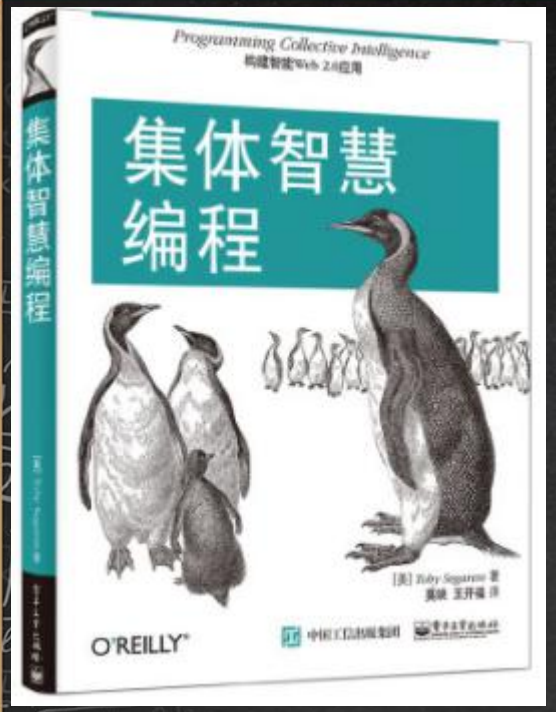


发现群组

小白分类——无监督学习的聚类分析

祥水村东头@AI识堂

《集体智慧编程》读书笔记系列(对应第3章 pp29-53)

本次胶片内容、及涉及相关代码均可移步至Github进行下载

我的代码 Github 地址：
https://github.com/david-cal/
Reading-Note-For-Programming-Collective-Intelligence

目录

何为聚类？
监督学习 与 无监督学习 (p29-30)

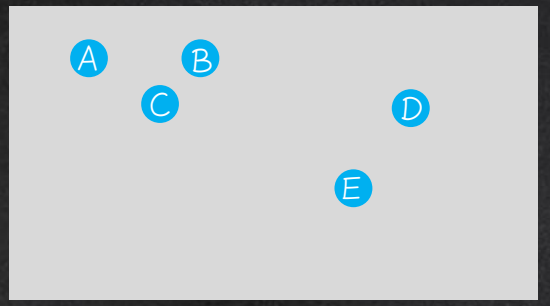
✓ 监督学习（supervised learning）

- 利用样本输入和期望输出学习如何预测的技术
- 包括神经网络、决策树、支持向量机、贝叶斯过滤

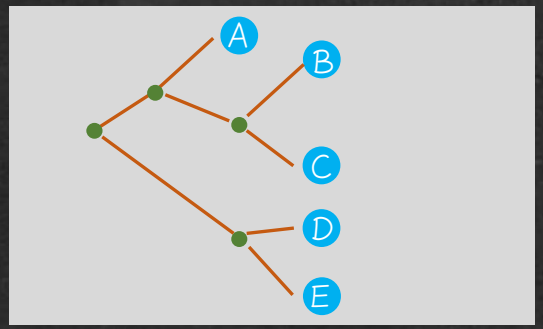
✓ **聚类/无监督学习（unsupervised learning）**

- 不是利用带有正确答案的样本进行训练，而是在一组样本中直接找到某种分类结构
- 包括k-means、 K-medoids、CLARANS、BIRCH算法、DBSCAN算法等

如何对客观事物进行归类？

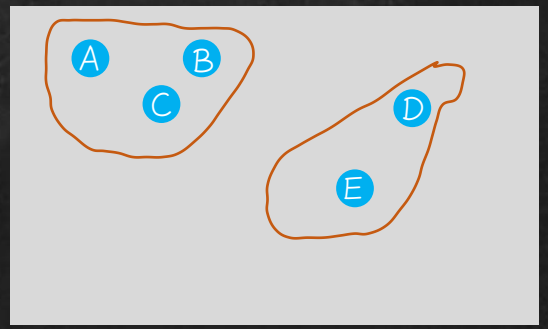


思路1：
自上而下/自下而上逐层级归类
可表达出各样本间的关系



分级聚类

思路2：
直接分割归类
可表达出聚类数目



分割聚类

01

分级聚类

- “分类树” = “枝节点” + “叶节点”
- 基于文本进行聚类分析实践

-实践案例1：基于教材RSS源英文文本的聚类

-实践案例2：基于知乎RSS源中文文本的聚类

➢ 图解层级聚类

定义距离 → 两两距离 → 两两合并

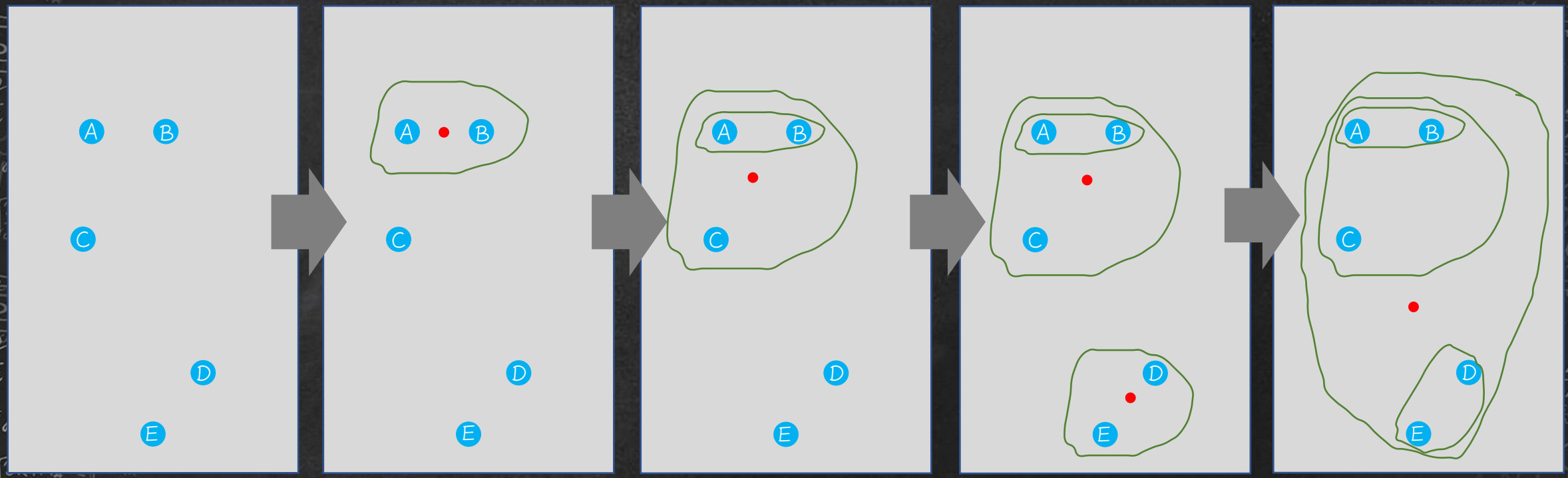
初始状态下
将每个样本视为
单独的一类

计算两两样本之间的距离
B与A间距离最近
合并为(AB)

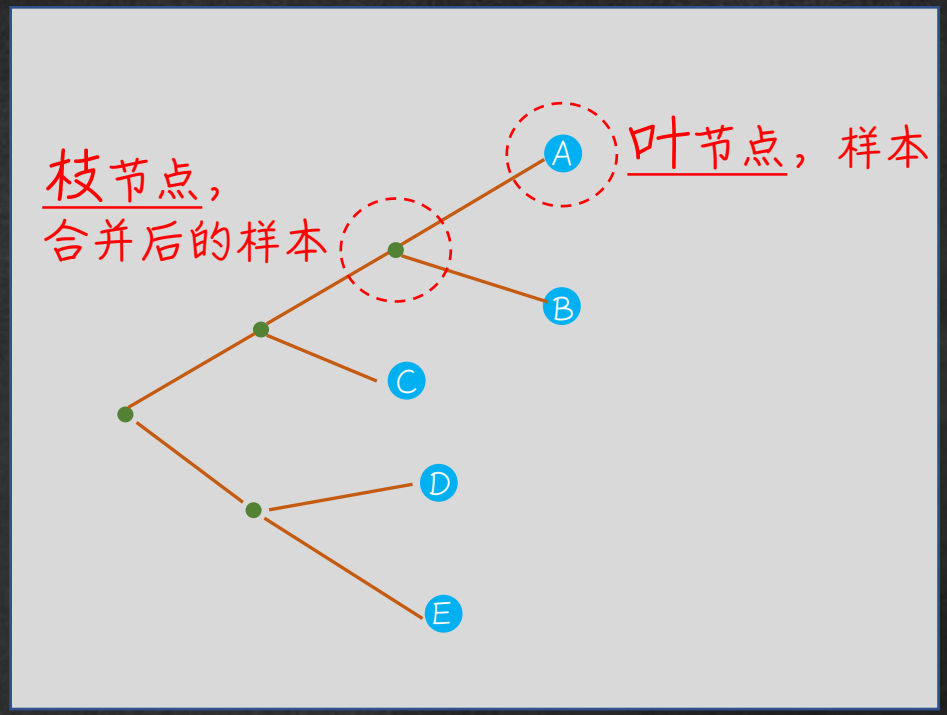
继续计算两两样本间距
(AB)与C之间距离最近
合并为(ABC)

继续计算两两样本间距
D与E之间距离最近
合并为(DE)

继续计算两两样本间距
(ABC)与(DE)之间距离最近
合并为(ABCDE)



- 最终层级聚类结果示例



定义距离

- 计算样本之间的紧密度（closeness, p35）
- 计算方法类同ch2中提到的样本相似度方法，如pearson系数法
- 特别适合比较文本类型的样本间相似度(p47)

以文本样本集为例
将原始文本转化为以下矩阵
每行为某个样本所包含的分词统计情况

样本	word 1	word 2	word n
Sample 1	6	3	1
Sample 2	5	1	8
Sample n	4	6	7

如此一来，每个样本可用一组向量进行表征

Sample 1（6, 3, ..., 1）

通过pearson系数法定义两组向量(样本)间的紧密度(相似度)

```
from math import sqrt
def pearson(v1,v2):
    sum1 = sum(v1)  #求和
    sum2 = sum(v2)

    sum1Sq = sum([pow(v,2) for v in v1]) #求平方的和
    sum2Sq = sum([pow(v,2) for v in v2])

    pSum = sum([v1[i] * v2[i] for i in range(len(v1))]) #求乘积的和

    num = pSum - (sum1 * sum2/len(v1))
    den = sqrt((sum1Sq - pow(sum1,2)/len(v1)) * (sum2Sq - pow(sum2,2)/len(v1)))

    if den ==0: return 0

    return 1.0 - num/den #相似度越高，距离越小，将p系数转换为距离
```

