北京郵電大學
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

Data Analytics with R
语言数据分析

方以类聚、物以群分

艾新波 / 2018 • 北京

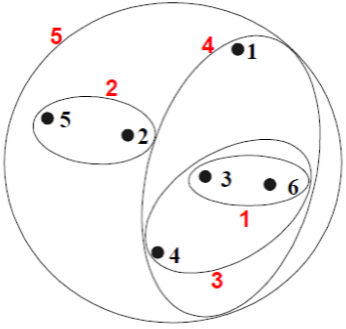
课程体系

- R语言数据分析
  - 上部：论道 
    - 第1章 气象万千、数以等观
    - 第2章 所谓学习、归类而已
    - 第3章 格言联璧话学习
    - 第4章 源于数学、归于工程
  - 中部：执具 
    - 第5章 工欲善其事必先利其器
    - 第6章 基础编程
    - 第7章 数据对象
  -
    - 第8章 人人都爱tidyverse
    - 第9章 最美不过数据框
  - 下部 博术 
    - 第10章 观数以形
    - 第11章 相随相伴、谓之关联
    - 第12章 既是世间法、自当有分别
    - 第13章 方以类聚、物以群分
    - 第14章 庐山烟雨浙江潮

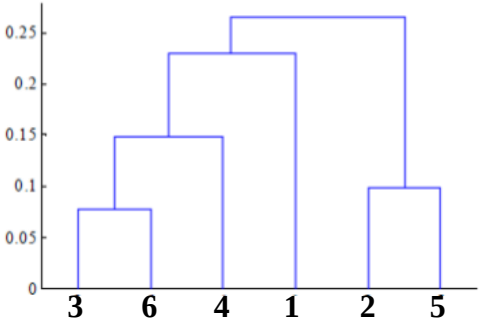
层次聚类基本原理

层次聚类hierarchical clustering试图在不同层次上对数据集进行划分

通过树状图dendrogram（又称谱系图）来表征对象的远近关系



嵌套簇图



树状图

层次聚类基本原理

基本凝聚层次聚类算法：

(1) 计算邻近性矩阵

(2) repeat

(3) 合并最接近的两个簇

(4) 更新邻近性矩阵，以反映新的簇与原来簇之间的邻近性

(5) until 仅剩下一个簇

类间距离

层次聚类关注的是簇之间的距离。簇之间的距离由其所包含的点所定义：

最小距离： $dist_{\min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \{|p - p'|\}$

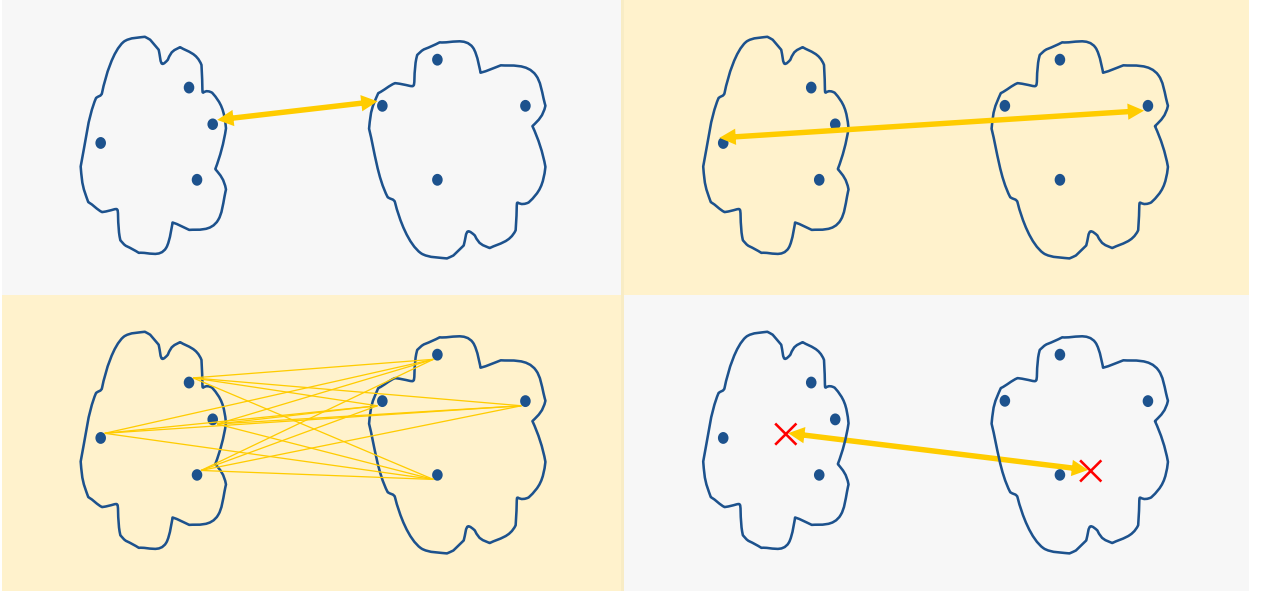
最大距离： $dist_{\max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \{|p - p'|\}$

