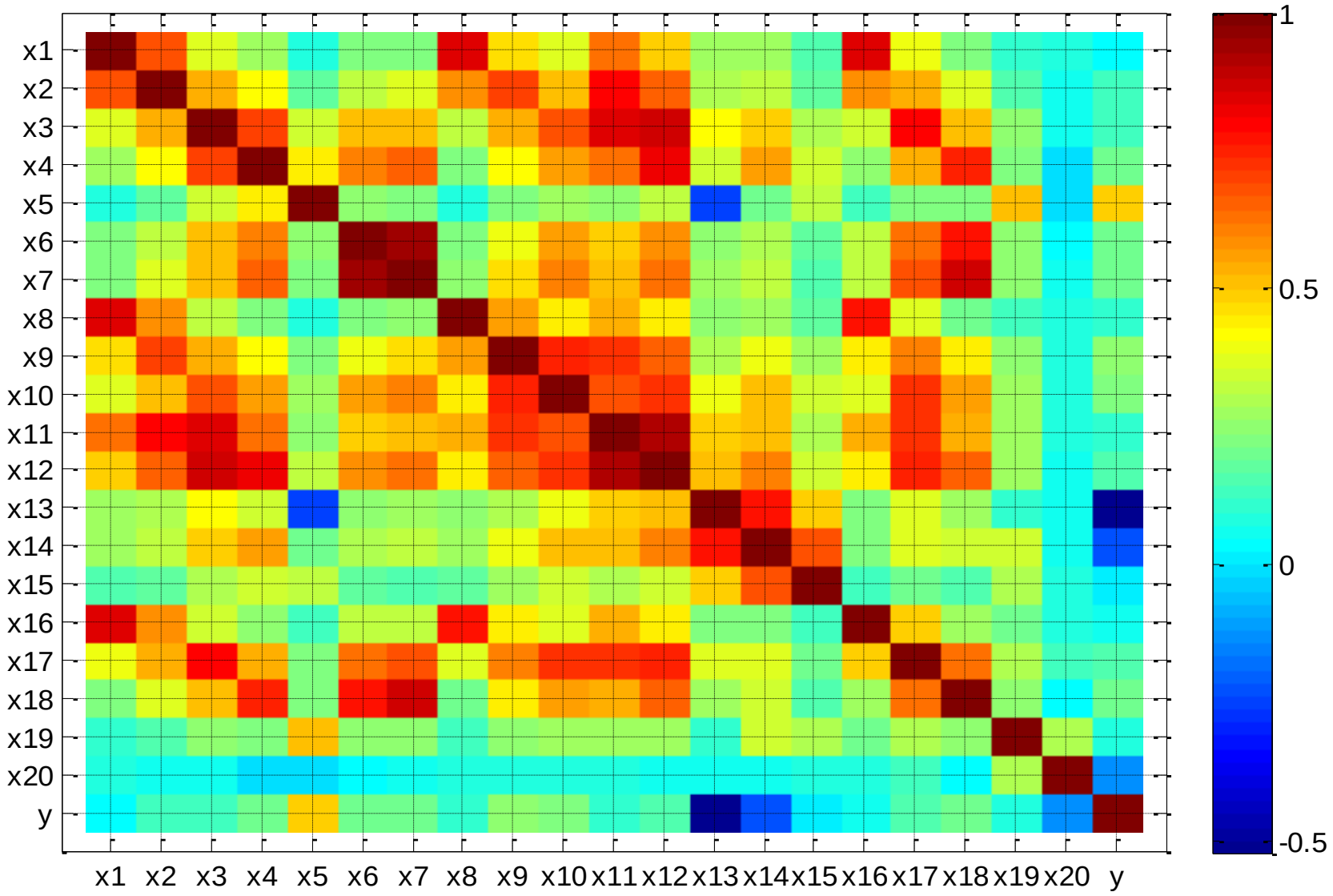
相关系数法

相关系数用*r*表示，*r*在-1和+1之间取值。相关系数*r* 的绝对值大小（即| *r* |）表示两个变量之间的直线相关强度。相关系数r的正负号，表示相关的方向，分别是正相关和负相关。

r=0：零相关；
r=1：完全相关。

0.7<|r |<1：高度相关；
0.4<|r |<0.7：中度相关；
0.2<|r |<0.4：低度相关；
|r |<0.2：极低相关或接近零相关；

相关系数法的实现


x1
x2
x3
x4
x5
x6
x7
x8
x9
x10
x11
x12
x13
x14
x15
x16
x17
x18
x19
x20
y
x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15 x16 x17 x18 x19 x20 y
1
0.5
0
-0.5

关联算法应用实例：行业关联选股法

	银行	券商	钢铁	能源	医药	化工
T1	1	1	0	0	1	0
T2	0	0	0	1	0	1
T3	1	1	1	0	0	0
T4	1	1	0	1	0	1
T5	0	0	1	0	1	0
T6	0	1	1	0	0	0
T7	1	0	1	0	0	0
T8	1	1	1	0	1	1
T9	1	1	1	0	0	0
T10	1	1	0	1	0	0

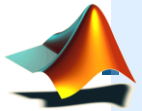
Apriori算法 →

1	0	0	0	0	0	7
0	1	0	0	0	0	7
0	0	1	0	0	0	6
0	0	0	1	0	0	3
0	0	0	0	1	0	3
0	0	0	0	0	1	3
1	1	0	0	0	0	6
1	0	1	0	0	0	4
0	1	1	0	0	0	4
1	1	1	0	0	0	3

由程序的执行结果可以看出，满足最小支持度3的包含3个行业项的项集只有一个，即{银行，券商，钢铁：3/10}，这说明这3个行业在一定周期内具有较高(3/10)的联动可能性。
再看包含2个行业项的项集，这里发现2个：{银行，钢铁：4/10}、{银行，券商：4/10}，而且它们出现联动的概率是一致的，都为4/10，所以在实践中，如果出现这3个行业中的一个出现涨势，那么就可以考虑从其他两个行业中选择代表性的股票进行买入，这样就可以在其他行业还没上涨的时候，提前埋伏进去，以此获得较高的收益。