定义距离

- 除了person系数法外，教材p47还介绍了一种紧密度计算方式
- Tanimoto系数法，表示样本之间的交集与并集的比率

以p47为例——用户偏好集合
每行表示每个用户对物品的喜好
1表示想拥有、0表示不想拥有

样本	item 1	item 2	item n
User 1	1	0	1
User 2	0	0	8
User n	0	1	1

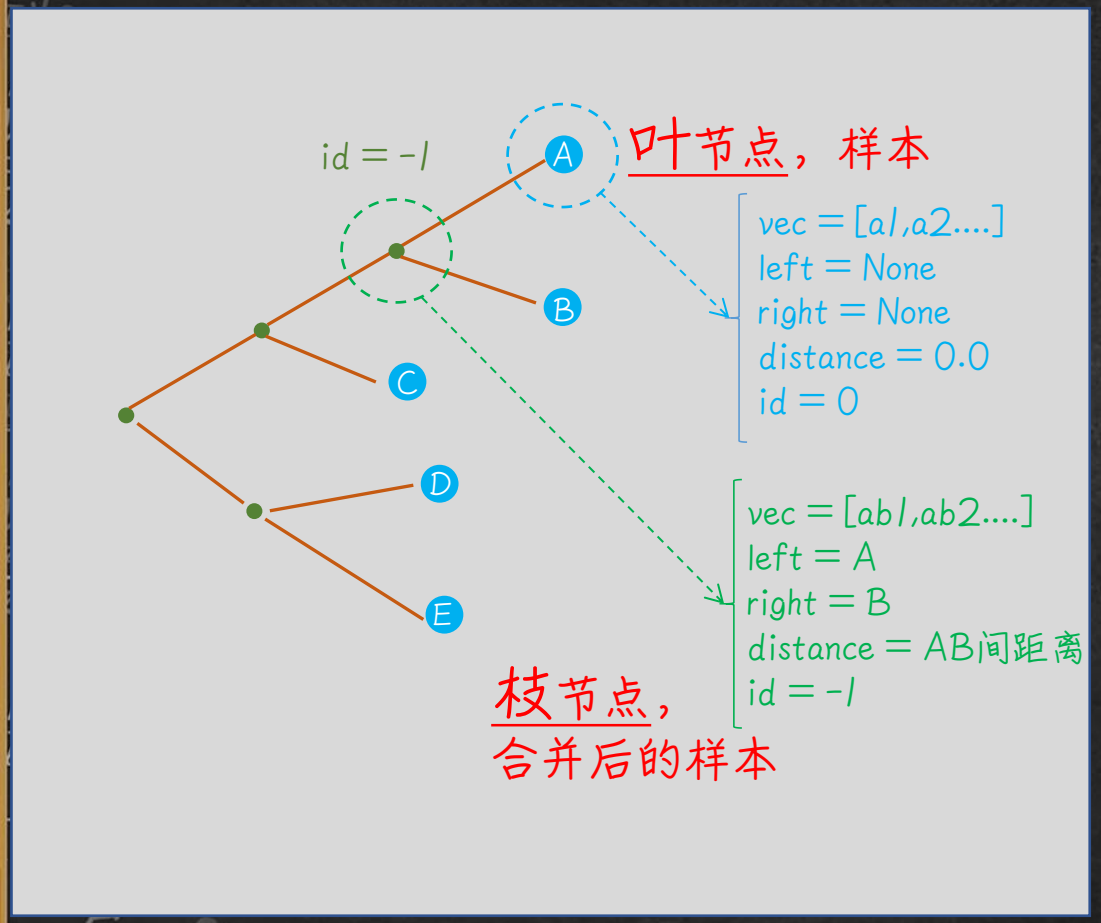
如此一来，每个用户的偏好可用一组向量表征
且向量元素只有0和1构成

User 1 (1, 0, ..., 1)

通过tanimoto系数法定义两组向量(样本)间的紧密度(相似度)

```
def my_tanimoto(v1,v2):

    c1,c2,shr = 0,0,0

    for i in range(len(v1)):
        if v1[i] != 0: c1+=1  #对v1序列中的1进行计数
        if v2[i] != 0: c2+=1  #对v2序列中的1进行计数
        if v1[i] != 0 and v2[i] != 0: shr+=1 #对v1、v2序列中同为1进行计数

    return 1.0 - (float(shr)/(c1+c2-shr))  #v1与v2同为1的个数 / v1或v2为1的数目
```

定义“树”



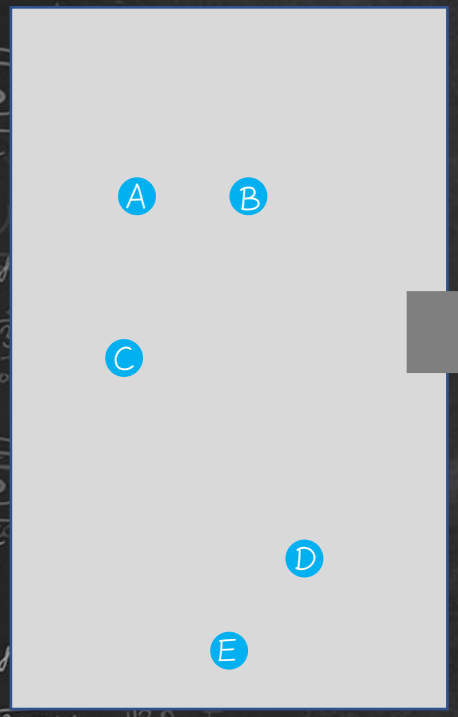

```
#无论是枝节点还是叶节点，都可以定义为一个bicluster的实例对象
class bicluster:

    def __init__(self,vec,left = None, right = None, distance = 0.0, id = None):
        #对叶节点，为None；对枝节点，为合并前的节点
        self.left = left

        #对叶节点，为None；对枝节点，为合并前的节点
        self.right = right

#对叶节点，为原始特征向量；对枝节点，为参与合并的两个节点平均向量值
        self.vec = vec

    #若为叶节点，则从0开始一次编号；若为枝节点，则从-1开始依次编号
        self.id = id

        #对叶节点，默认为0.0；对枝节点，为参与合并两个节点间的距离
        self.distance = distance
```

核心：更新一对、一组

clust，更新一组样本列表

两两距离 → lowestpair，更新一对最短距离样本 → 两两合并

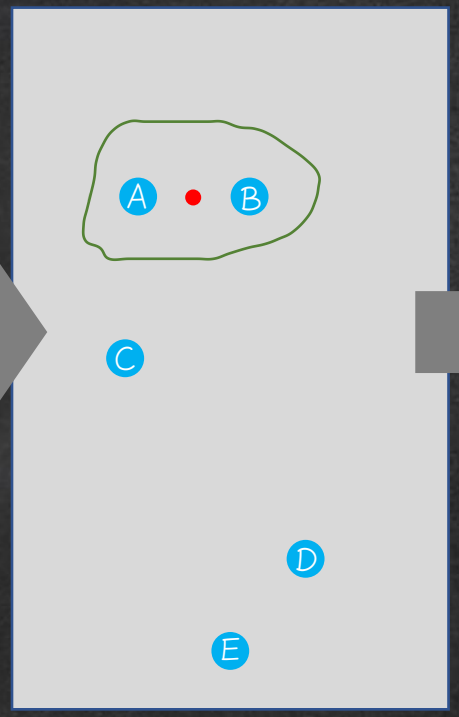
初始状态下
将每个样本视为
单独的一类



lowestpair = ()

clust = [A,B,C,D,E]

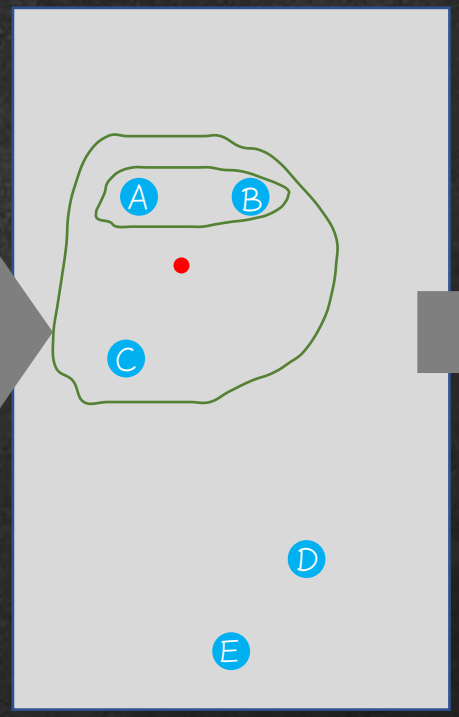
计算两两样本之间的距离
B与A间距离最近
合并为(AB)



lowestpair = (A,B)

clust = [(AB),C,D,E]

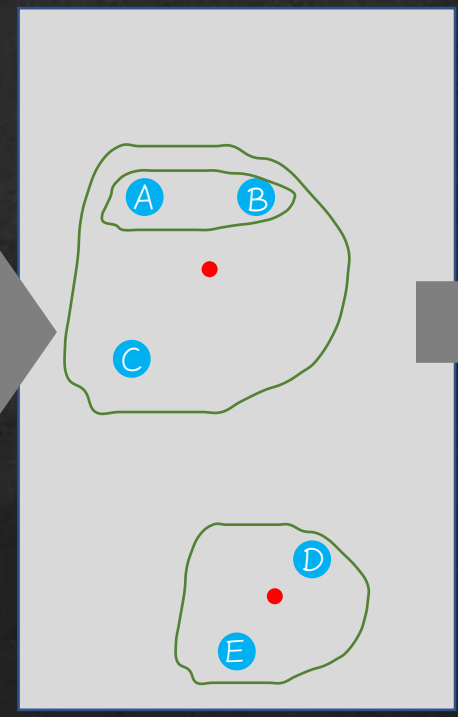
继续计算两两样本间距
(AB)与C之间距离最近
合并为(ABC)



lowestpair = ((AB),C)

clust = [(ABC),D,E]

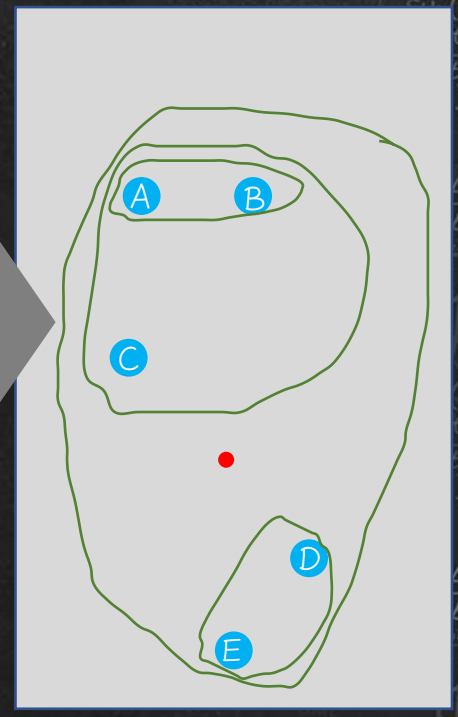
继续计算两两样本间距
D与E之间距离最近
合并为(DE)



lowestpair = (D,E)

clust = [(ABC),(D,E)]

继续计算两两样本间距
(ABC)与(DE)之间距离最近
合并为(ABCDE)



lowestpair = ((ABC),(DE))

clust = [(ABC),(D,E)]

核心：更新一对、一组

clust，更新一组样本列表

两两距离 → lowestpair，更新一对最短距离样本 → 两两合并

```
def hcluster(rows, distance = pearson):
    #参数1：特征向量列表
    #参数2：距离计算函数，默认为p系数计算
    distances = {}#字典，存放两两配对的p系数距离,其中key为(id1,id2)，value为两个ID所代表节点的距离
    currentclustid = -1#为合并后的枝节点统一分配编号-1,随着不断更新，依次减1
    #创建初始节点列表，初始情况下全部为叶节点
    clust = [ bicluster(rows[i], id = i) for i in range(len(rows))]
    while len(clust) > 1:#只要节点列表中的节点数>1，就继续执行聚类
        lowestpair = (0,1)#初始化一对最短距离样本
        closest = distance(clust[0].vec,clust[1].vec)
        #内外层循环配合，遍历(i,i+1)节点对，并计算p系数距离
        for i in range(len(clust)):
            for j in range(i+1,len(clust)):
                #如果距离字典中没有该对节点的距离
                if (clust[i].id,clust[j].id) not in distances:
                    distances[(clust[i].id,clust[j].id)] = distance( clust[i].vec, clust[j].vec )
                d = distances[(clust[i].id,clust[j].id)]
                if d < closest:
                    #更新距离最小配对
                    closest = d
                    lowestpair = (i,j)
        #计算距离最短配对的平均特征向量
        mergevec = [ (clust[lowestpair[0]].vec[i] + clust[lowestpair[1]].vec[i])/2.0 for i in range(len(clust[0].vec)) ]
        #创建合并后的枝节点
        newcluster = bicluster(mergevec, left = clust[lowestpair[0]], right = clust[lowestpair[1]], distance = closest, id = currentclustid)
        #更新一次，减1
        currentclustid -= 1
        #删除已合并的节点
        del clust[lowestpair[1]]
        del clust[lowestpair[0]]
        #添加合并后的枝节点
        clust.append(newcluster)

    #返回最终的枝节点
    return clust[0]
```

➢ 实践案例1：基于教材RSS源英文文本的聚类(p30-31)，已知条件

- ✓ 由于教材中的英文博客RSS源大部分已停止服务，因此根据教材提示，我们直接从网上下载已整理好的数据集（可直接通过本人github下载数据集）
- ✓ 数据集文件名为blogdata.txt
- ✓ 数据集由39 x 396的矩阵组成
- ✓ 行表示博客样本，每一条样本都是一段英文文本，累计有39个博客
- ✓ 列表示各博客中出现的分词，用于记录每个分词在每一段博客中出现的频数，累计有396个分词

分词 ←→

博客 ↑↓

	“china”	“kids”	“music”	“yahoo”	……
Gothamist	6	3	1	词频	
GigaOM	5	1	8		
Quick Online Tips	4	6	7		
……					