平均距离： $dist_{\min}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i, p' \in C_j} |p - p'|$

均值距离： $dist_{\min}(C_i, C_j) = |m_i - m_j|$

其中： $|p - p'|$**是两个对象或点之间的距离；** m_i**是簇** C_i**的均值**

类间距离



算法实现：常用的包及函数

CRAN Task View:

Cluster Analysis & Finite Mixture Models

Functions **hclust**() from package stats and **agnes**() from cluster **are the primary functions for agglomerative hierarchical clustering**, function diana() can be used for divisive hierarchical clustering.

The dendextend package provides functions for easy visualization, manipulation and comparison of dendrograms

算法实现：hclust

```
#为便于演示，选出10名同学进行聚类
selected_students <- c(
  "伊礼贤", "鲁孟秋", "焦金音", "宁琦", "赖旺",
  "于知平", "方顺", "谭思缘", "僪福星", "尚玉芳")
scores <- cjb %>%
  filter(xm %in% selected_students) %>%
  select(xm, yw:sw) %>%
  column_to_rownames(var = "xm")#带行名的数据框
demo_dist <- dist(scores)#计算距离矩阵
imodel <- hclust(demo_dist) #利用hclust进行聚类
```

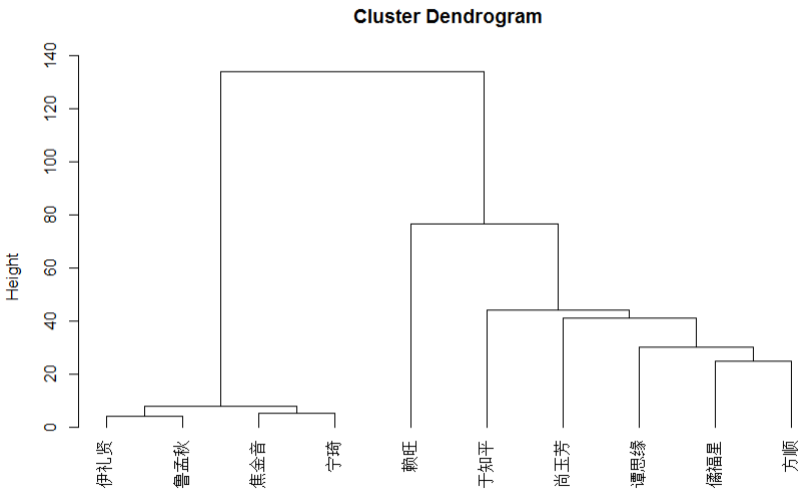
算法实现：建模结果

```
imodel
#> Call:
#>   hclust(d = demo_dist)
#>
#> Cluster method   : complete
#> Distance         : euclidean
#> Number of objects: 10
names(imodel)
#> [1] "merge"       "height"      "order"       "labels"
#> [5] "method"      "call"        "dist.method"
```

