

# 信息检索导论

## An Introduction to Information Retrieval

### 第14讲 扁平聚类

### Flat Clustering

授课人：古晓艳

中国科学院信息工程研究所/国科大网络空间安全学院

\*改编自“An introduction to Information retrieval”网上公开的课件，地址 <http://nlp.stanford.edu/IR-book/>

# 提纲

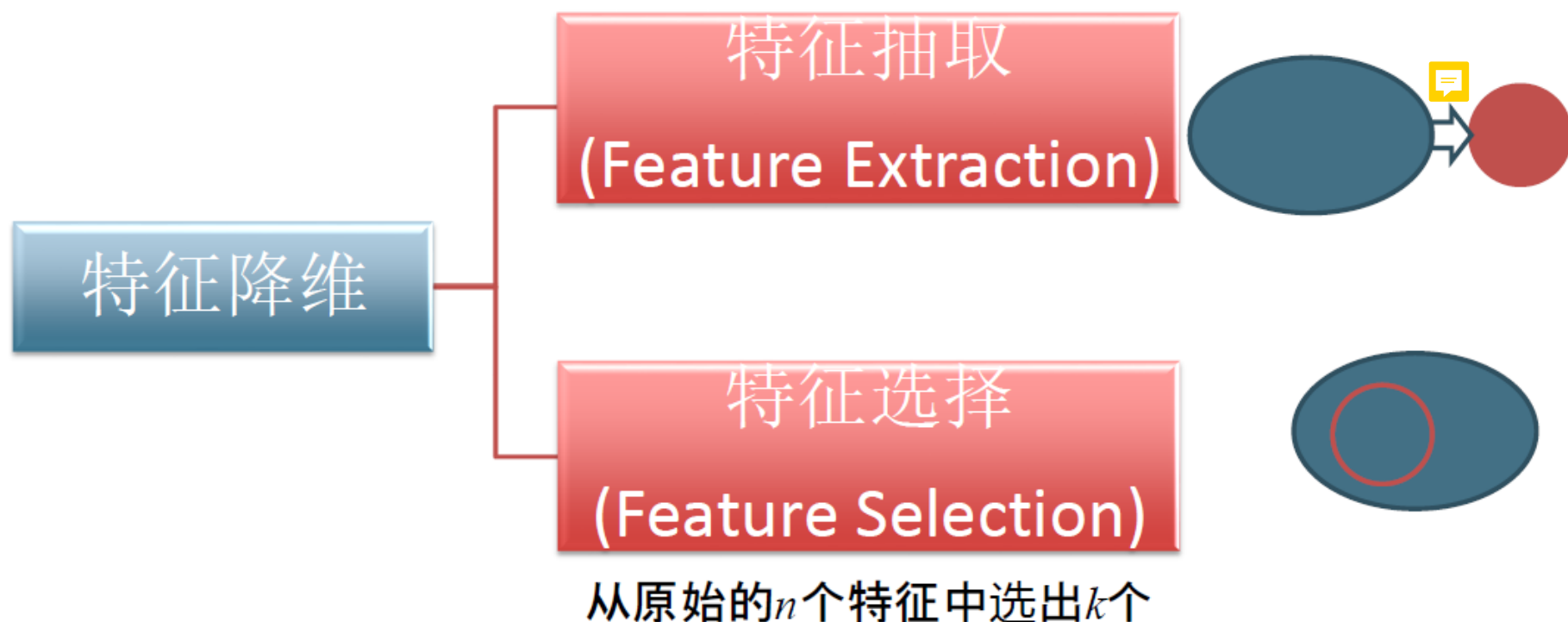
- ① 上一讲回顾
- ② 聚类介绍
- ③ 聚类在IR中的应用
- ④ K-均值聚类算法
- ⑤ 聚类评价
- ⑥ 簇个数确定
- ⑦ 深度聚类