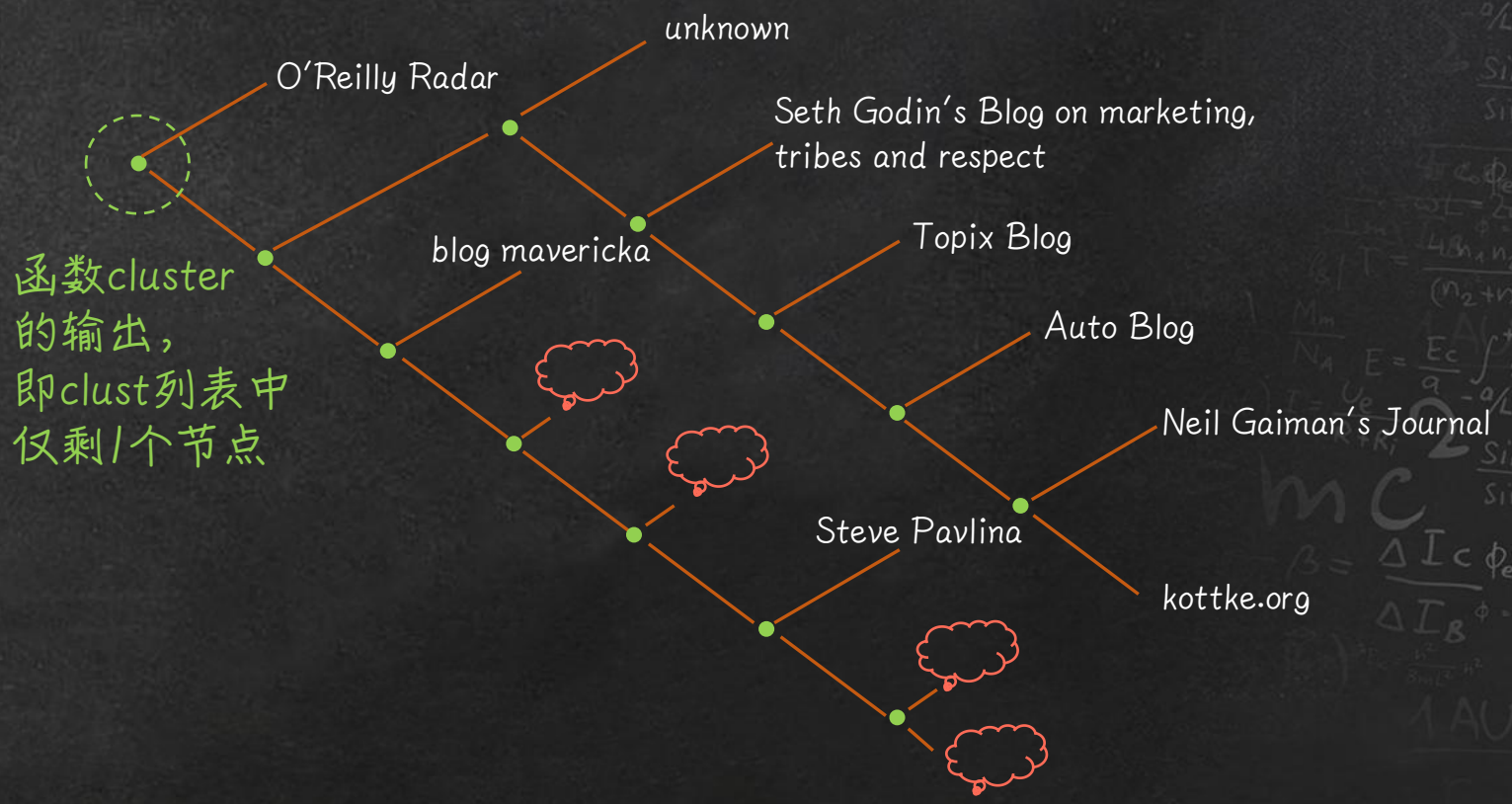
➢ 实践案例1：基于教材RSS源英文文本的聚类，可视化层级聚类结果（p37）

函数printclust()：以树形结构打印整个聚类结果(局部截图），但并不易于理解，可以转换为右侧的鱼骨图形式

```
-
 O'Reilly Radar
 -
  -
   unknow
   -
    Seth Godin's Blog on marketing, tribes and respect
    -
     Topix Blog
     -
      Autoblog
      -
       Neil Gaiman's Journal
       kottke.org
 -
  blog maverick
  -
   -
    -
     Steve Pavlina
     -
      Joho the Blog
      -
       Guy Kawasaki
       -
        Lifehack
        -
         ThinkProgressPage Array - ThinkProgress
         WIRED
    -
     Copyblogger
     -
      Search Engine Watch
      -
       Valleywag
       -
        Lifehacker
        -
         Kotaku
         -
          Gawker
          Deadspin
```

枝节点

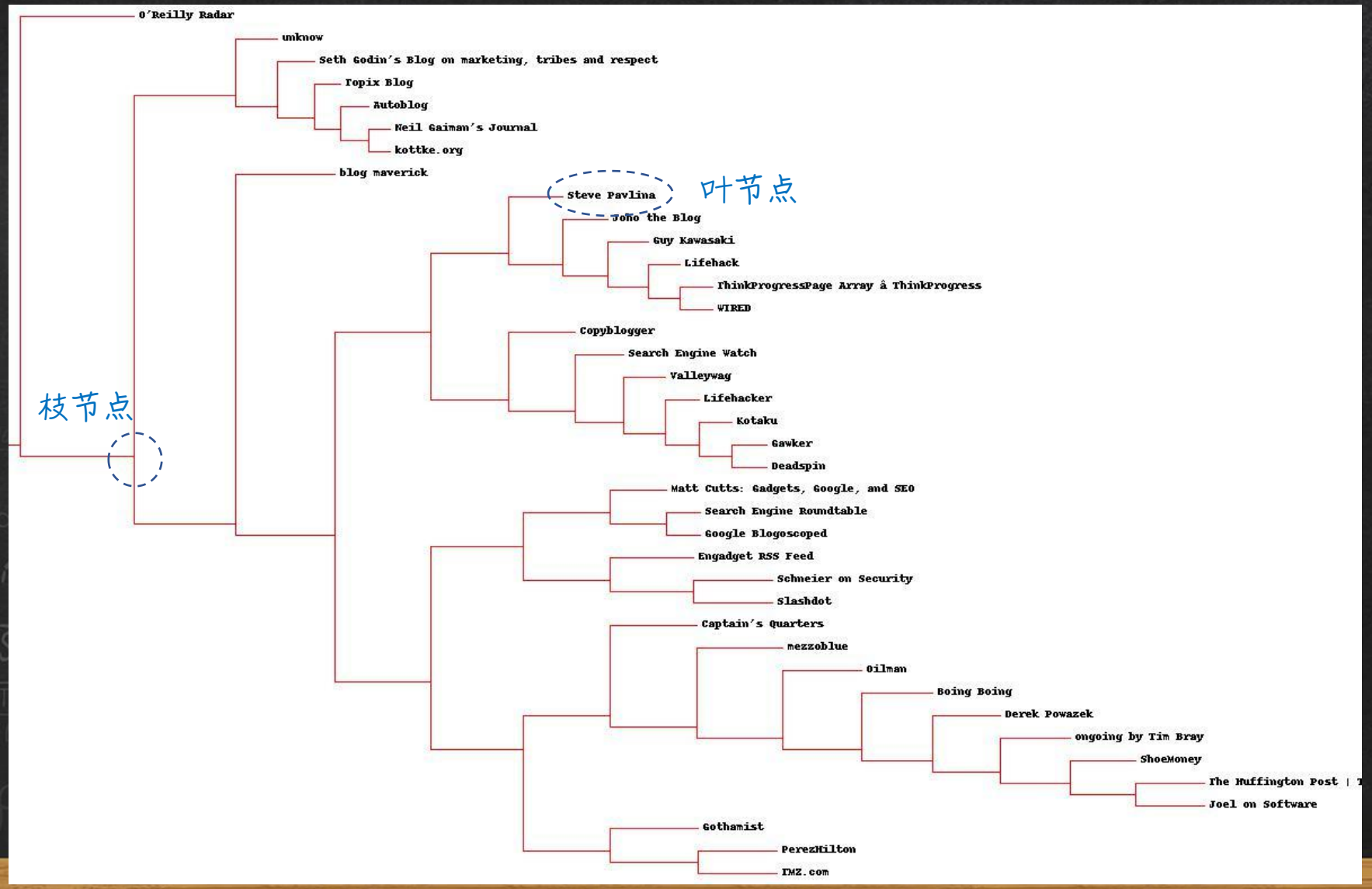
叶节点



函数cluster的输出，即clust列表中仅剩1个节点

➢ 实践案例1：基于教材RSS源英文文本的聚类，可视化层级聚类结果（p37）

函数drawdendrogram()：以更形象的绘图方式打印树形聚类结构



➢ 建立层级聚类及可视化聚类结果的函数清单

代码文件	函数	参数	输出
clusters.py	hcluster(rows, distance = pearson)	-rows, m(样本数) x n(分词数) -distance, 样本间距离算法，默认使用person	最终合并后的1个节点，bicluster实例对象
	printclust(clust,labels = blognames)	-clust，函数hcluster返回的最终节点 -labels，节点名称列表	逐行打印聚类结果
	drawdendrogram(clust, labels, jpeg = 'blog_hicluster_treegram.jpg')	-clust，函数hcluster返回的最终节点 -labels，节点名称列表 -jpeg, 树状图保存路径	树状图将按要求生成并保存至指定路径

实践案例2：基于知乎RSS源中文文本的聚类

- ✓ 为了能完整体现教材中RSS源示例，包括从RSS源获取、预处理等过程，我们以中文知乎RSS为数据源，参考书中处理过程
- ✓ 何为RSS？
- ✓ 以下图为例，我们通过google RSS插件获取国家统计局RSS源

➢ 实践案例2：基于知乎RSS源中文文本的聚类

✓ 何为RSS？

- RSS是一种共享消息的格式规范，以XML形式存在。你可以通过这种格式传递信息给其他人，可以通过这种格式从各种来源接受消息。

✓ 何为RSS feed（RSS源）？

- 一个网站所提供的信息记录在特定XML文件中称为RSS feed。

✓ 为什么使用RSS？

- 以XML格式来获得特定网站提供的信息，并将获得信息展示在网页上。
- 对于内容的提供者，RSS允许他传播自己网站的信息和新闻。

✓ RSS工作过程

- RSS feed由一个XML文件定义，该XML文件包括每个想要展示页面的URL，标题和摘要。
- 想要在自己电脑上阅读RSS feed，只需使用RSS阅读器或者浏览器，将这些feed添加到你的阅读器中。
- 其他接收者网站想要显示 feed 须要从提供者处获得RSS文件，提取URL页面，并显示标题和摘要。这些工作可以通过PHP脚本完成。
- 通过点击新闻列表中的某一项，可以访问该新闻提供者的原网页。

➢ 实践案例2：基于知乎RSS源中文文本的聚类

✓ 知乎RSS源的XML文件，包含n个知乎话题和摘要，这是我们要提取出来的2个重要字段值

{'bozo': False,
 'entries': [{'title': 'PS5 国行版已经发售，拿到手后体验如何？',
   'title_detail': {'type': 'text/plain',
    'language': None,
    'base': 'https://www.zhihu.com/rss',
    'value': 'PS5 国行版已经发售，拿到手后体验如何？'},
   'links': [{'rel': 'alternate',
     'type': 'text/html',
     'href':
'http://www.zhihu.com/question/459622050/answer/1888024843?utm_campaign=rss&utm_medium
=rss&utm_source=rss&utm_content=title'}],
   'link':
'http://www.zhihu.com/question/459622050/answer/1888024843?utm_campaign=rss&utm_medium
=rss&utm_source=rss&utm_content=title',
   'summary': ' 最重要的信息放在文章开头：「系统提供的备份还原功能与 PS4 国行相比没有变
化」。 对于一台发售约半年之后仍然深陷缺货和加价泥潭的主机来说，.....