算法实现：建模结果

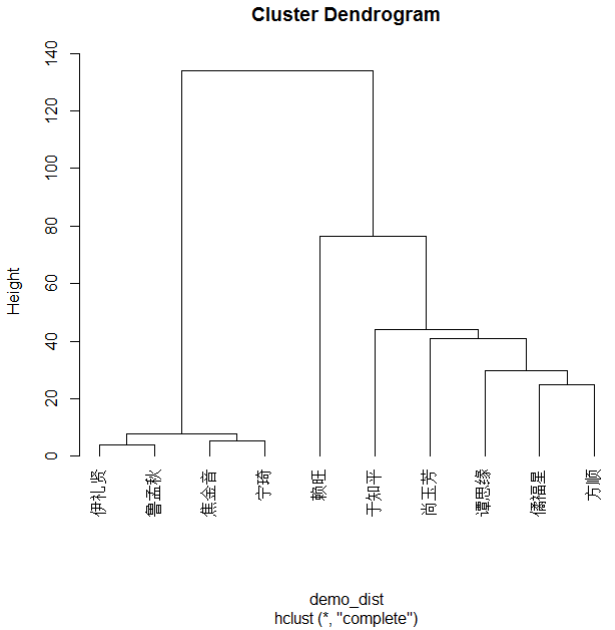
```
imodel$merge
#>      [,1] [,2]
#> [1,]   -7   -8
#> [2,]   -6   -9
#> [3,]    1    2
#> [4,]   -2  -10
#> [5,]   -3    4
#> [6,]   -5    5
#> [7,]   -1    6
#> [8,]   -4    7
#> [9,]    3    8
```

```
[1] "于知平" "儁福星" "谭思缘" "赖旺"   "尚玉芳"
[6] "焦金音" "伊礼贤" "鲁孟秋" "宁琦"   "方顺"
```



算法实现：建模结果

```
imodel$height
#> [1]   4.000000
#> [2]   5.291503
#> [3]   7.937254
#> [4]  24.819347
#> [5]  29.933259
#> [6]  41.073106
#> [7]  44.068129
#> [8]  76.360985
#> [9] 134.000000
```



算法实现：建模结果

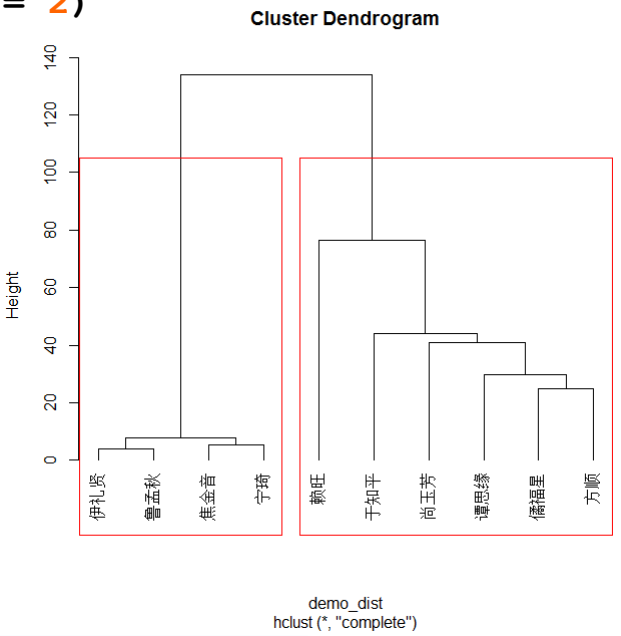
```
imodel$height
#> [1]   4.000000   5.291503   7.937254  24.819347
#> [5]  29.933259  41.073106  44.068129  76.360985
#> [9] 134.000000
sort(dist(scores))
#> [1]   4.000000   5.196152   5.291503   5.567764
#> [5]   7.000000   7.937254  24.819347  26.888659
#> [33]  91.021975  91.656969  92.238820  92.293012
#> [37]  93.616238 100.682670 100.935623 101.113797
#> [41] 102.815369 132.461315 133.540256 133.787144
#> [45] 134.000000
```

算法实现：建模结果

```
imodel$order
#> [1]  7  8  6  9  4  1  5  3  2 10
imodel$labels
#> [1] "于知平" "僪福星" "谭思缘" "赖旺"   "尚玉芳"
#> [6] "焦金音" "伊礼贤" "鲁孟秋" "宁琦"   "方顺"
imodel$method
#>[1] "complete"
imodel$call
#>hclust(d = demo_dist)
imodel$dist.method
#>[1] "euclidean"
```