内容提要

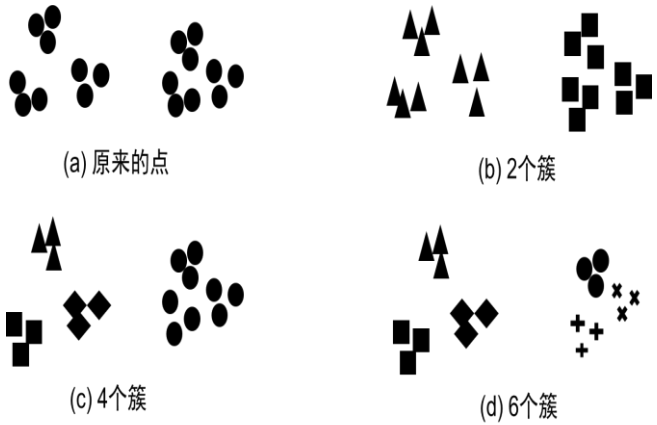
- 回归算法
- 关联算法



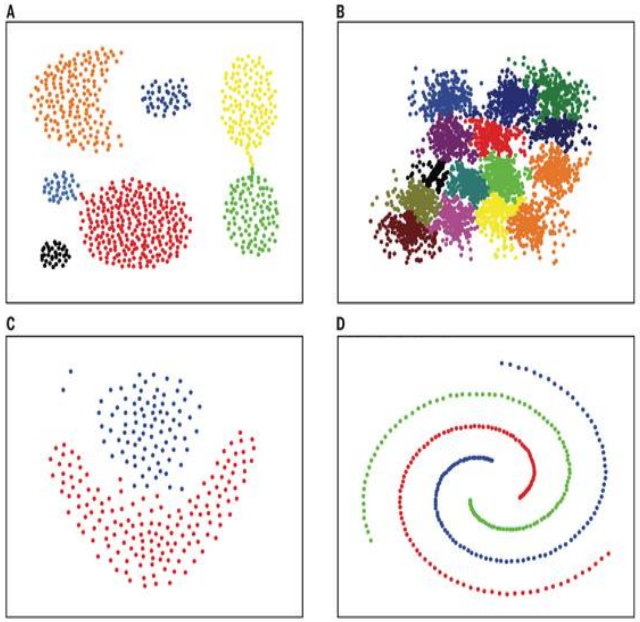
聚类算法

聚类的概念

将物理或抽象对象的集合分成由类似的对象组成的多个类或簇的过程被称为**聚类**。



上图显示了18个点和将它们划分成簇的3种不同方法。该图表明簇的定义是不精确的，而最好的定义依赖于数据的特性和期望的结果。



簇的形象表现在空间分布上不是确定的，而是呈各种不同的形状，在二维平面里就可以有各种不同的形状，如上图所示，在多维空间里有更多的形状。

类的度量方法

距离：度量样品之间的相似性。

常用的距离：

1、闵可夫斯基（Minkowski）距离

$$d_{ij}(q)=\left[\sum_{k=1}^{p}\left|x_{ik}-x_{jk}\right|^{q}\right]^{1/q},\quad i=1,2,\cdots,n;\ j=1,2,\cdots,n$$

2、兰氏（Lance和Williams）距离

$$d_{ik}(L)=\sum_{k=1}^{p}\frac{\left|x_{ik}-x_{jk}\right|}{x_{ij}+x_{jk}},\quad i=1,2,\cdots,n;\ j=1,2,\cdots,n$$

3、马哈拉诺比斯（Mahalanobis）距离

$$d_{ij}^{*}=\left[\frac{1}{p^{2}}\sum_{k=1}^{p}\sum_{l=1}^{p}(x_{ik}-x_{jk})(x_{il}-x_{jl})r_{kl}\right]^{1/2},$$

$$i=1,2,\cdots,n;\ j=1,2,\cdots,n$$

相似系数：度量变量之间的相似性。

常用的相似系数：

1、夹角余弦

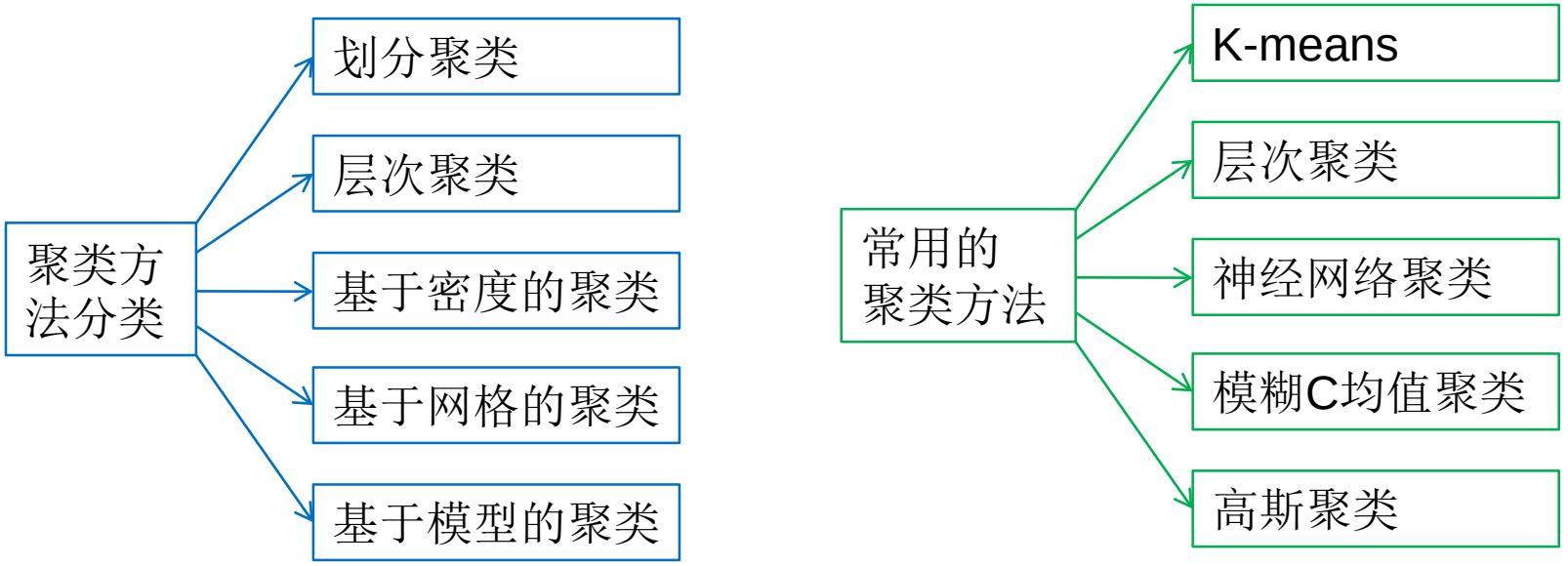
$$C_{ij}(1)=\frac{\sum_{k=1}^{n}x_{ki}x_{kj}}{\left[(\sum_{k=1}^{n}x_{ki}^{2})(\sum_{k=1}^{n}x_{kj}^{2})\right]^{1/2}},\quad i=1,2,\cdots,p;\ j=1,2,\cdots,p$$