<- - - - - - 一个知乎话题

该话题摘要

➢ 处理RSS源的函数清单

代码文件	函数	参数	输出
import feedparser	url = 'https://www.zhihu.com/rss' d = feedparser.parse(url)	-url，特定网站发布的RSS源地址	将XML文件转换为键值对形式的字典
import BeautifulSoup	for e in d.entries: soup = BeautifulSoup(e.summary,'html.parser')	-带有HTML标记的文本 -指定解析器	去除HTML标记的文本
import jieba	for i in jieba.cut(str):	-指定一段文本	文本的分词结果列表
import re	for s in re.findall('[\u4e00-\u9fa5a-zA-Z]+',text):	-匹配中文字符及英文字符	-
	stopwords = [line.strip() for line in open('.....\stopwords.txt',encoding='UTF-8').readlines()]	-	停用词列表

实践案例2：基于知乎RSS源中文文本的聚类

经过一系列文本处理后，可以得到 m（知乎话题数） x n(分词数)的矩阵

Question	PS	国行	主机	游戏	功能	玩家	商店	系统	版本	发售	海外	体验	手柄	PSN	认证	支持	相关	SSD	提供	相比	变化	价格	内容
PS5 国行版已经发售，拿	49	21	18	16	11	11	10	9	9	8	7	7	7	7	7	7	6	6	5	5	5	5	
法国新浪潮电影对于电影	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	7	0	
如何看待 Playstation 中	11	0	0	10	0	9	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	
如何评价《画江湖之不良人	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何评价国产漫画《BLISS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
顶刊综述吵起来了，我们这	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
可以有已经通关了《生化危	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	
中国漫画的寒冬到底是因为	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
新生儿脐带该怎么护理？	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
98版仙剑 ，真的有办法可	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
怎么在 Limited Run Gam	0	0	0	60	0	6	0	0	6	6	0	0	0	0	0	0	8	0	0	0	0	0	
《龙珠》和《达伊的大冒险	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何评价张颂文和姜武主演	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
什么性格的人容易得抑郁症	0	0	0	0	0	0	0	8	0	0	0	5	0	0	0	0	5	0	0	0	0	0	
如何评价中国漫画家描述中	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
《秘密访客》中的餐桌戏和	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何评价游戏《天地劫：幕	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何评价 24 Entertainm	0	0	0	12	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
在你的心目中，奥特曼是什	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何评价《画江湖之不良人	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何通过玩游戏拯救地球？	0	0	0	97	16	9	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	
里基·戴维斯遇到刚出道的	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何评价2021年4月国产动	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何评价苹果搭载 M1 芯片	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何评价国产动画《盗墓笔	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
父母都还在受苦，我怎么敢	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
眶隔脂肪释放丨眼部整形术	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
有哪些关于护肤品的「伪概	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何评价《一人之下》第5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
There and back again一	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
中国第六代导演有哪些共同	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
日本动漫中有哪些著名的猫	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
诈骗为什么那么难抓，现在	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何评价东契奇和约基奇这	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何评价《特利迦奥特曼》	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
家长该如何去激发孩子学习	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
北京一小学到校先上体育课	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
新加坡major全记录	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
什么是谷圈？新人有哪些要	0	0	0	6	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何评价黄蜂后卫特里·墨	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	
如何评价《奈克瑟斯奥特曼	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何评价2021年四月番《S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
如何评价游戏《炎龙骑士团	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

my_first_sheet Sheet

➢ 实践案例2：基于知乎RSS源中文文本的聚类

以树形结构打印整个聚类结果(局部截图)

-　　——————————→ 枝节点
*-
**2021 年 5 月 18 日是国际博物馆日，你认为「一生一定要去」的博物馆是哪座？
**古徽州旅游资源有哪些？ ——————————→ 叶节点
*-
**如何评价苹果搭载 M1 芯片的 iPad Pro，有哪些亮点和槽点，值得购买吗？
**-
***-
****奥运模拟赛男单决赛黑马周启豪一黑到底 4:2 战胜樊振东，如何评价本场比赛？
****里基·戴维斯遇到刚出道的詹皇时是一个什么样的人?他的实力究竟配不配得上骑士前老大的身份?
***-
****98版仙剑 ，真的有办法可以复活林月如吗？
****-
*****可以有已经通关了《生化危机 8》的大神来讲解一下剧情吗？
*****-
******如何评价游戏《天地劫：寰神结》？
******-
*******-
********《圣斗士星矢》中的众神体系是怎么样的？

聚类结果树形图（全部）

聚类结果树形图（局部）

以更形象的绘图方式打印树形聚类结构

02

K-均值聚类

- 用“扁平化”替代“层级”
- 基于文本进行聚类分析

-案例1：基于教材RSS源英文文本的聚类

-案例2：基于知乎RSS源中文文本的聚类

➢ 图解k均值，假设将5个样本分为2类

随机点 → 近距离 → 转移点

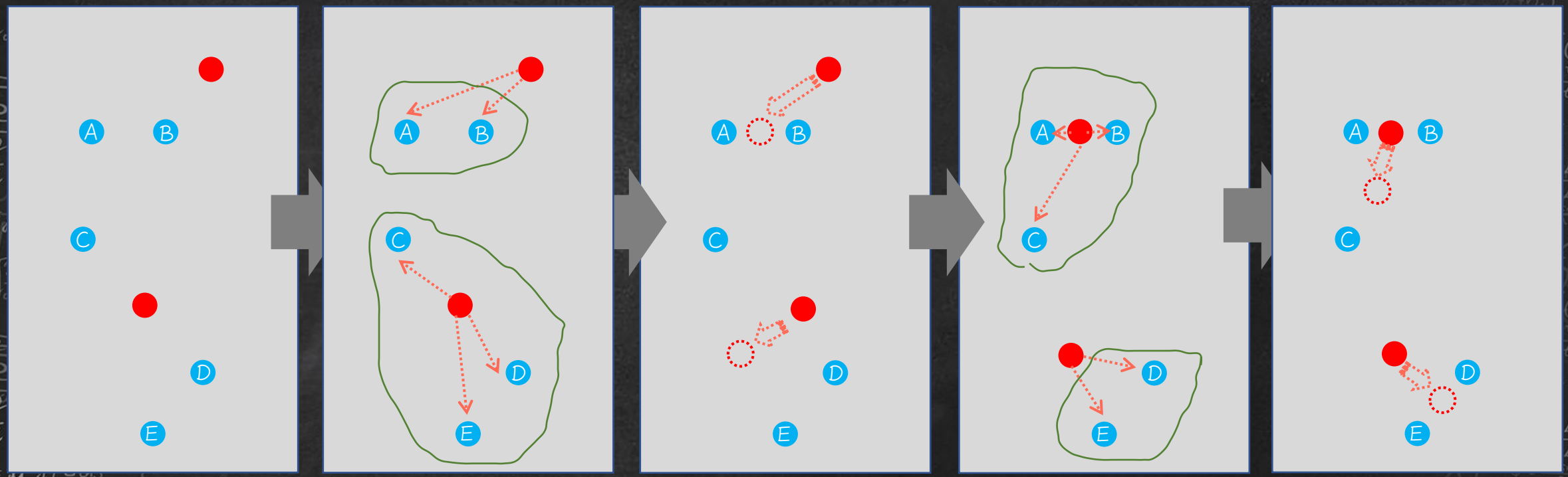
随机2个中心点

判断离中心点最近的样本

求最近样本集的平均位置转移中心点

再次判断离中心点最近的样本

再次转移中心点



当分类结果不再变化时，可停止

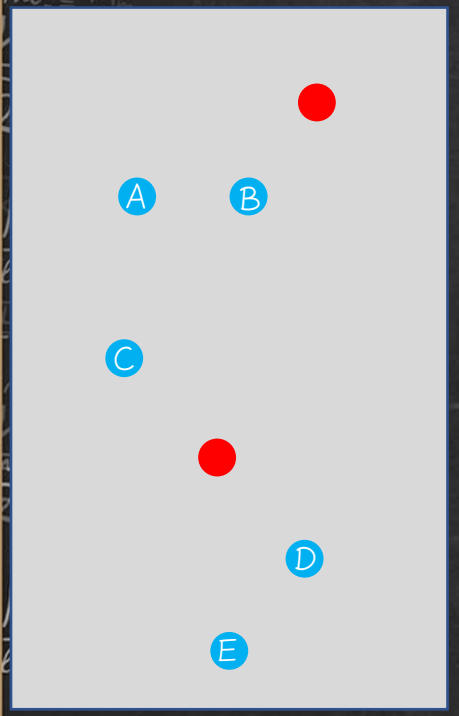
最终分类结果

-类别1：ABC

-类别2：DE

随机点

随机n个中心点(类)



样本	word 1	word 2	word n
Sample 1	6 max	3	1
Sample 2	5	1	8
Sample n	4 min	6	7
中心点 1（初始）	?	?	?
中心点 k（初始）	?	?	?

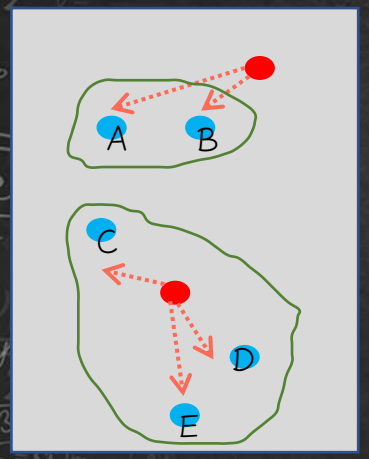
(min, max) = min + (0,1) x (max - min)

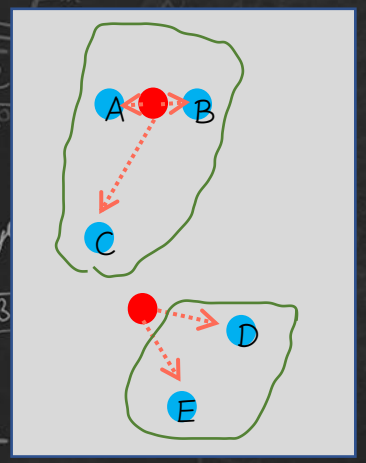
```
#找到每一个特征（分词）的最大词频数和最小词频数
ranges = [(min([row[i] for row in rows]), max([row[i] for row in rows]))
for i in range(len(rows[0]))]

#随机创建k个质心，即根据每一个分词的词频数范围，每一个分词
随机频数 = 0-1的随机数 * （最大频数 - 最小频数） + 最小频数，
再随机出k条记录
 clusters = [ [random.random() * (ranges[i][1] - ranges[i][0]) +
ranges[i][0] for i in range(len(rows[0]))] for j in range(k) ]
```

分词	样本中最大频数	样本中最小频数
word 1	6	4
word 2	6	1
word n	8	1

近距离

判断离中心点<u>最近</u>的样本





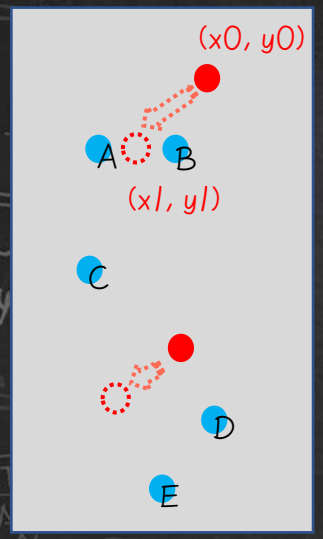

```
for t in range(100):
    print('当前迭代次数 %d' % t)

    #初始化最佳质心，共k个
    bestmatches = [[] for i in range(k)]

    #遍历每一个样本
    for j in range(len(rows)):
        bestmatch = 0#假定第0号质心与当前点距离最近

        #遍历每一个中心
        for i in range(k):
            #计算每一个中心与当前点的距离，为每一个点找到离其最近的质心，并加入到bestmatches列表中
            d = distance( rows[j], clusters[i] )
            #bestmach会不断更新
            if d < distance( rows[j], clusters[bestmatch] ) : bestmatch = i

        #找到与当前点最近的质心后，将该点添加到bestmatches列表中
        bestmatches[bestmatch].append(j)

    #如果在100次以内，bestmatches列表不再更新，则自动推出迭代循环
    if bestmatches == lastmatches: break

    #t每迭代一次后，需要记录上一次点分类结果
    lastmatches = bestmatches
```