算法实现：类别划分

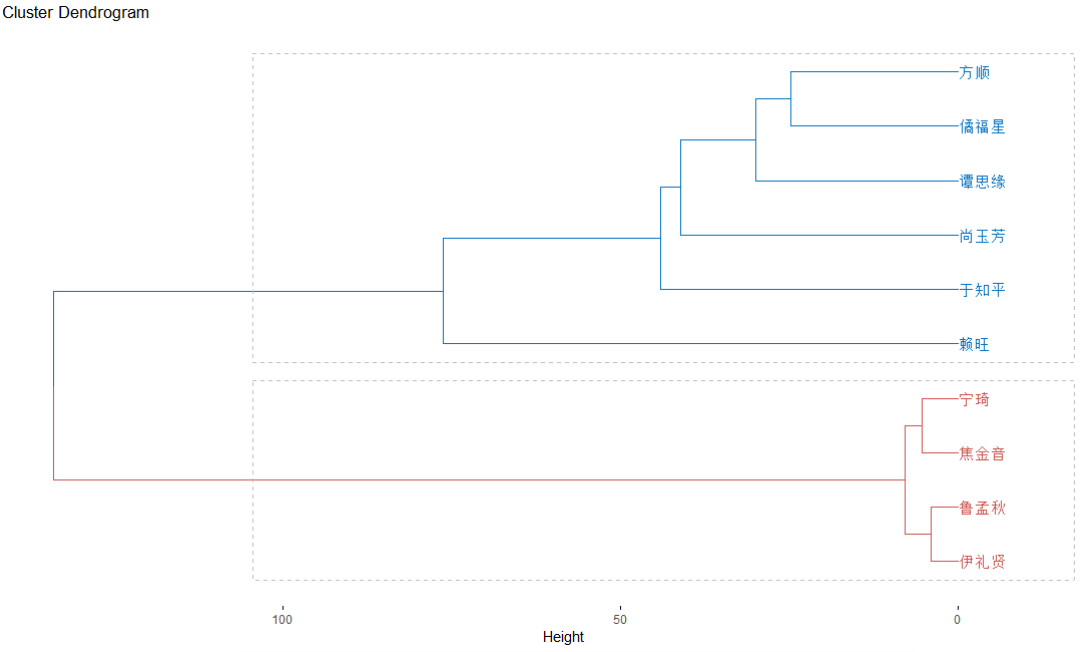
```
cluster_idx <- cutree(imodel, k = 2)
#> 于知平 傅福星 谭思缘
#> 1      1      1
#> 赖旺 尚玉芳 焦金音
#> 1      1      2
#> 伊礼贤 鲁孟秋 宁琦
#> 2      2      2
#> 方顺
#> 1
plot(imodel, hang = -1)
rect.hclust(imodel, k = 2)
```



算法实现：更漂亮一点的谱系图

```
library(factoextra)
res <- hcut(
  dist(scores), k = 2,
  hc_func = "hclust", hc_method = "complete",
  hc_metric = "euclidean", stand = FALSE,
  graph = FALSE)
fviz_dend(
  res, rect = TRUE, cex = 0.75,
  horiz = TRUE, type = "rectangle",
  k_colors = c("#CD534CFF","#0073C2FF"))
```