2、相关系数

$$C_{ij}(2)=\frac{\sum_{k=1}^{n}(x_{ki}-\bar{x}_{i})(x_{kj}-\bar{x}_{j})}{\sqrt{\left[(\sum_{k=1}^{n}(x_{ki}-\bar{x}_{i})^{2}\right]\left[\sum_{k=1}^{n}(x_{kj}-\bar{x}_{j})^{2}\right]}},$$

$$i=1,2,\cdots,p;\ j=1,2,\cdots,p$$

聚类方法分类


聚类方法分类
划分聚类
层次聚类
基于密度的聚类
基于网格的聚类
基于模型的聚类
常用的聚类方法
K-means
层次聚类
神经网络聚类
模糊C均值聚类
高斯聚类

K-means原理和步骤

工作原理：

首先随机从数据集中选取 K个点，每个点初始地代表每个簇的聚类中心，然后计算剩余各个样本到聚类中心的距离，将它赋给最近的簇，接着重新计算每一簇的平均值，整个过程不断重复，如果相邻两次调整没有明显变化，说明数据聚类形成的簇已经收敛。

算法步骤：

（1）从n个数据对象任意选择k个对象作为初始聚类中心。

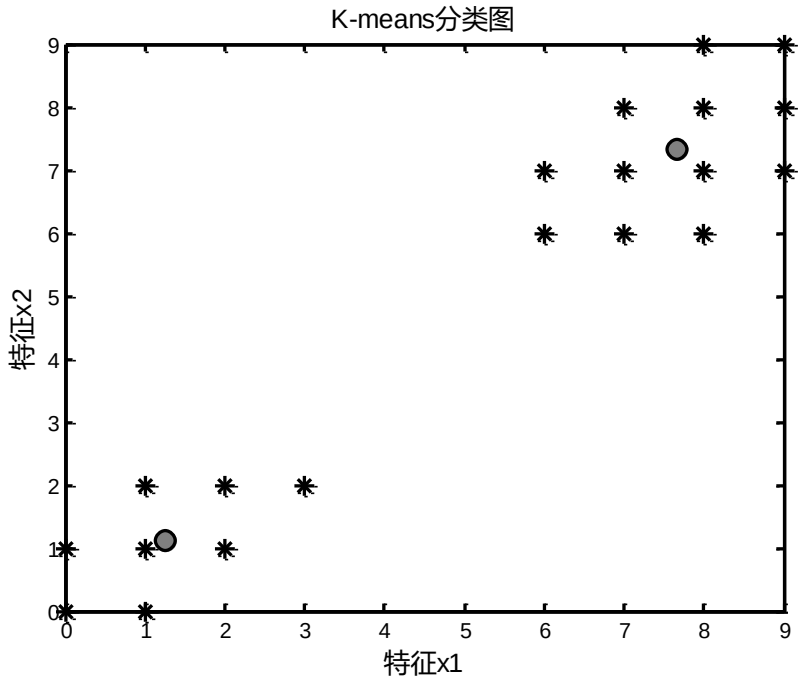