# 提纲

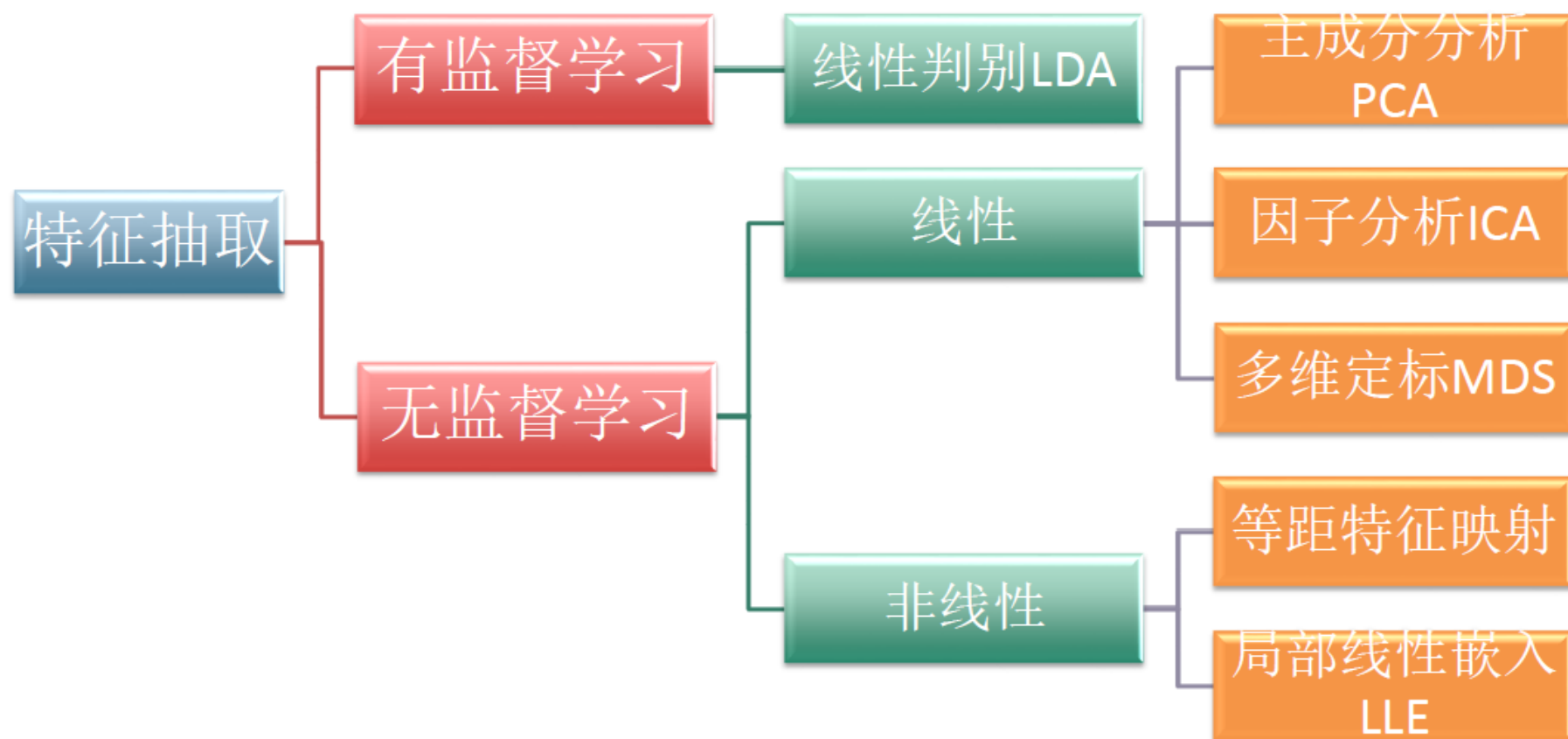
- ① 上一讲回顾
- ③ 聚类介绍
- ③ 聚类在IR中的应用
- ④ K-均值聚类算法
- ⑤ 聚类评价
- ⑥ 簇个数确定
- ⑦ 深度聚类

# 特征降维的种类

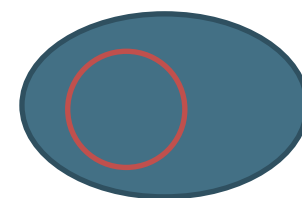
**特征抽取**和**特征选择**都是从原始特征中找出最有效（同类样本的不变性、不同样本的鉴别性、对噪声的鲁棒性）的特征。



# 特征抽取



# 特征选择



- 文本分类中，通常要将文本表示在一个高维空间下，每一维对应一个词项
- 本讲义中，我们不特意区分不同的概念：每个坐标轴 = 维 = 词语 = 词项 = 特征
- 许多维上对应是罕见词
- 罕见词可能会误导分类器
- 这些会误导分类器的罕见词被称为噪音特征（noise feature）
- 去掉这些噪音特征会同时提高文本分类的效率和效果
- 上述过程称为特征选择（feature selection）

# Reuters 语料中 *poultry*/EXPORT 的 MI 计算

	$e_c = e_{\text{poultry}} = 1$	$e_c = e_{\text{poultry}} = 0$
$e_t = e_{\text{export}} = 1$	$N_{11} = 49$	$N_{10} = 27\ 652$
$e_t = e_{\text{export}} = 0$	$N_{01} = 141$	$N_{00} = 774\ 106$

$$I(U; C) =$$

$$\begin{aligned} & \frac{N_{11}}{N} \log_2 \frac{NN_{11}}{N_{1.}N_{.1}} + \frac{N_{01}}{N} \log_2 \frac{NN_{01}}{N_{0.}N_{.1}} \\ & + \frac{N_{10}}{N} \log_2 \frac{NN_{10}}{N_{1.}N_{.0}} + \frac{N_{00}}{N} \log_2 \frac{NN_{00}}{N_{0.}N_{.0}} \end{aligned}$$