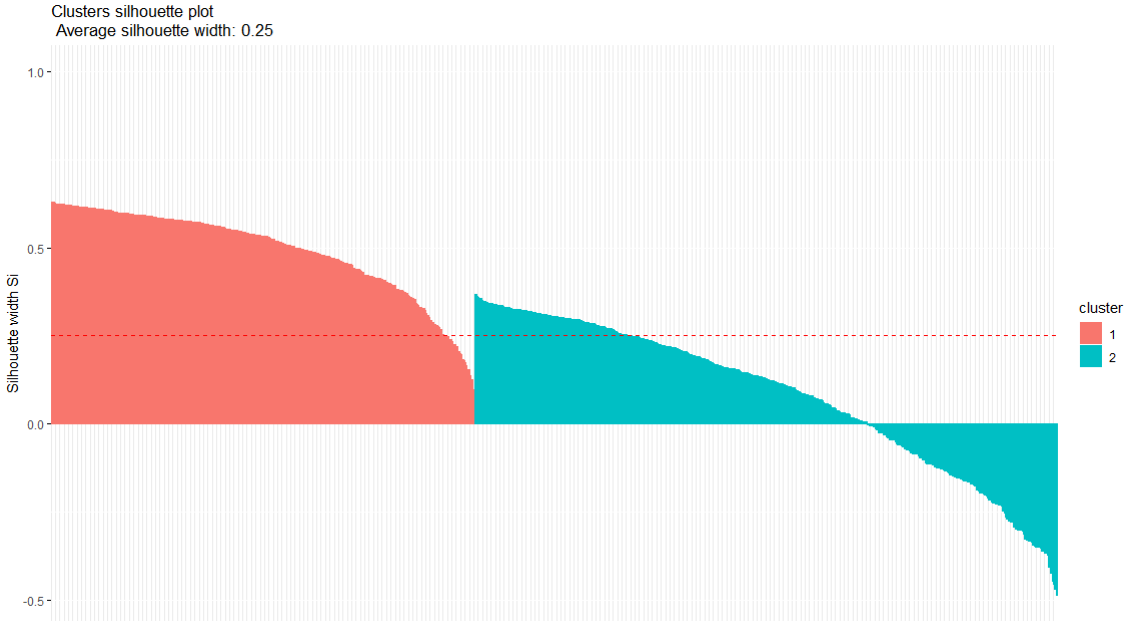
算法实现：更漂亮一点的谱系图



算法实现：模型评估

```
require(cluster)
scores <- cjb %>%
  select(yw:sw)
scores_dist <- dist(scores)
imodel <- hclust(scores_dist, method = "ward.D")
cluster_idx <- cutree(imodel, k = 2)
#计算轮廓系数
kmeans_k2_silhouette <- silhouette(cluster_idx, scores_dist)
#绘制轮廓系数
fviz_silhouette(kmeans_k2_silhouette)
```

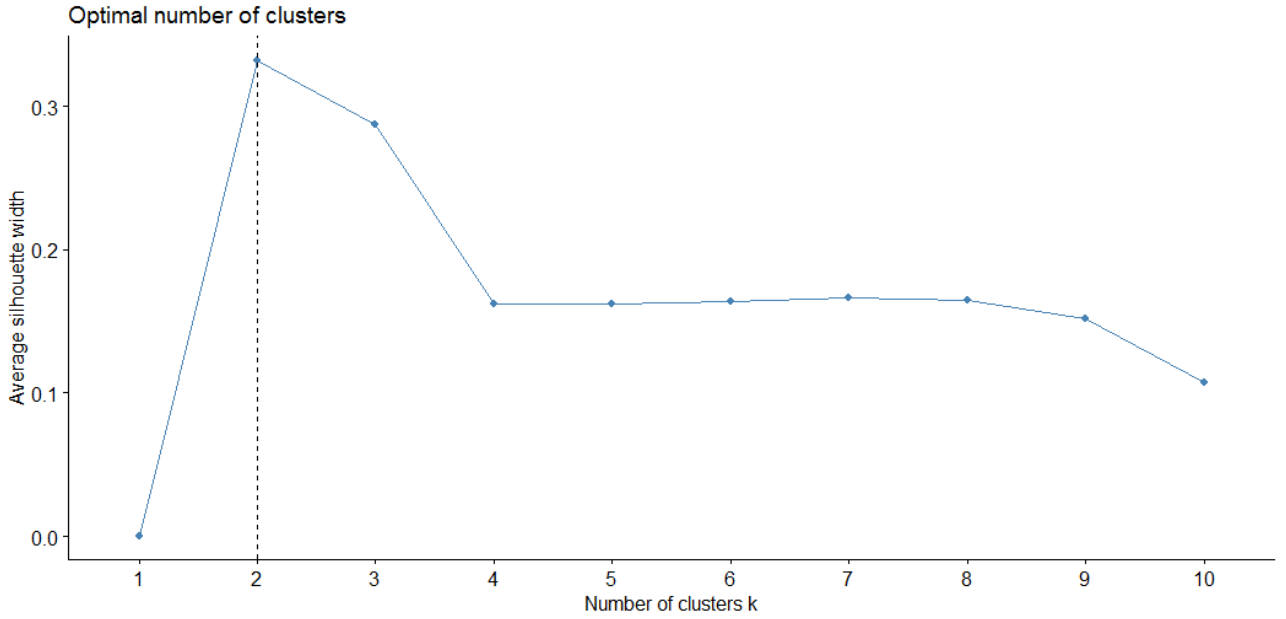
算法实现：模型评估


Clusters silhouette plot
Average silhouette width: 0.25
Silhouette width Si
1.0
0.5
0.0
-0.5
cluster
1
2