（2）循环第（3）步到第（4）步直到每个聚类不再发生变化为止。

（3）根据每个聚类对象的均值（中心对象），计算每个对象与这些中心对象的距离；并根据最小距离重新对相应对象进行划分。

（4）重新计算每个聚类的均值（中心对象），直到聚类中心不再变化。这种划分使得下式最小：

$$E = \sum_{j=1}^{k} \sum_{x_i \in \omega_j} \left\| x_i - m_j \right\|^2$$

K-means实例1：自主编程



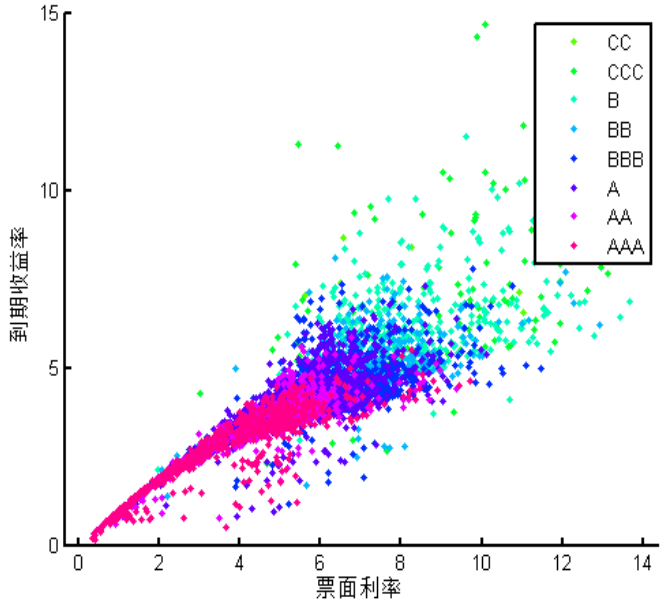
上图为对下表20个数据进行聚类的结果

X1	0	1	0	1	2	1	2	3	6	7
X2	0	0	1	1	1	2	2	2	6	6
X1	8	6	7	8	9	7	8	9	8	9
X2	6	7	7	7	7	8	8	8	9	9

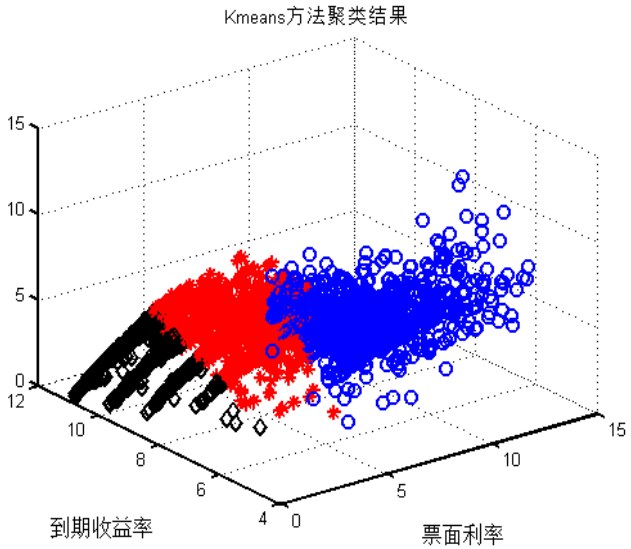
从图中可以看出，聚类的效果是非常显著的。

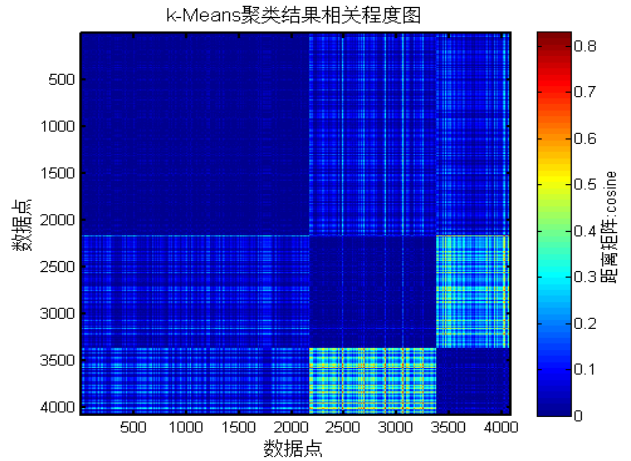
K-means实例2：内置函数

Matlab函数：
kmeans



K-means