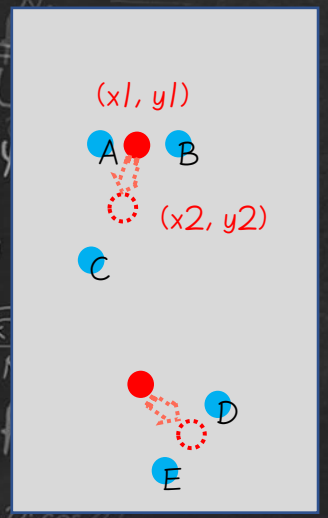
	第1轮最近样本	第2轮最近样本
中心点1	A/B	A/B/C
中心点2	C/D/E	D/E

转移点

求最近样本集的平均位置

转移中心点





	中心点 初始位置	第1轮最近样本	第1次转移后 中心位置	第2轮最近样本	第2次转移后 中心位置
中心点1	(x0, y0)	A/B	(x1, y1)	A/B/C	(x2, y2)
中心点2	(x'0, y'0)	C/D/E	(x'1, y'1)	D/E	(x'2, y'2)

条件

```
for i in range(k):
        #初始化平均点位置值
        avgs = [0.0] * len(rows[0])

        #注意可能存在该质心下不存在离其最近的点，因此要加判断
        if len(bestmatches[i]) > 0:

            #遍历当前质心下，与其最近的所有点
            for rowid in bestmatches[i]:
                for colid in range(len(rows[rowid])):#可以为0
                    avgs[colid] += rows[rowid][colid]
            #求平均值
            for colid in range(len(rows[0])):
                avgs[colid] = avgs[colid] / len(bestmatches[i])
            #得到最终的质心
            clusters[i] = avgs
```

➢ 实践案例1：英文博客语料库k-均值分类结果,假设将39条英文博客分为5类

第1次初始化分类结果 迭代次数: 1	第1类集合 12个	'Schneier on Security', 'PerezHilton', 'mezzoblue','The Huffington Post | The Full Feed', 'Joho the Blog', 'ongoing by Tim Bray','Kotaku','Derek Powazek', "Captain's Quarters",'Boing Boing','WIRED','Guy Kawasaki'
	第2类集合 9个	'Search Engine Roundtable','Copyblogger', 'ShoeMoney','Slashdot','Matt Cutts: Gadgets, Google, and SEO','Google Blogoscoped', "O'Reilly Radar",'Lifehacker', 'Oilman'
	第3类集合 6个	'unknow',"Neil Gaiman's Journal",'Topix Blog','Autoblog','kottke.org',"Seth Godin's Blog on marketing, tribes and respect"
	第4类集合 1个	'Steve Pavlina'
	第5类集合 11个	'Engadget RSS Feed','Gothamist','Joel on Software','Search Engine Watch', 'ThinkProgressPage Array - ThinkProgress','blog maverick', 'Gawker','Deadspin','TMZ.com','Lifehack','Valleywag'

第2次初始化分类结果 迭代次数: 1	第1类集合 21个	'Schneier on Security','Search Engine Roundtable', 'Copyblogger','Engadget RSS Feed', 'mezzoblue','ShoeMoney', 'Slashdot','Matt Cutts: Gadgets, Google, and SEO', 'Gothamist','The Huffington Post | The Full Feed','Joho the Blog','Search Engine Watch', 'Kotaku','Derek Powazek',"Captain's Quarters",'Gawker','Google Blogoscoped', 'TMZ.com',"O'Reilly Radar",'Lifehacker','Valleywag'
	第2类集合 6个	'Joel on Software', 'Steve Pavlina','ThinkProgressPage Array - ThinkProgress', 'Lifehack','Boing Boing','Oilman'
	第3类集合 2个	'ongoing by Tim Bray', 'blog maverick'
	第4类集合 4个	'unknow','PerezHilton','Autoblog', "Seth Godin's Blog on marketing, tribes and respect"
	第5类集合 6个	"Neil Gaiman's Journal",'Topix Blog','Deadspin', 'kottke.org','WIRED', 'Guy Kawasaki'

实践案例2：中文RSS语料库k-均值第1次分类结果,假设将56条中文话题分为5类(1/2)

第1次初始化分类结果 迭代次数: 3	第1类集合 9个	'如何评价中国漫画家描述中国支援印度抗疫的作品《同桌的你》？','如何评价《画江湖之不良人》第四季第 1 - 2集？','父母都还在受苦，我怎么敢过得好？追踪调查100人，揭穿中国式家庭关系拧巴的根源','诈骗为什么那么难抓，现在什么都实名制了追踪钱的去向那么难吗？','如何评价东契奇和约基奇这两个核心超巨，他们带队的方式有什么异同？','家长该如何去激发孩子学习的斗志？','北京一小学到校先上体育课再上文化课，你赞成这样的上课顺序吗？有哪些注意事项？','如何评价2021年四月番《SSSS.电光机王》（SSSS.DYNAZENON）？','如何评价阿尔德里奇在马刺的职业生涯？'
	第2类集合 16个	'法国新浪潮电影对于电影发展影响有多大？','顶刊综述吵起来了，我们还应该推荐大众少吃盐吗？','新生儿脐带该怎么护理？','《龙珠》和《达伊的大冒险》中这两个相似的动作场景，鸟山明的处理高明在哪里？', '如何评价张颂文和姜武主演的电影《扫黑决战》？','在你的心目中，奥特曼是什么？','里基·戴维斯遇到刚出道的詹皇时是一个什么样的人?他的实力究竟配不配得上骑士前老大的身份?','有哪些关于护肤品的「伪概念」？','There and back again——跨越半个世纪的《魔戒》改编路','中国第六代导演有哪些共同的特征？如何评价他们的作品和成就？','如何评价《特利迦奥特曼》？','什么是谷圈？新人有哪些要注意的地方？','如何评价《奈克瑟斯奥特曼》？', '湖人名宿埃尔金·贝勒去世，享年 86 岁，你对他有什么印象？','为什么武汉被称为东方芝加哥？','如何评价独行侠球员杰伦·布伦森今年的表现？这三年来他的进步有多大？'
	第3类集合 8个	'如何评价国产漫画《BLISS～极乐幻奇谭》？','中国漫画的寒冬到底是因为网文，还是说原创漫画家挣不到钱？','《秘密访客》中的餐桌戏和餐的设计有什么寓意？','眶隔脂肪释放丨眼部整形系列科普第十七篇','日本动漫中有哪些著名的猫？','新加坡major全记录','如何评价游戏《炎龙骑士团II：黄金城之谜》？','《DOTA2》新加坡 Major 决赛 iG 让二追三击败 EG 夺冠，你有什么想说的？'
	第4类集合 13个	'如何评价《画江湖之不良人》第四季 第 4 集？','可以有已经通关了《生化危机 8》的大神来讲解一下剧情吗？','如何评价游戏《天地劫：寰神结》？', '如何评价2021年4月国产动画《秘宝之国》？','如何评价苹果搭载 M1 芯片的 iPad Pro，有哪些亮点和槽点，值得购买吗？','如何评价国产动画《盗墓笔记：秦岭神树》？', '如何评价《一人之下》第522（554）话？','如何评价黄蜂后卫特里·罗齐尔？','如何评价漫画《摩登神仙》？', '快船有更好的进攻，但湖人却更能打到篮网的痛点','如何评价游戏《流言侦探》最新 DLC《好久不见》？',' “我最大的恐惧是食物和水”丨8.2分纪录片《强迫症·心魔》'
	第5类集合 10个	'PS5 国行版已经发售，拿到手后体验如何？','如何看待 Playstation 中国的 PS5 国行庆典活动？', '98版仙剑，真的有办法可以复活林月如吗？','怎么在 Limited Run Games 上购买游戏卡带？','什么性格的人容易得抑郁症？','如何评价 24 Entertainment 研发的游戏《永劫无间》?','如何通过玩游戏拯救地球？应该怎么做？','如何评价游戏《天地劫：幽城幻剑录》？','为什么有32个关卡的游戏《超级马里奥兄弟》只要40KB?','如何评价电影《西游记之再世妖王》?'

➢ 以第1次聚类结果为例，对第1类样本集进行进一步分析（1/2）

关键词：漫画评价

关键词：篮球运动员评价

1类

‘如何评价中国漫画家描述中国支援印度抗疫的作品《同桌的你》？’,
‘如何评价《画江湖之不良人》第四季第1-2集？’,
‘父母都还在受苦，我怎么敢过得好？追踪调查100人，揭穿中国式家庭关系拧巴的根源’,
‘诈骗为什么那么难抓，现在什么都实名制了追踪钱的去向那么难吗？’,
‘如何评价东契奇和约基奇这两个核心超巨，他们带队的方式有什么异同？’,
‘家长该如何去激发孩子学习的斗志？’,
‘北京一小学到校先上体育课再上文化课，你赞成这样的上课顺序吗？有哪些注意事项？’,
‘如何评价2021年四月番《SSSS.电光机王》（SSSS.DYNAZENON）？’,
‘如何评价阿尔德里奇在马刺的职业生涯？’

关键词：家庭教育

✓ 问题1：同一个分类样本集的分类结果并不完全准确，存在一定误差

- 比如，针对第1类样本集中的9个样本，可进一步分为：家庭教育（3个）、漫画评价（3个）、篮球运动员评价（2个）、诈骗（1个）
- 其中，漫画评价、篮球运动员评价等语料可进一步抽象为评价类语料
- 但是，家庭教育、诈骗等语料很难再进一步进行合并，本次聚类结果内容明显存在较大差异

➢ 以第1次聚类结果为例，对第1类样本集进行进一步分析（2/2）

1类

'如何评价中国漫画家描述中国支援印度抗疫的作品《同桌的你》？',
'如何评价《画江湖之不良人》第四季第1-2集？',
'父母都还在受苦，我怎么敢过得好？追踪调查100人，揭穿中国式家庭关系拧巴的根源',
'诈骗为什么那么难抓，现在什么都实名制了追踪钱的去向那么难吗？',
'如何评价东契奇和约基奇这两个核心超巨，他们带队的方式有什么异同？',
'家长该如何去激发孩子学习的斗志？',
'北京一小学到校先上体育课再上文化课，你赞成这样的上课顺序吗？有哪些注意事项？',
'如何评价2021年四月番《SSSS.电光机王》（SSSS.DYNAZENON）？',
'如何评价阿尔德里奇在马刺的职业生涯？'

关键词：漫画评价

3类

'如何评价国产漫画《BLISS～极乐幻奇谭》？',
'中国漫画的寒冬到底是因为网文，还是说原创漫画家挣不到钱？',
'《秘密访客》中的餐桌戏和餐的设计有什么寓意？',
'眶隔脂肪释放丨眼部整形系列科普第十七篇','日本动漫中有哪些著名的猫？',
'新加坡major全记录',
'如何评价游戏《炎龙骑士团II：黄金城之谜》？',
'《DOTA2》新加坡 Major 决赛 iG 让二追三击败 EG 夺冠，你有什么想说的？'

✓ 问题2：不同分类样本集中存在同类样本，即样本集和样本集的分类界限存在一定误差

- 比如，针对第1类和第3类两个不同的样本集
- 其中，第1类中有3个样本与漫画评价有关，第3类中也有3个与漫画评价有关
- 但是本次聚类结果没有将上述样本有效合并归类