算法实现：模型评估

```
#选取最佳的簇数
library(factoextra)
fviz_nbclust(scores,
             FUNcluster = hcut,
             method = "silhouette",
             kmax = 20) +
  geom_vline(xintercept = 2, linetype = 2)
```

算法实现：模型评估


Optimal number of clusters
Average silhouette width
0.0
0.1
0.2
0.3
1
2
3
4
5
6
7
8
9
10
Number of clusters k

算法实现：与实际类标签的比较

```
imodel <- hclust(scores_dist, method = "ward.D")
cluster_idx <- cutree(imodel, k = 2)
(ic_metric <- min(Metrics::ce(
  cjb$wlfk,
  c( "理科", "文科")[cluster_idx]),
  1- Metrics::ce(
    cjb$wlfk,
    c( "理科", "文科")[cluster_idx])))
#> [1] 0.2860892
```

尝试一些学术创新



创新 ≈ 杂交 / 嫁接 / 混血 / 跨界 /

混搭 / 学科交叉 / 结合 /

遇见 / 阴阳相交 / ……

尝试一些学术创新

从某种意义上来看，世间一切，都是遇见。就像：

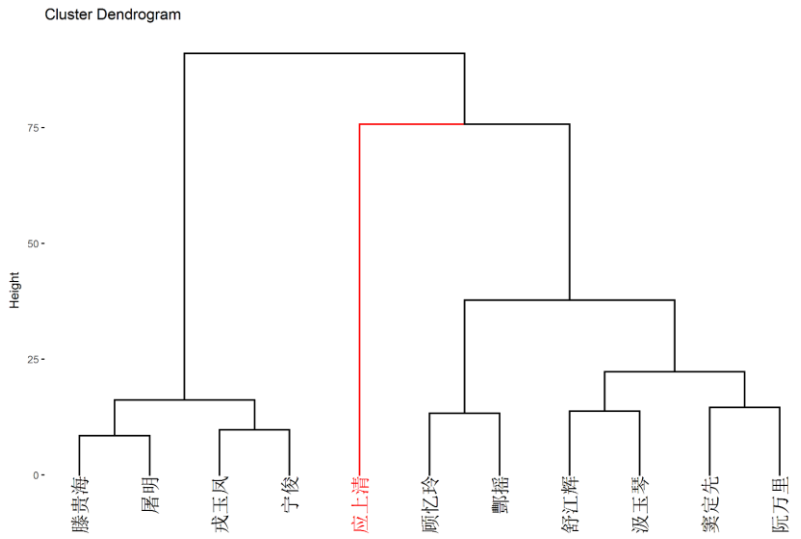
冷遇见暖，就有了雨；

春遇见冬，有了岁月；

天遇见地，有了永恒；

人遇见人，有了生命。

与异常点检测进行交叉



Resource-bounded Fraud Detection

Luis Torgo

LIAAD-INESC Porto LA / FEP, University of Porto
R. de Ceuta, 118, 6., 4050-190 Porto, Portugal
ltorgo@liaad.up.pt – http://www.liaad.up.pt/~ltorgo

Abstract. This paper describes an approach to fraud detection targeted at applications where this task is followed by a posterior human analysis of the signaled frauds. This is a frequent setup on fraud detection applications (e.g. credit card misuse, telecom fraud, etc.). In real world applications this human inspection is usually constrained by limited resources. In this context, standard fraud detection methods that simply tag each case as being (or not) a possible fraud are not very useful if the number of tagged cases surpasses the available resources. A much more useful approach is to produce a ranking of fraud that can be used to optimize the available inspection resources by first addressing the cases with higher rank. In this paper we propose a method that produces such ranking. The method is based on the output of standard agglomerative hierarchical clustering algorithms, resulting in no significant additional computational costs. Our comparisons with a state of the art method provide convincing evidence of the competitiveness of our proposal.