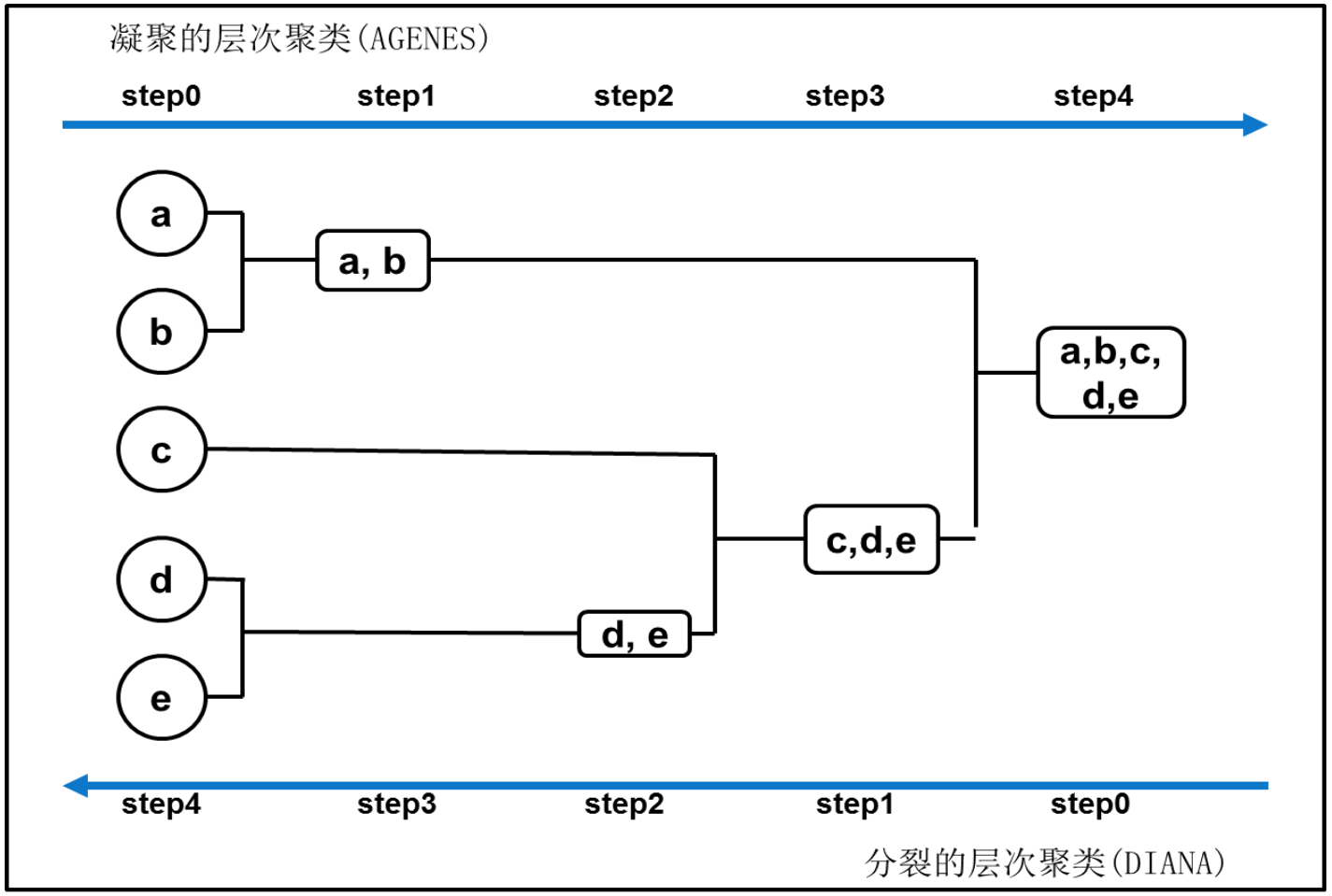




上图为实例的数据分布图。
右上图为K-means方法聚类效果图；右下图为簇间相似度矩阵。

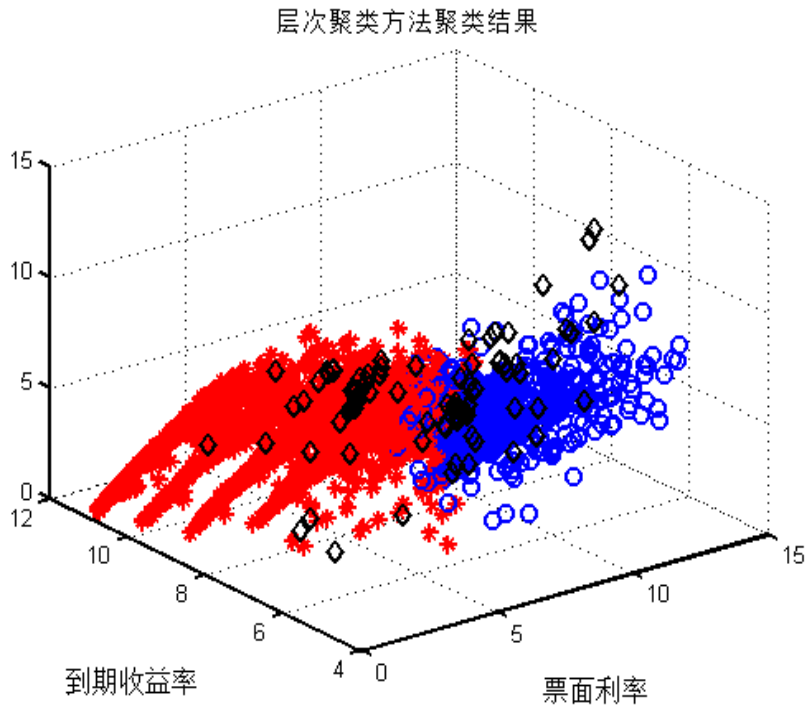
层次聚类原理和步骤

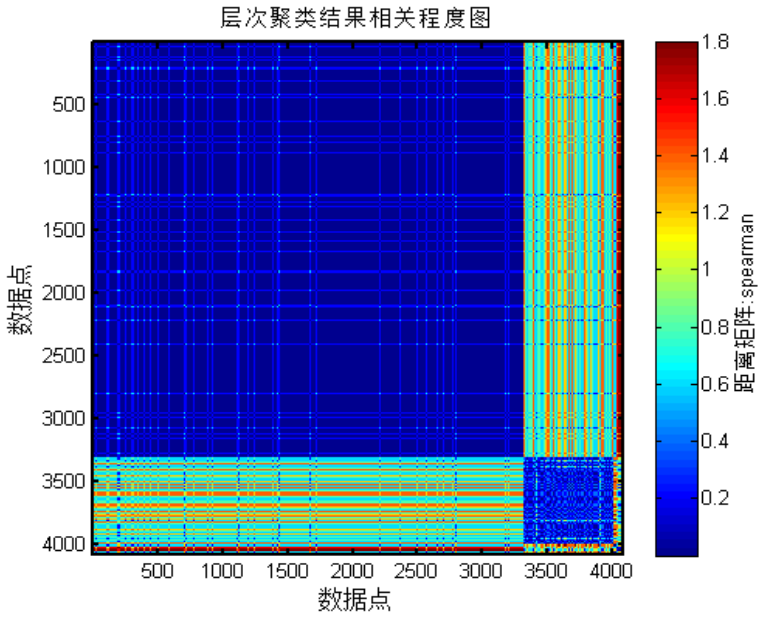
层次聚类算法是通过将数据组织为若干组并形成一个相应的树来进行聚类的。在实际应用中一般有两种层次聚类方法。上图描述了一个凝聚层次聚类方法AGENES和一个分裂层次聚类方法DIANA在一个包括五个对象的数据的集合{a,b,c,d,e}上的处理的过程。

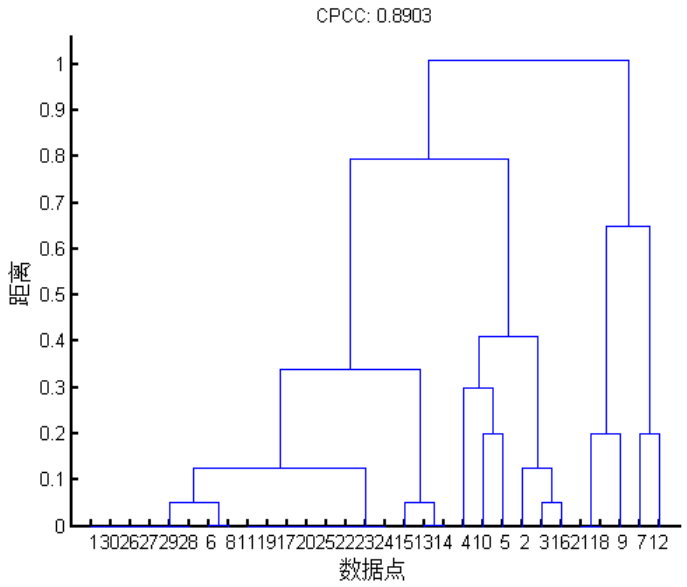


层次聚类实例

Matlab函数：
clusterdata



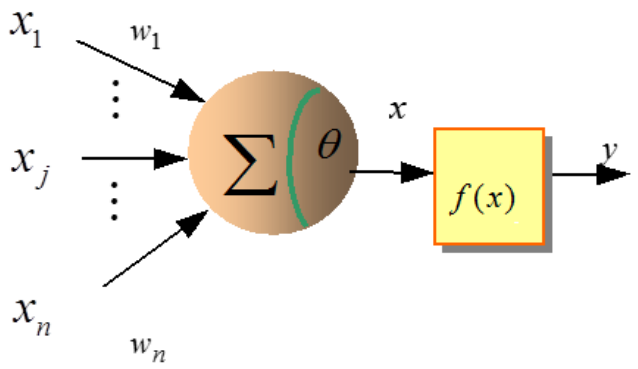




R2016b

神经网络聚类原理和步骤

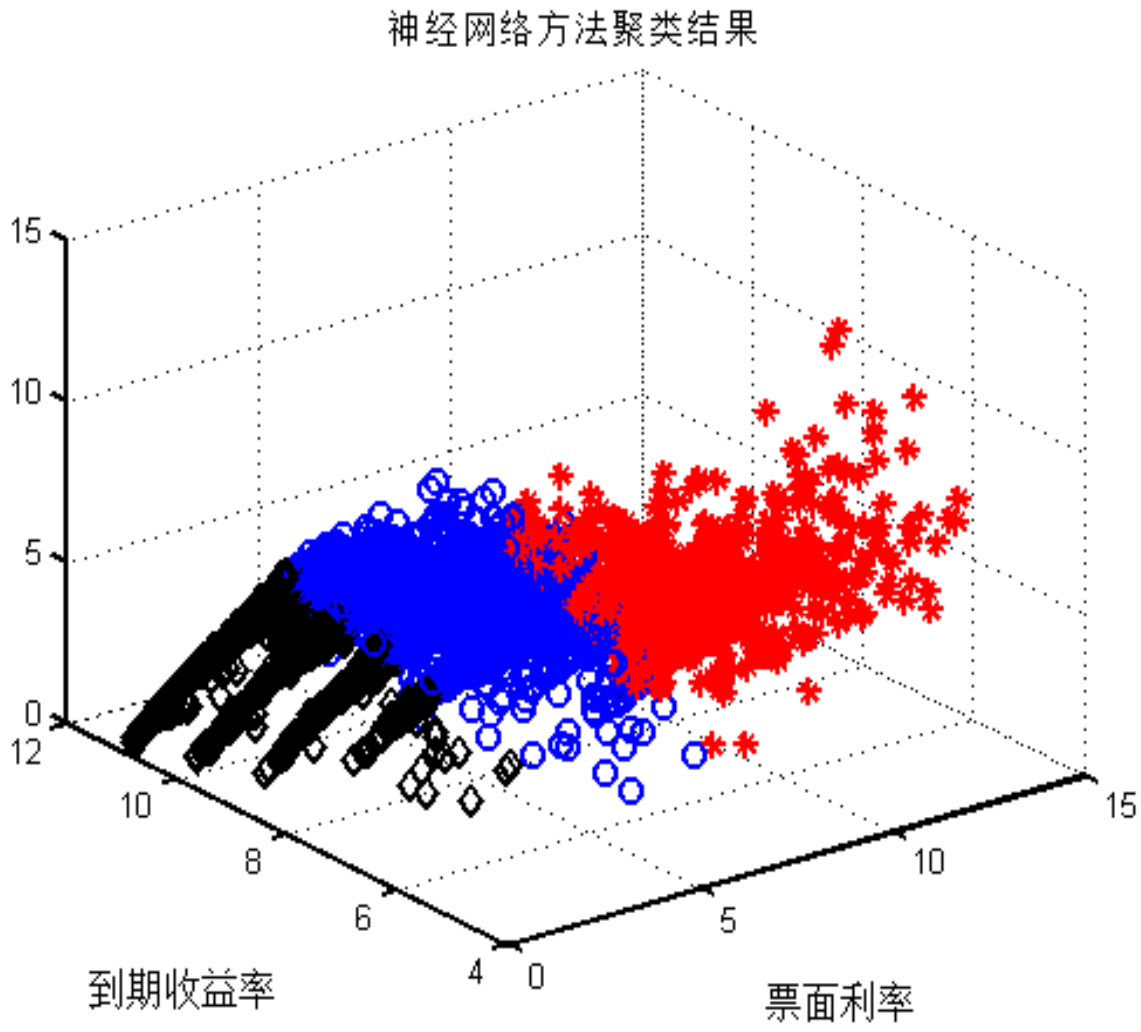
人工神经网络（Artificial Neural Networks，ANN）是一种应用类似于大脑神经突触联接的结构进行信息处理的数学模型。