➢ 实践案例2：中文RSS语料库k-均值第2次分类结果

第2次初始化分类结果 迭代次数: 4	第1类集合 12个	'法国新浪潮电影对于电影发展影响有多大？', '如何评价张颂文和姜武主演的电影《扫黑决战》？','如何评价中国漫画家描述中国支援印度抗疫的作品《同桌的你》？','日本动漫中有哪些著名的猫？','家长该如何去激发孩子学习的斗志？', '北京一小学到校先上体育课再上文化课，你赞成这样的上课顺序吗？有哪些注意事项？','什么是谷圈？新人有哪些要注意的地方？','如何评价黄蜂后卫特里·罗齐尔？', '快船有更好的进攻，但湖人却更能打到篮网的痛点', '为什么武汉被称为东方芝加哥？','如何评价阿尔德里奇在马刺的职业生涯？', '“我最大的恐惧是食物和水”丨8.2分纪录片《强迫症·心魔》'
	第2类集合 6个	'如何评价《画江湖之不良人》第四季 第 4 集？', 'There and back again——跨越半个世纪的《魔戒》改编路', '如何评价东契奇和约基奇这两个核心超巨，他们带队的方式有什么异同？', '新加坡major全记录', '如何评价独行侠球员杰伦·布伦森今年的表现？这三年来他的进步有多大？','《DOTA2》新加坡 Major 决赛 iG 让二追三击败 EG 夺冠，你有什么想说的？'
	第3类集合 8个	'PS5 国行版已经发售，拿到手后体验如何？', '如何看待 Playstation 中国的 PS5 国行庆典活动？', '98版仙剑，真的有办法可以复活林月如吗？','怎么在 Limited Run Games 上购买游戏卡带？', '《秘密访客》中的餐桌戏和餐的设计有什么寓意', '如何评价 24 Entertainment 研发的游戏《永劫无间》?', '如何通过玩游戏拯救地球？应该怎么做？', '为什么有32个关卡的游戏《超级马里奥兄弟》只要40KB?'
	第4类集合 20个	'如何评价国产漫画《BLISS～极乐幻奇谭》？', '顶刊综述吵起来了，我们还应该推荐大众少吃盐吗？', '可以有已经通关了《生化危机 8》的大神来讲解一下剧情吗？','中国漫画的寒冬到底是因为网文，还是说原创漫画家挣不到钱？', '《龙珠》和《达伊的大冒险》中这两个相似的动作场景，鸟山明的处理高明在哪里？', '在你的心目中，奥特曼是什么？', '如何评价《画江湖之不良人》第四季第 1 - 2集？', '如何评价2021年4月国产动画《秘宝之国》？','如何评价苹果搭载 M1 芯片的 iPad Pro，有哪些亮点和槽点，值得购买吗？', '如何评价国产动画《盗墓笔记：秦岭神树》？', '眶隔脂肪释放丨眼部整形系列科普第十七篇','有哪些关于护肤品的「伪概念」？','如何评价《一人之下》第522（554）话？','中国第六代导演有哪些共同的特征？如何评价他们的作品和成就？' '如何评价《特利迦奥特曼》？','如何评价《奈克瑟斯奥特曼》？', '如何评价2021年四月番《SSSS.电光机王》（SSSS.DYNAZENON）？','如何评价游戏《炎龙骑士团II：黄金城之谜》？', '如何评价漫画《摩登神仙》？', '如何评价游戏《天地劫：幽城幻剑录》？'
	第5类集合 9个	'如何评价《画江湖之不良人》第四季 第 4 集？','眶隔脂肪释放丨眼部整形系列科普第十七篇','新加坡major全记录', '什么是谷圈？新人有哪些要注意的地方？', '如何评价2021年四月番《SSSS.电光机王》（SSSS.DYNAZENON）？','如何评价游戏《炎龙骑士团II：黄金城之谜》？','为什么武汉被称为东方芝加哥？', '《DOTA2》新加坡 Major 决赛 iG 让二追三击败 EG 夺冠，你有什么想说的？'

➢ 以2次聚类结果进行比较分析

第1次

'法国新浪潮电影对于电影发展影响有多大？',
'如何评价张颂文和姜武主演的电影《扫黑决战》？',
'如何评价中国漫画家描述中国支援印度抗疫的作品《同桌的你》？',
'日本动漫中有哪些著名的猫？','家长该如何去激发孩子学习的斗志？',
'北京一小学到校先上体育课再上文化课，你赞成这样的上课顺序吗？有哪些注意事项',
'什么是谷圈？新人有哪些要注意的地方？',
'如何评价黄蜂后卫特里·罗齐尔？',
'快船有更好的进攻，但湖人却更能打到篮网的痛点',
'为什么武汉被称为东方芝加哥？',
'如何评价阿尔德里奇在马刺的职业生涯？',
'“我最大的恐惧是食物和水”丨8.2分纪录片《强迫症·心魔》'

第2次

'法国新浪潮电影对于电影发展影响有多大？',
'顶刊综述吵起来了，我们还应该推荐大众少吃盐吗？',
'新生儿脐带该怎么护理？',
'《龙珠》和《达伊的大冒险》中这两个相似的动作场景，鸟山明的处理高明在哪里？', '
如何评价张颂文和姜武主演的电影《扫黑决战》？',
'在你的心目中，奥特曼是什么？',
'里基·戴维斯遇到刚出道的詹皇时是一个什么样的人?他的实力究竟配不配得上骑士前老大的身份?',
'有哪些关于护肤品的「伪概念」？',
'There and back again——跨越半个世纪的《魔戒》改编路',
'中国第六代导演有哪些共同的特征？如何评价他们的作品和成就？',
'如何评价《特利迦奥特曼》？',
'什么是谷圈？新人有哪些要注意的地方？',
'如何评价《奈克瑟斯奥特曼》？',
'湖人名宿埃尔金·贝勒去世，享年 86 岁，你对他有什么印象？',
'为什么武汉被称为东方芝加哥？',
'如何评价独行侠球员杰伦·布伦森今年的表现？这三年来他的进步有多大？'

- ✓ 问题3：2次聚类结果迭代次数存在随机性
  - 第1次聚类需要迭代3次；
  - 而第2次聚类需要迭代4次。
- ✓ 问题4：2次聚类结果不完全相同，且差异较大
  - 将第1次聚类结果中的第2类样本集与第2次聚类结果中的第1类样本集进行比较
  - 2类样本集仅有4个样本一致（左侧黄字）
  - 其余大部分样本均不一致

小结

- ✓ 上述提到的4个问题，本质体现出k-均值聚类结果的不确定性
- ✓ 这种不确定性既体现在同一次分类结果的分类误差中，也体现在不同次分类结果的误差中
- ✓ 归根结底在于：每一次k-均值聚类过程中的中心点初始位置都是随机出来的，有很大不确定性，因此每一次k-均值计算过程的迭代次数、最终分类结果也存在一定程度的不确定性

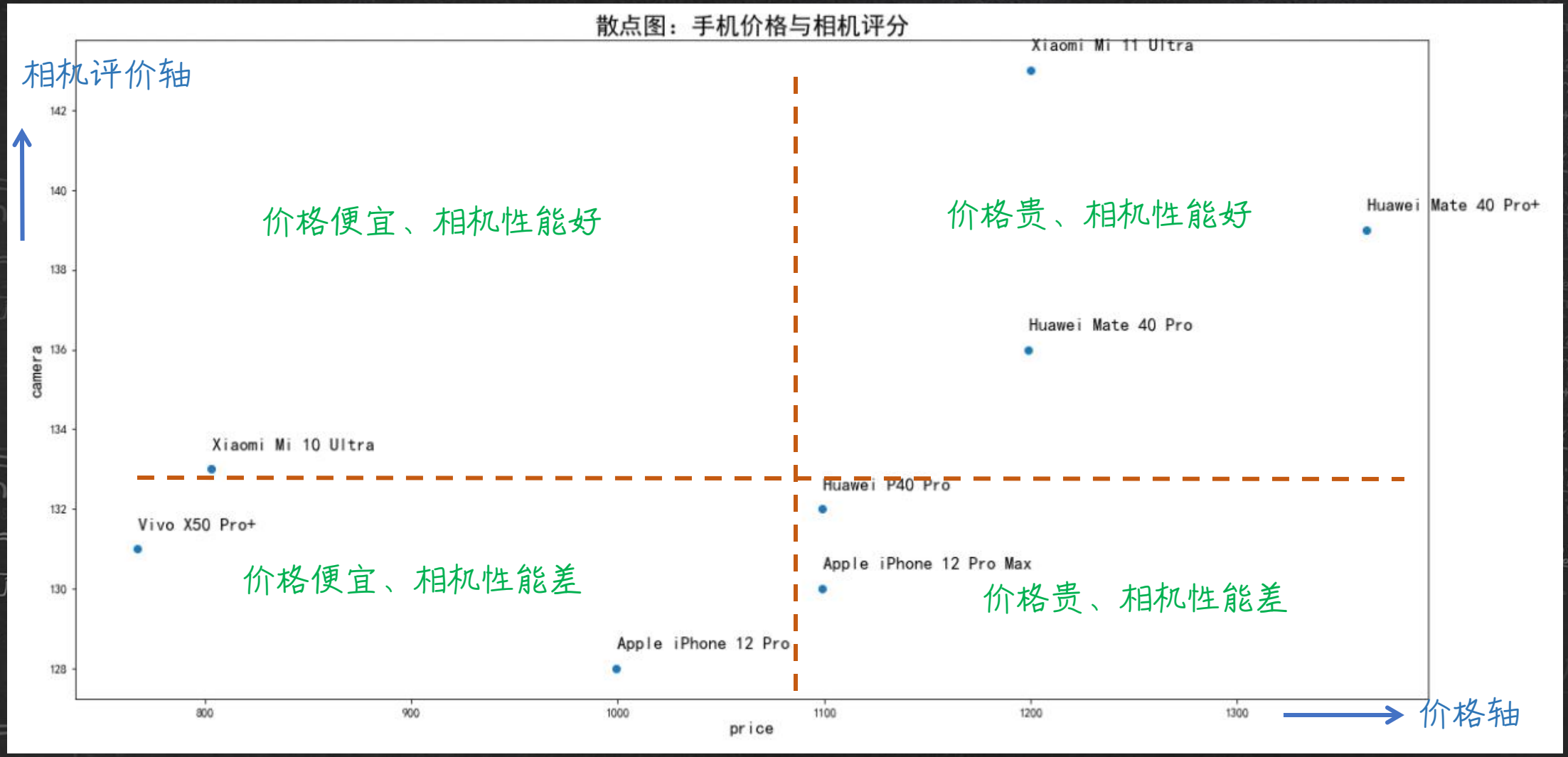
➢ 补充案例

- ✓ 为了进一步观察k-均值这种聚类方法的不确定性，我们可以通过一个更加直观、简单的案例来进行补充说明
- ✓ 补充案例：在小米、华为、苹果等8款手机终端中，针对价格与相机性能，对8款终端进行分类，并找到性价比最高的一类
- ✓ 手机价格及相机评分数据均取自https://www.dxomark.com/rankings/

cellphone	price($)	camera
Xiaomi Mi 11 Ultra	1200	143
Huawei Mate 40 Pro+	1363	139
Huawei Mate 40 Pro	1199	136
Xiaomi Mi 10 Ultra	803	133
Huawei P40 Pro	1099	132
Vivo X50 Pro+	767	131
Apple iPhone 12 Pro Max	1099	130
Apple iPhone 12 Pro	999	128

➢ 手机终端价格与相机评分二维特征图

从直观上看，将二维特征图划分为4个象限，其中xiaomi MI 10 Ultra应该属于性价比相对不错的一款



➢ 通过k-均值法将8款终端分为4类

测试次数	第1类样本集	第2类样本集	第3类样本集	第4类样本集	人工经验判断结果
1	'Xiaomi Mi 11 Ultra'	'Huawei Mate 40 Pro+', 'Apple iPhone 12 Pro Max'	'Xiaomi Mi 10 Ultra', 'Huawei P40 Pro', 'Vivo X50 Pro+', 'Apple iPhone 12 Pro'	'Huawei Mate 40 Pro'	不合理
2	'Vivo X50 Pro+'	'Huawei Mate 40 Pro'	'Xiaomi Mi 11 Ultra', 'Xiaomi Mi 10 Ultra', 'Huawei P40 Pro', 'Apple iPhone 12 Pro Max', 'Apple iPhone 12 Pro'	'Huawei Mate 40 Pro+'	不合理
3	'Huawei P40 Pro', 'Vivo X50 Pro+', 'Apple iPhone 12 Pro Max', 'Apple iPhone 12 Pro'	'Xiaomi Mi 11 Ultra'	'Huawei Mate 40 Pro+', 'Huawei Mate 40 Pro', 'Xiaomi Mi 10 Ultra'	-	相对合理，但少了一个分类
4	'Huawei Mate 40 Pro+', 'Apple iPhone 12 Pro Max', 'Apple iPhone 12 Pro'	'Xiaomi Mi 11 Ultra', 'Xiaomi Mi 10 Ultra', 'Huawei P40 Pro'	'Huawei Mate 40 Pro'	'Vivo X50 Pro+'	不合理
5	'Huawei Mate 40 Pro+', 'Huawei Mate 40 Pro', 'Xiaomi Mi 10 Ultra', 'Huawei P40 Pro', 'Vivo X50 Pro+', 'Apple iPhone 12 Pro Max'	'Apple iPhone 12 Pro'	'Xiaomi Mi 11 Ultra'	-	相对合理，但中间档位的机型没有进行有效区分

补充案例小结

- ✓ 分类结果的不确定性：每次聚类结果都不一样
- ✓ 结果正确与否的不确定性：经过人工经验验证，5次实验中，有3次分类结果相对不太合理，有2次分类结果相对合理