层次聚类应用于异常检测

层次分析法给我们的直觉：离群值不易于合并，即当他们最终被合并的时候，他们合并前所属类的大小和他们被合并进去的类的大小相差应该很大。大多数情况下会在聚类的后期进行合并，通常是与一个更大的类进行合并

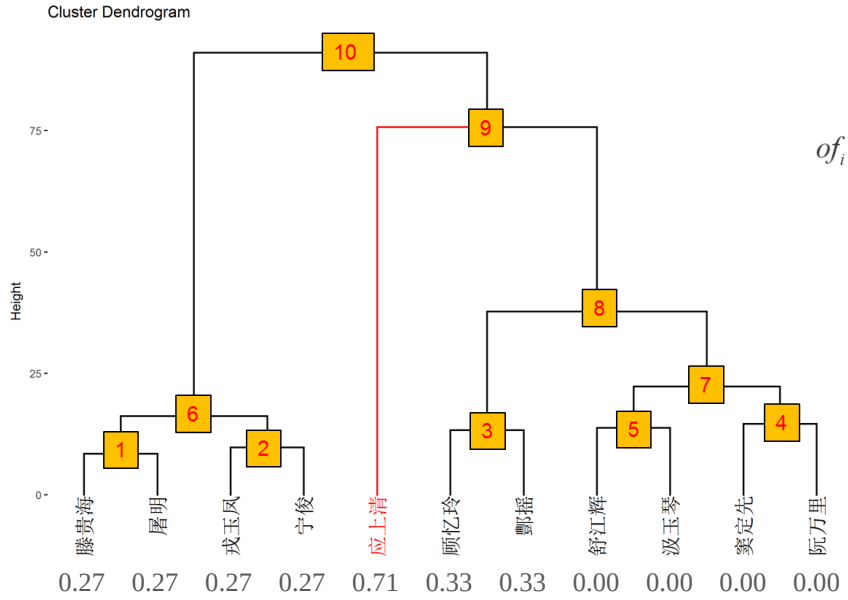
转换成数学语言：若对象x参与第i 次合并，令离群分值为：

$$of_i(x) = \max\left(0, \frac{|g_{y,i}| - |g_{x,i}|}{|g_{y,i}| + |g_{x,i}|}\right)$$

对象x最终的离群值为：

$$OF_H(x) = \max_i\left(of_i(x)\right)$$

层次聚类应用于异常检测



$$of_i(x) = \max\left(0, \frac{|g_{y,i}| - |g_{x,i}|}{|g_{y,i}| + |g_{x,i}|}\right)$$

层次聚类应用于异常检测

```
library(DMwR)
out_rank <- outliers.ranking(scores_dist,
  clus = list(dist = "euclidean",
    alg = "hclust", meth = "ward.D"))
cjb %>%
  arrange(desc(out_rank$prob.outliers)) %>%
  View()
#与箱线图异常值检测作比较
(outliers <- boxplot.stats(cjb$zcj)$out)
outliers_idx <- which(cjb$zcj %in% outliers)
View(cjb[outliers_idx, ])
```

异常检测结果的对照

通过总成绩箱线图分析异常值:

姓名	班级	性别	语文	数学	外语	政治	历史	地理	物理	化学	生物	文理分科	总成绩
张良平	115	男	0	0	0	0	0	0	0	0	0	理科	0
滑亚	113	男	33	46	30	65	65	76	56	76	59	理科	523
赖旺	103	男	65	26	53	87	91	96	21	56	58	文科	553
舒茂	113	男	66	58	73	67	67	80	51	80	72	理科	605
于知平	101	男	70	67	74	92	73	88	40	52	58	文科	614
方顺	114	男	77	62	78	77	77	80	50	60	70	文科	621

异常检测结果的对照

基于层次聚类的方法进行异常检测：

姓名	性别	语文	数学	外语	政治	历史	地理	物理	化学	生物	文理分科	总成绩	异常系数
张良平	男	0	0	0	0	0	0	0	0	0	理科	0	0.97
成朝龙	男	84	96	91	91	0	96	89	98	91	理科	736	0.95
赖旺	男	65	26	53	87	91	96	21	56	58	文科	553	0.91
滑亚	男	33	46	30	65	65	76	56	76	59	理科	523	0.91
吕秀芳	女	81	95	88	86	68	96	88	92	85	理科	779	0.78
彭书弼	男	86	93	85	87	81	84	47	98	90	理科	751	0.76

谢谢聆听

Thank you

教师个人联系方式

艾新波

手机：13641159546

QQ： 23127789

微信：13641159546

E-mail: 13641159546@126.com

axb@bupt.edu.cn

地址：北京邮电大学科研楼917室

课程网址：https://github.com/byaxb/RDataAnalytics