左图展示了一个简单的神经网络结构——感知器。感知器包含两种结点：几个输入结点，用来表示输入属性；一个输出结点，用来提供模型输出。神经网络结构中的结点通常叫作神经元或单元。在感知器中，每个输入结点都通过一个加权的链连接到输出结点。这个加权的链用来模拟神经元间神经键连接的强度。像生物神经系统一样，训练一个感知器模型就相当于不断调整链的权值，直到能拟合训练数据的输入输出关系为止。

神经网络聚类实例

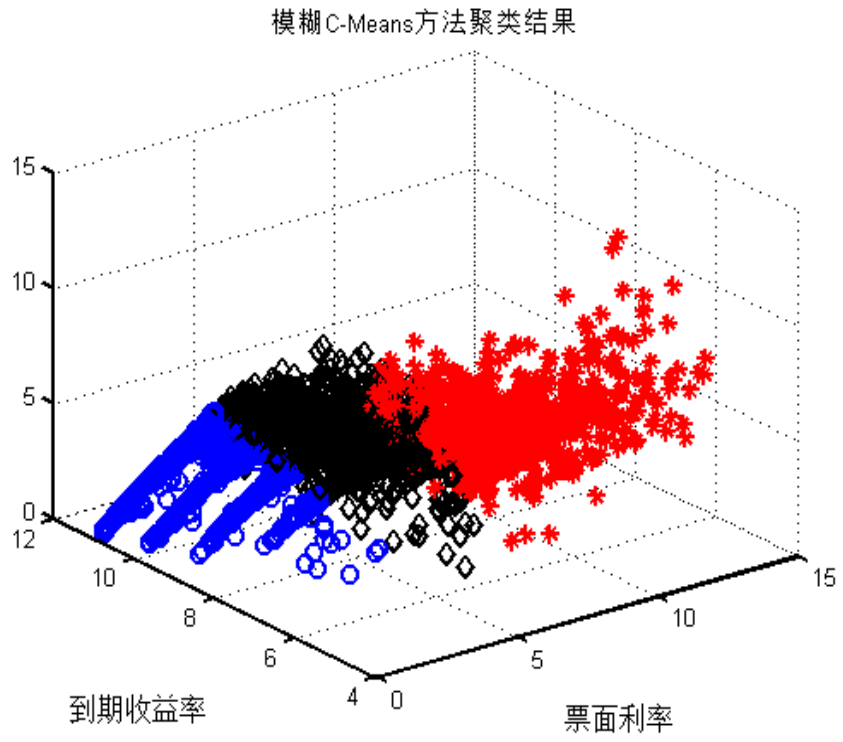
Matlab函数：
train



FCM原理和步骤

模糊C均值聚类算法是用隶属度确定每个数据点属于某个聚类的程度的一种聚类算法。

Matlab函数：
fcm

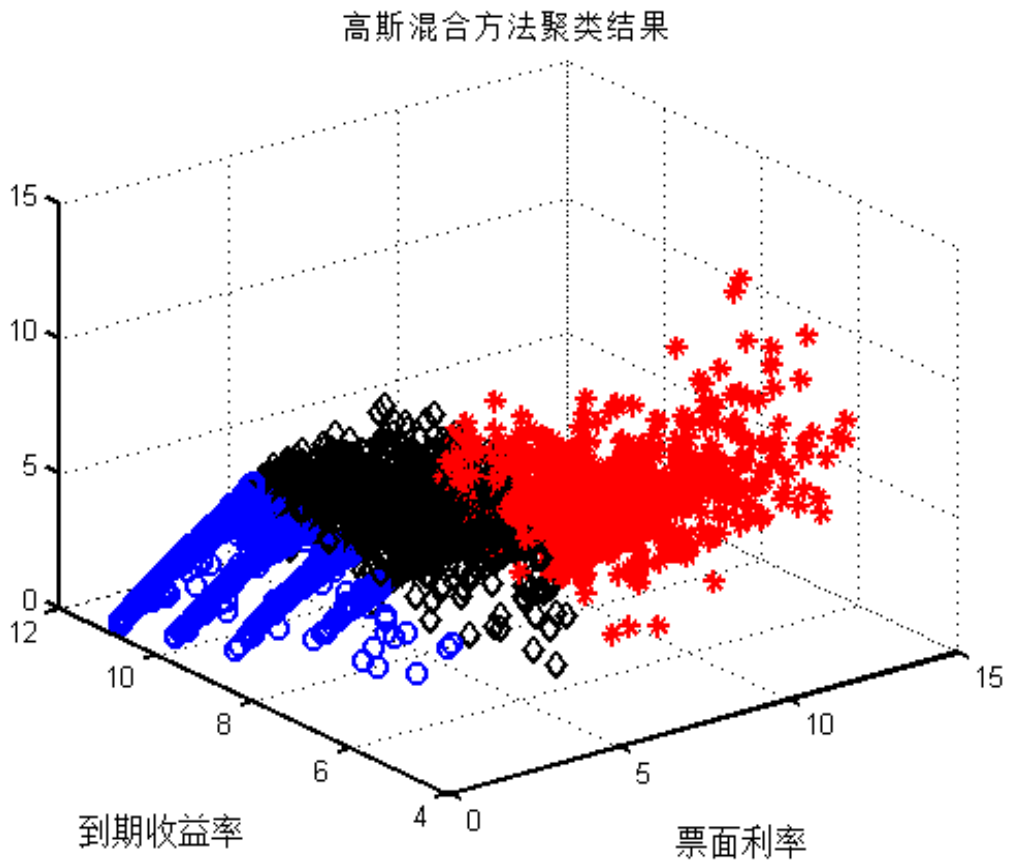


高斯混合聚类原理和步骤

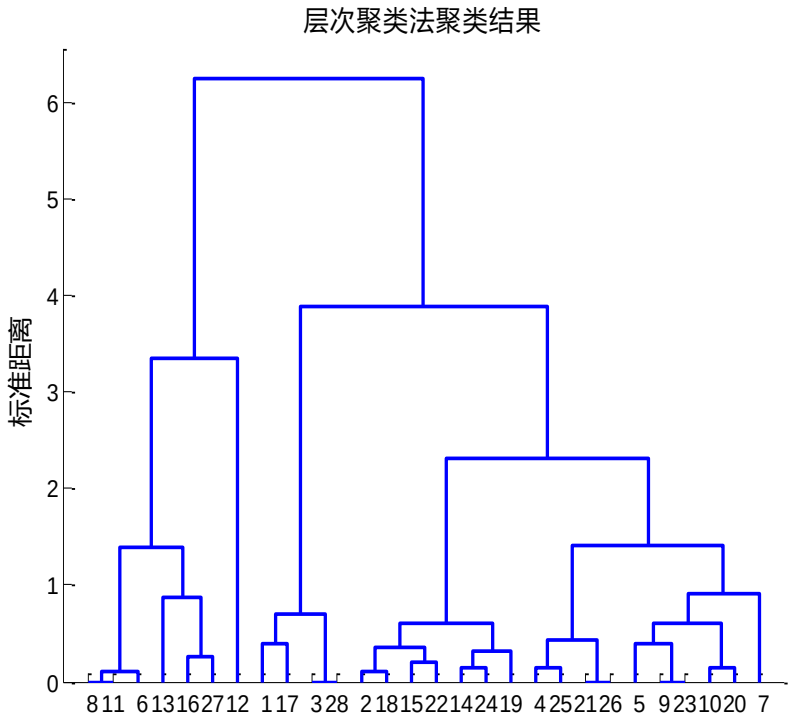
混合高斯模型的定义为：