03

样本分布可视化

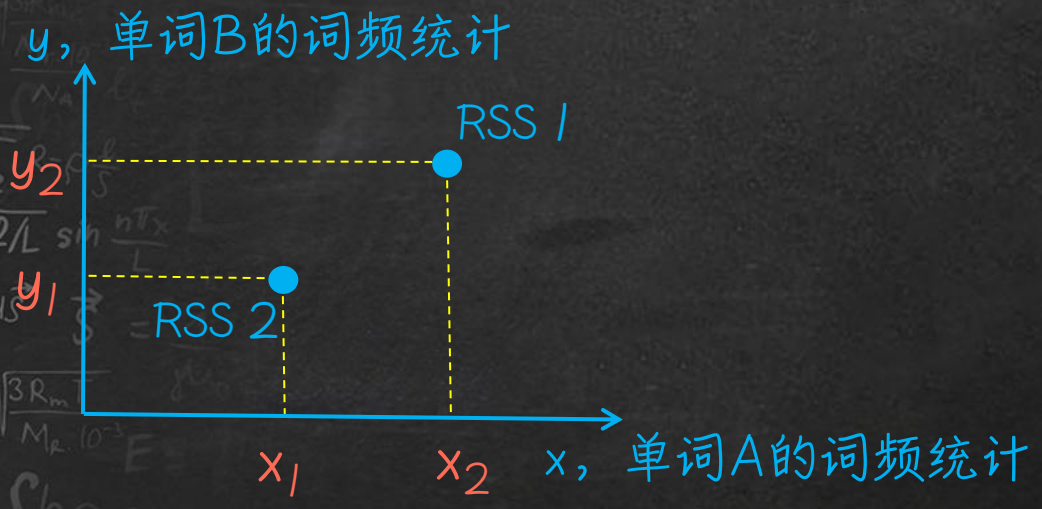
- 二维缩放图，可体现出样本间的远近距离

➢ 二维缩放图

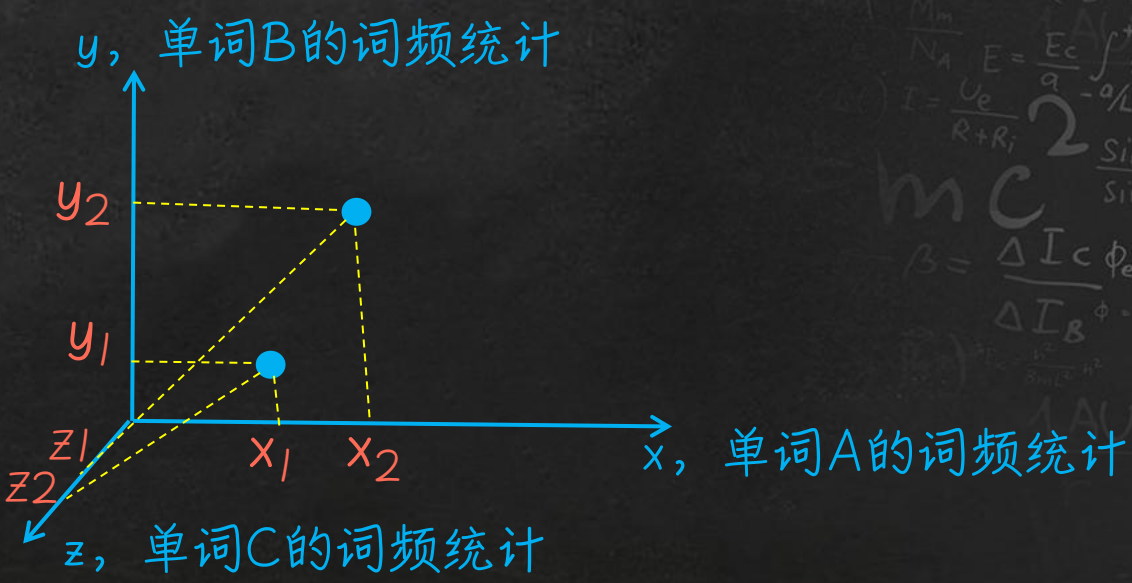
- ➢ 如何在一个二维平面中展现所有样本之间的关系？
- ➢ 还是以上一节当中的英文文本为例
- 样本名称：每个RSS源的英文标题
- 样本特征：对每个RSS源内容的词频统计结果

- ➢ 如果包含n个词，那么需要n维空间来表示
- ➢ 只有当词频统计结果仅包含2个单词才能将两个样本通过一个平面进行展现

仅出现2个单词,2维



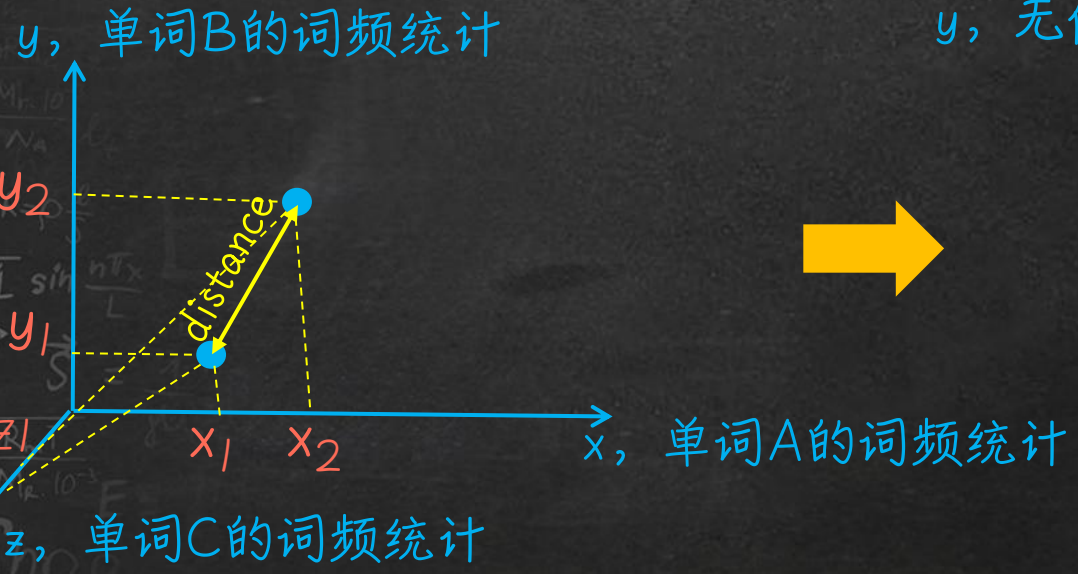
出现3个单词，3维

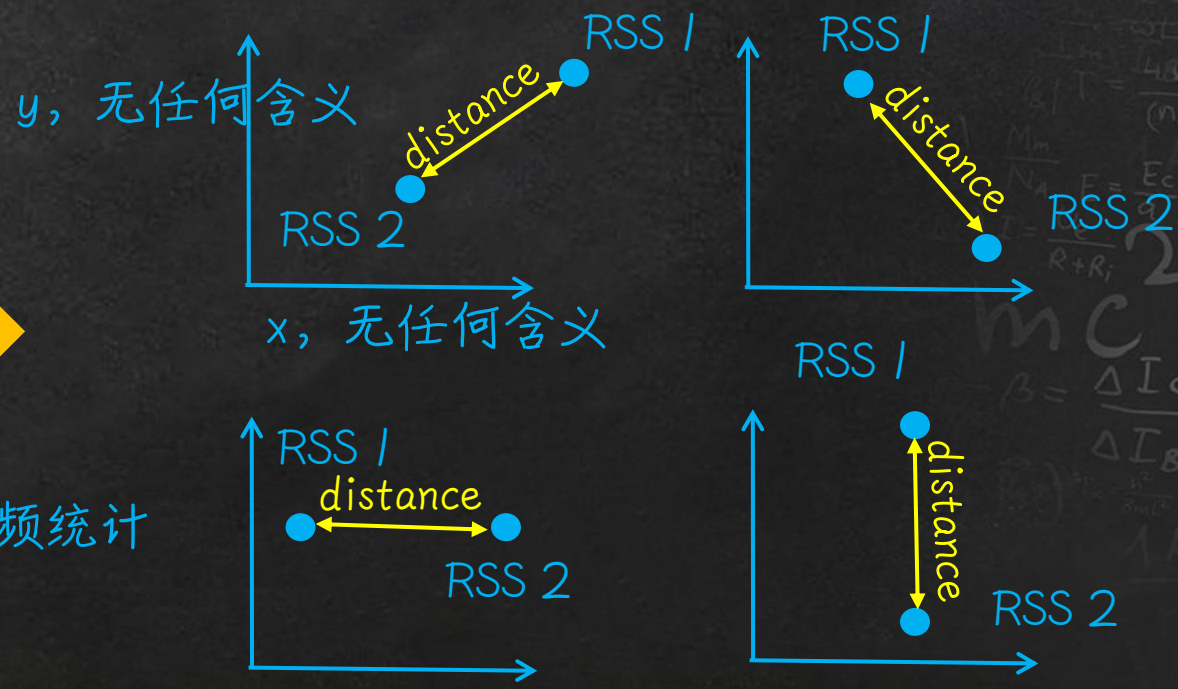


➢ 当词数（特征维度）> 2时，如何继续使用一个二维平面来表示不同样本之间的特征关系？

用x/y轴表示各样本的特征关系，与点本身的位置有关

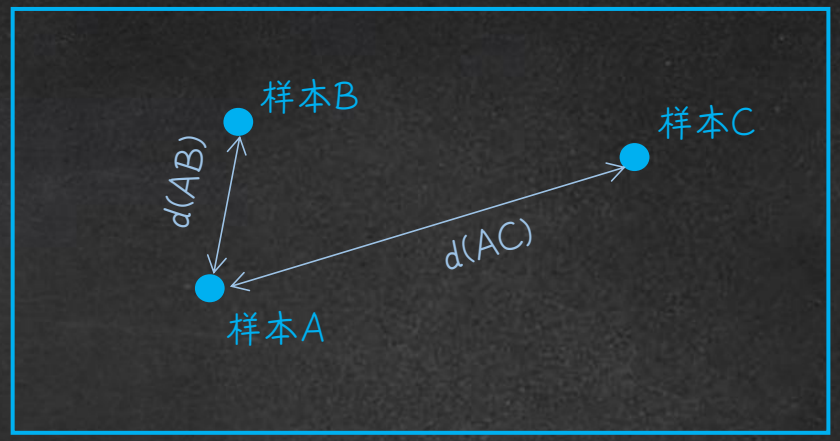
用平面中各点的远近距离表示各样本特征关系，而与点本身的位置无关



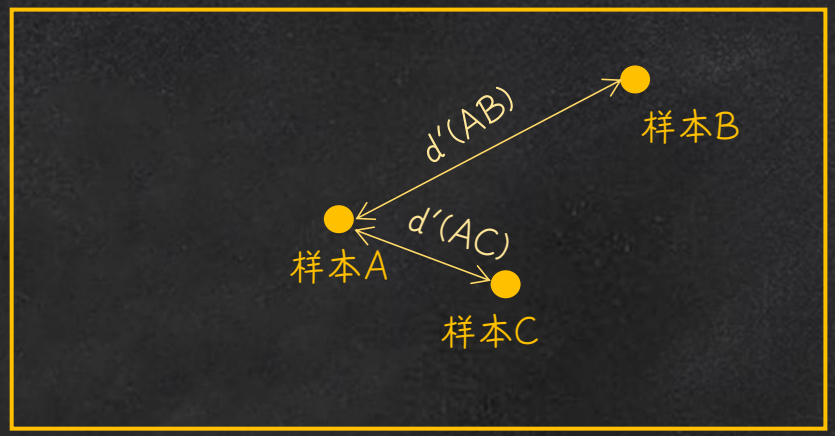


➢ 如何将多维空间中各样本间的距离关系映射到一个平面中，即2维坐标系中？

真实坐标系
（n个特征，n维）



初始化虚拟坐标系
（2维）



- **step 1,真距离:** 得到各样本之间真实距离d
- **step 2,伪距离:** 在平面中初始化各样本位置，并计算初始化样本间距离d'
- **step 3,真伪误差:** 以样本A为例，计算A与其余样本之间d与d'的相对误差，即errorterm = (d-d')/d
- **step 4,x/y偏移:** A若靠近B，求x/y轴的偏移量，

x轴偏移 Δ(AB)x' = ((Ax'-Bx') / d'(AB)) * errorterm

y轴偏移 Δ(AB)y' = ((Ay'-By') / d'(AB)) * errorterm

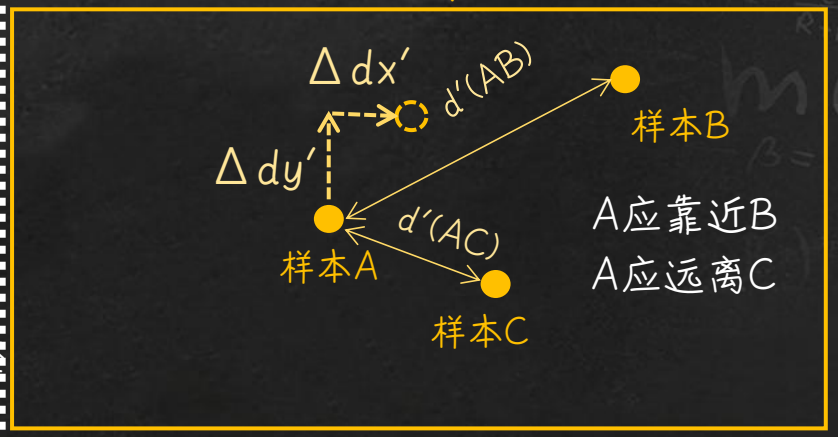
远离C，计算方法类同

- **step 5,补误差:** 对A的应偏移量求和，乘以调整因子后再统一调整

Δ(A)x' = Δ(AB)x' + Δ(AC)x' ; Δ(A)y' = Δ(AB)y' + Δ(AC)y'