$$p(x)=\sum_{k=1}^{K}\pi_k\,p(x\mid k)$$

Matlab函数：
gmdistribution.fit



“类”的确定方法



聚类过程中类别个数的确定方法：

（1）阈值法

（2）轮廓图法

轮廓图可由MATLAB中的Silhouette函数来绘制，此函数可以用来根据Cluster、Clusterdata、K-means的聚类结果绘制轮廓图，从图中可以判断每个点的分类是否合理。轮廓图上第i点的轮廓值定义为：

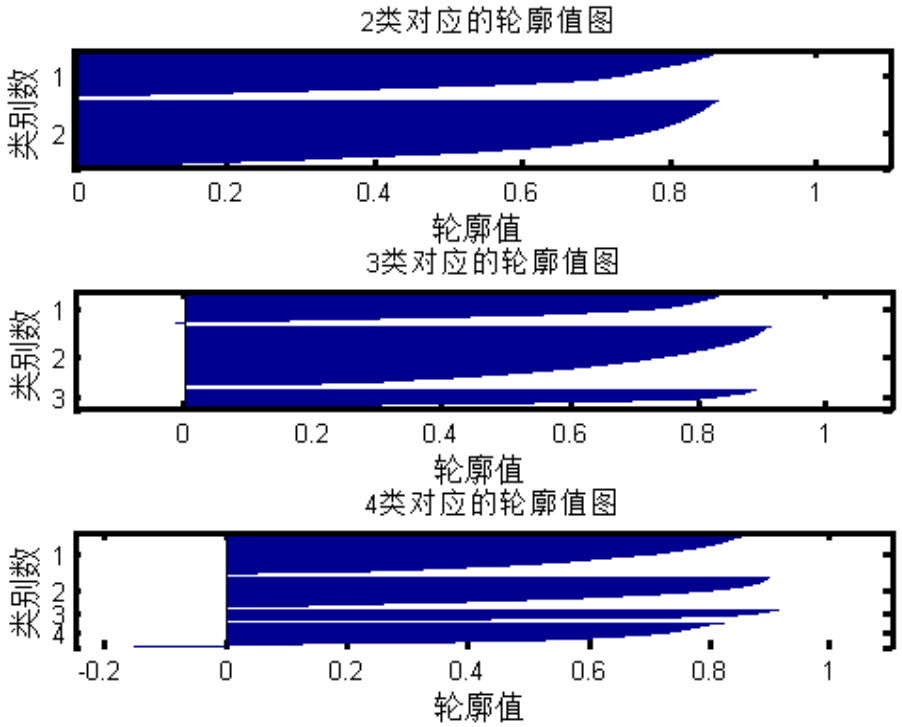
$$S(i) = \frac{\min(b) - a}{\max[a, \min(b)]}, i = 1, \cdots, n$$

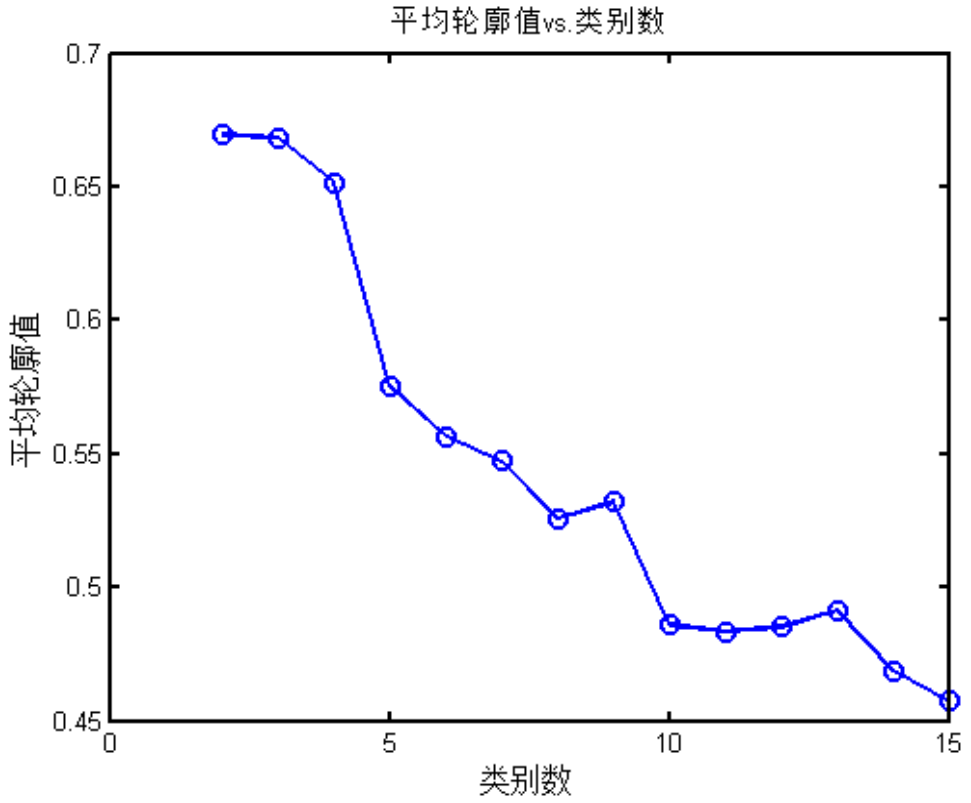
其中，a是第i个点与同类其他点的平均距离。b是向量，其元素表示第i个点与不同类的类内各点的平均距离。

S(i)的取值范围[-1，1]，此值越大，说明该点的分类越合理。特别当S(i)<0时，说明该点分类不合理。

“类”的确定实例

利用K-means方法和轮廓图法确定最佳的聚类类别数。





聚类目标和数据描述

聚类目标：进行股票的分池。

数据描述：

通常选择周期相对较长的指标作为股票的聚类指标，股票基本面的指标基本都是这类指标，所以选择这类指标比较合适。现在就以第3章获取的股票财务数据为基础，来研究股票的聚类问题。

对财务数据进行质量分析发现，指标X4至X13这10个指标的数据质量相对较好，且有明确意义，所以打算以这些指标对股票进行聚类。

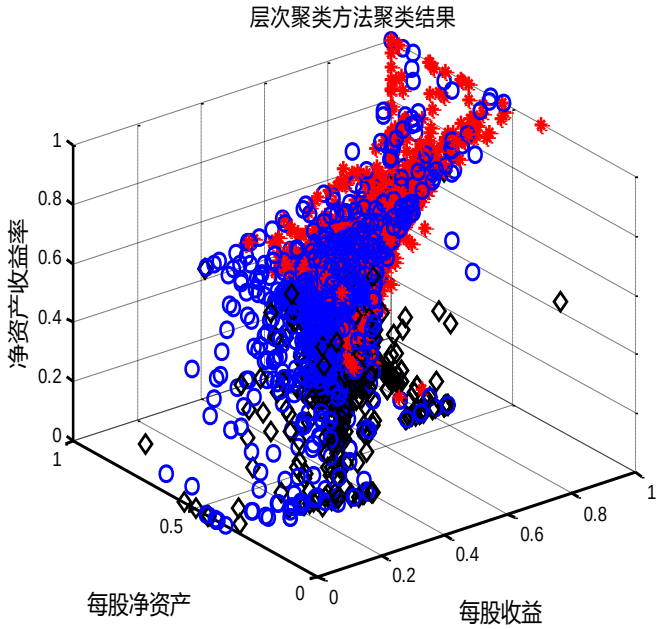
实现过程

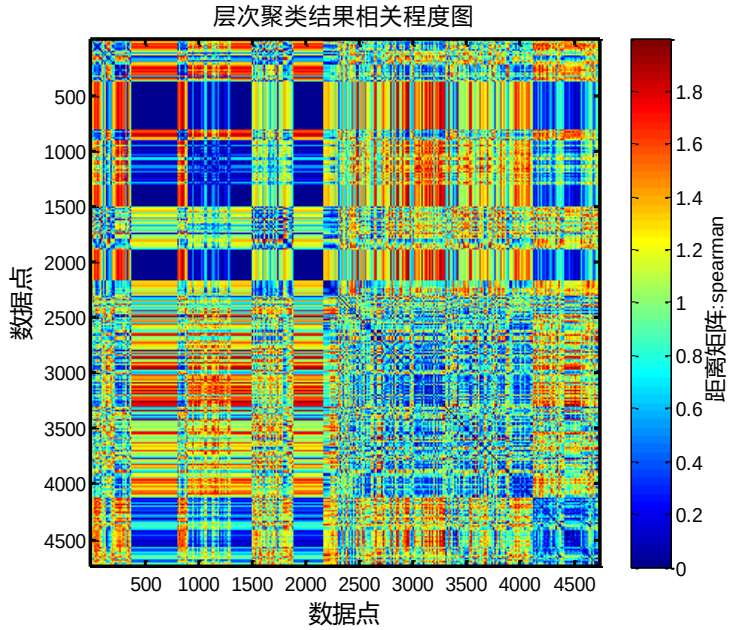
（1）确定选择哪个聚类算法

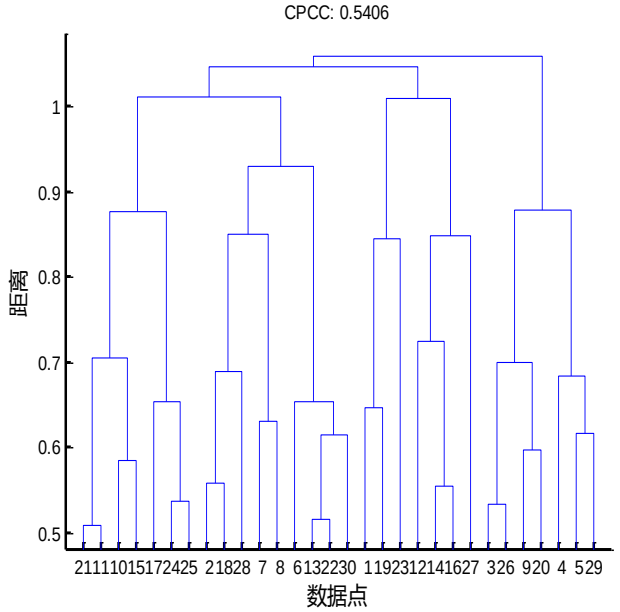
以上介绍的几个算法都可以，但我们想更清楚地了解这些股票的层级结构，所以选择层次分析法比较合适。

（2）编写代码

在以上介绍的层次聚类法代码的基础上略作修改，可以得到实现股票聚类的代码，具体代码见P9-2。







MATLAB 学习资源

- **www.mathworks.com**
  - 录制的讲座
  - 行业解决方案
  - MATLAB central
- **www.ilovematlab.cn**
  - 问题交流
  - 图书

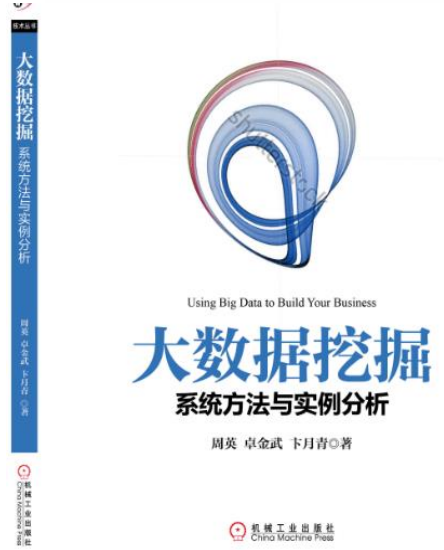
    《大数据挖掘：系统方法与实例分析》

- 购买正版**MATLAB**

  电话：010-59827000

- 答疑方式

  邮箱： 70263215@qq.com



实践与资源

第1讲：数据与程序 http://pan.baidu.com/s/1boGzSwn

第2讲：数据与程序 http://pan.baidu.com/s/1dELf87f

第3讲：数据与程序 http://pan.baidu.com/s/1c1Fcu5M

谢谢大家！