A''x = Ax' - 0.01 * Δ(A)x' ; A''y = Ay' - 0.01 * Δ(A)y'

调整后虚拟坐标系
（2维）



定位置

样本在坐标系中的位置

定距离

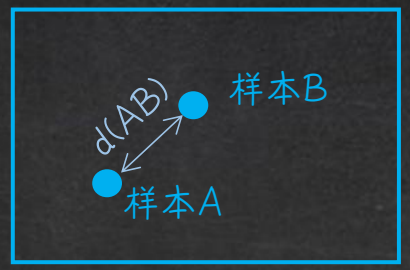
两两样本间的距离矩阵

求误差

距离误差

x/y等比例误差

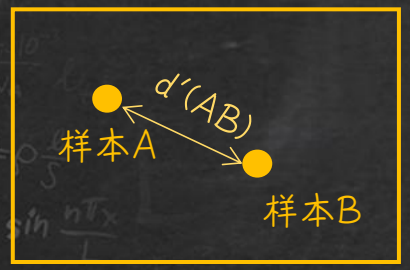
真实坐标系



	A	B	C
A	0	0.9	0.6
B	0.9	0	0.3
C	0.6	0.3	0

位置由样本的各维度特征决定 样本间真实距离采用pearson算法
由realdist矩阵表示，(n,n)

虚拟坐标系



	A	B	C
A	0	0.3	0.2
B	0.3	0	0.7
C	0.2	0.7	0

初始位置随机生成
由loc表示，(n,2)，比如
(0.5,0.8)
(0.2,0.5)
.....

样本间虚拟距离采用投影算法（欧式）
由fakedist矩阵表示，(n,n)

真实/虚拟距离间
相对误差
errorterm =
(df - dt)/dt

x轴移动，grad[C][0]
(loc(C)-loc(B)/fakedist[B][C])
*
errorterm

x轴移动，grad[C][1]
(loc(C)-loc(B)/fakedist[B][C])
*
errorterm

补误差

Σ → 位置偏移

调整后坐标 loc[C] = loc[C] - rate * grad[C]

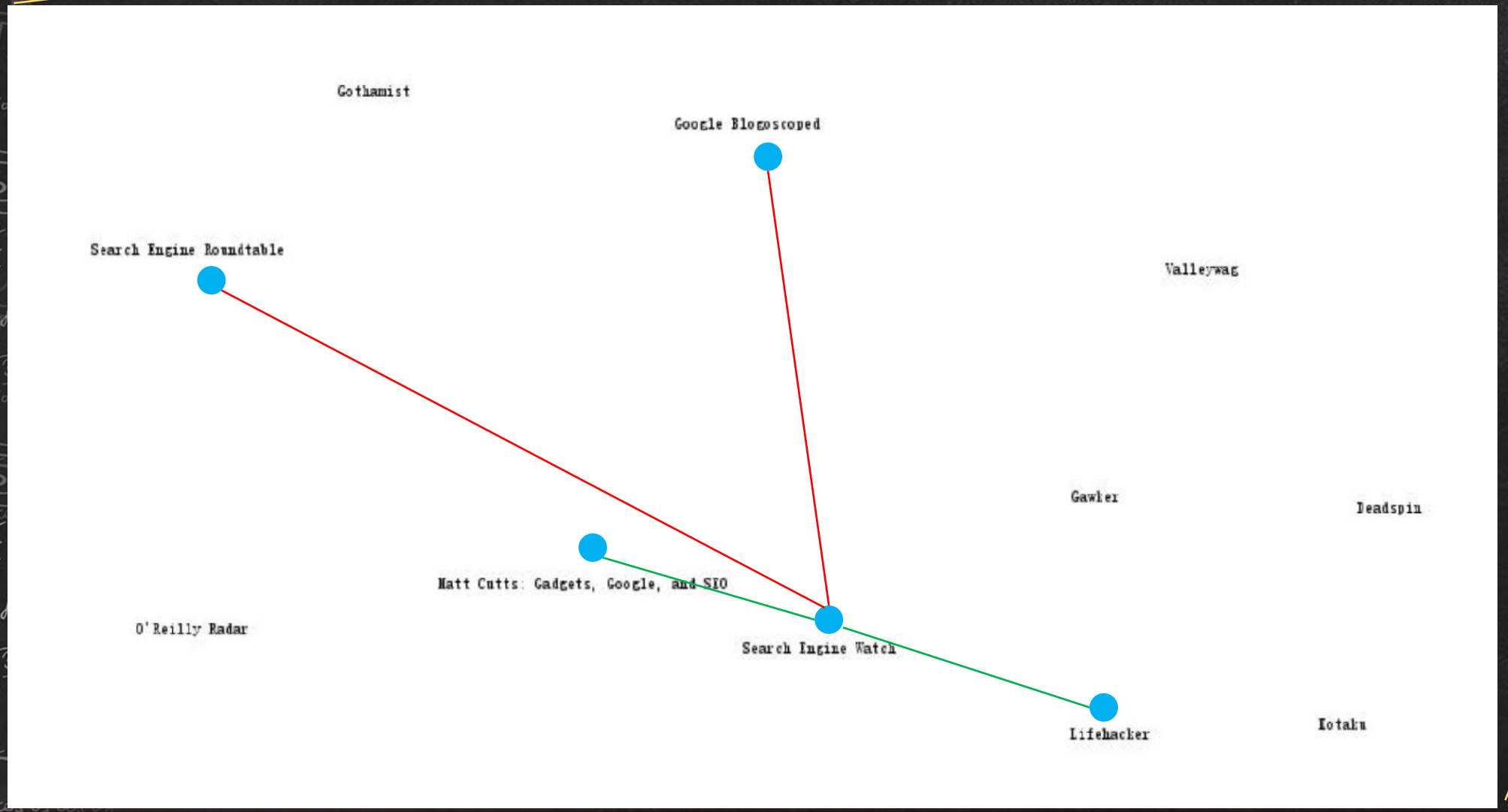
↓

调整因子，0.01

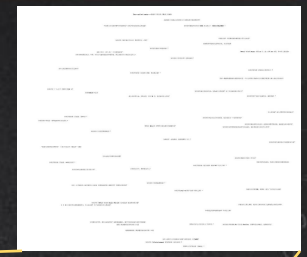
➢ 绘制二维缩放图的函数清单

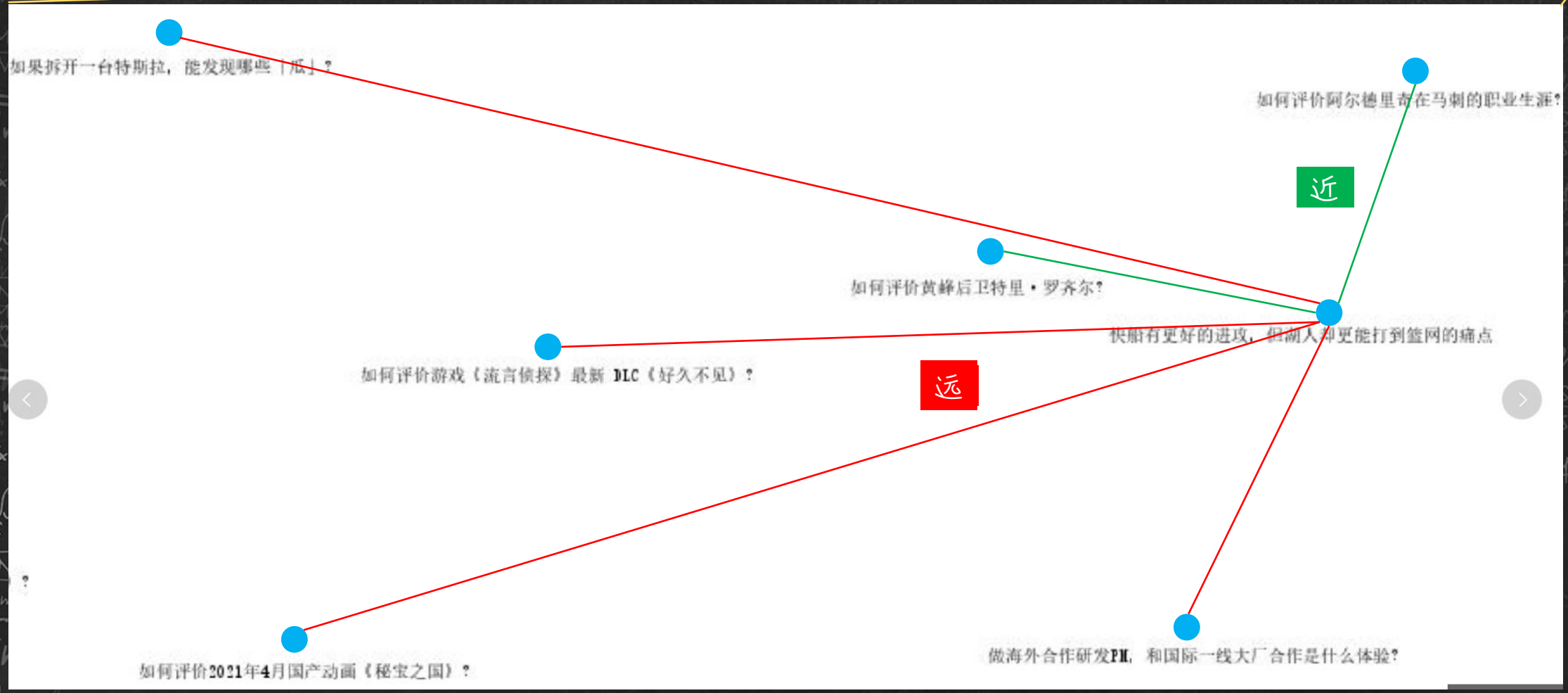
代码文件	函数	参数	输出
clusters.py	scaledown(data,distance = pearson, rate = 0.01)	-data, m(样本数) x n(分词数) -distance, 样本间距离算法，默认使用person -rate,调整因子，默认为0.01	loc, m(样本数) x 2(x/y轴) 即调整后样本的新坐标
	draw2d(data, labels, jpeg = 'mds2d.jpg')	-data, m(样本数) x n(分词数) -labels, 样本名称 -jpeg, 生成图片保存路径	二维缩放图片

➢ 实践案例1：英文博客样本的二维展现



- 实践案例2：中文知乎RSS源样本的二维展现





“人生之旅历途甚长，所争决不在一年半日，
万不可因此着急失望，招精神上之萎靡”

——梁启超致梁思成家书
献给正在逐梦路上努力奔跑的你我他

感谢聆听

THANK YOU

本次胶片内容、及涉及相关代码均可移步至Github进行下载
感谢您的投币三连！

我的代码 Github 地址：
https://github.com/david-cal/
Reading-Note-For-Programming-Collective-Intelligence