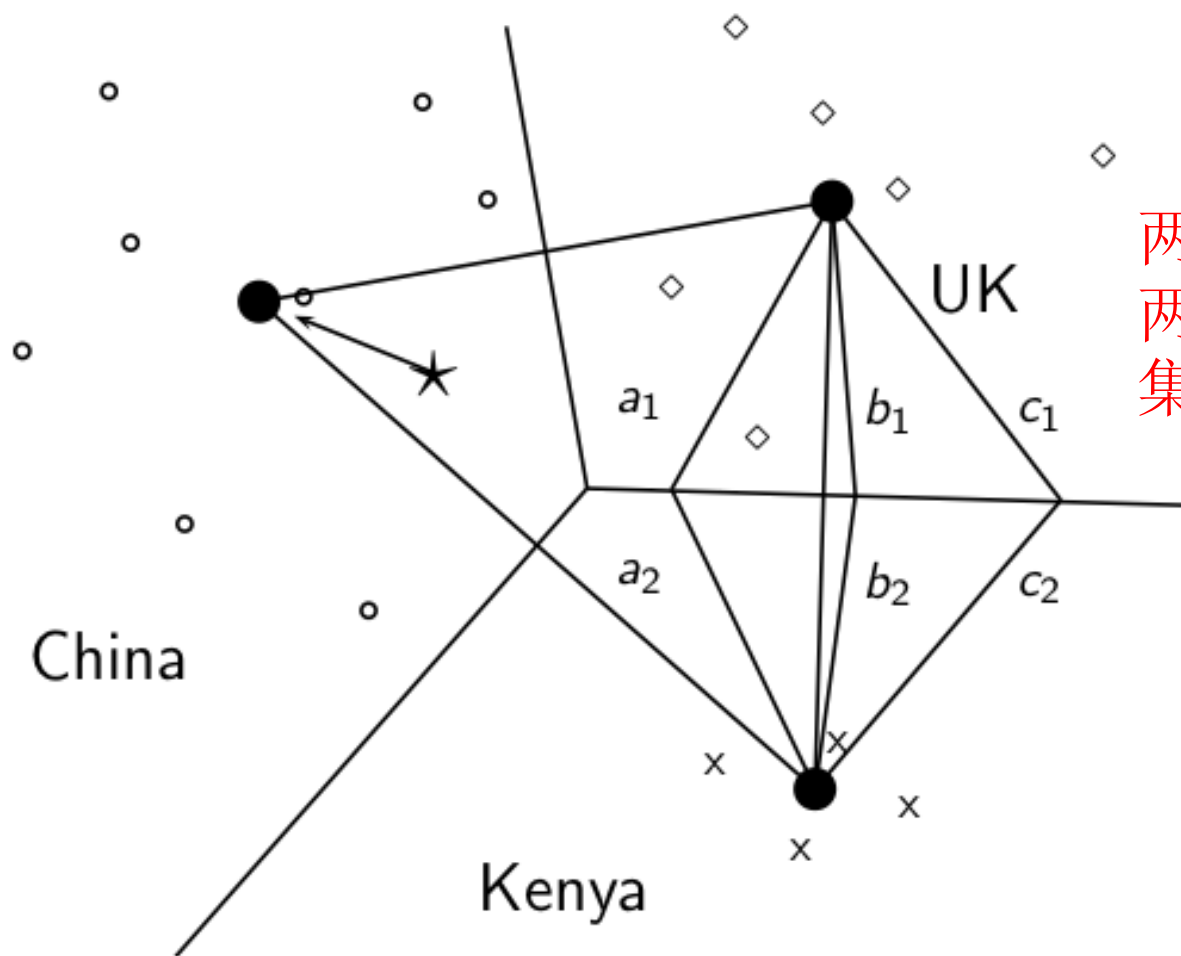
$$\begin{aligned} I(U; C) = & \frac{49}{801\ 948} \log_2 \frac{801\ 948 \times 49}{(49 + 27\ 652)(49 + 141)} \\ & + \frac{141}{801\ 948} \log_2 \frac{801\ 948 \times 141}{(141 + 774\ 106)(49 + 141)} \\ & + \frac{27\ 652}{801\ 948} \log_2 \frac{801\ 948 \times 27\ 652}{(49 + 27\ 652)(27\ 652 + 774\ 106)} \\ & + \frac{774\ 106}{801\ 948} \log_2 \frac{801\ 948 \times 774\ 106}{(141 + 774\ 106)(27\ 652 + 774\ 106)} \\ & \approx 0.000\ 110\ 5 \end{aligned}$$

# 向量空间分类

- 同前面一样，训练集包含一系列文档，每篇都标记着它的类别
- 在向量空间分类中，该集合对应着空间中一系列标记的点或向量。
- 假设 1: 同一类中的文档会构成一片连续区域（contiguous region）
- 假设2: 来自不同类别的文档没有交集
- 定义直线、平面、超平面来将上述不同区域分开

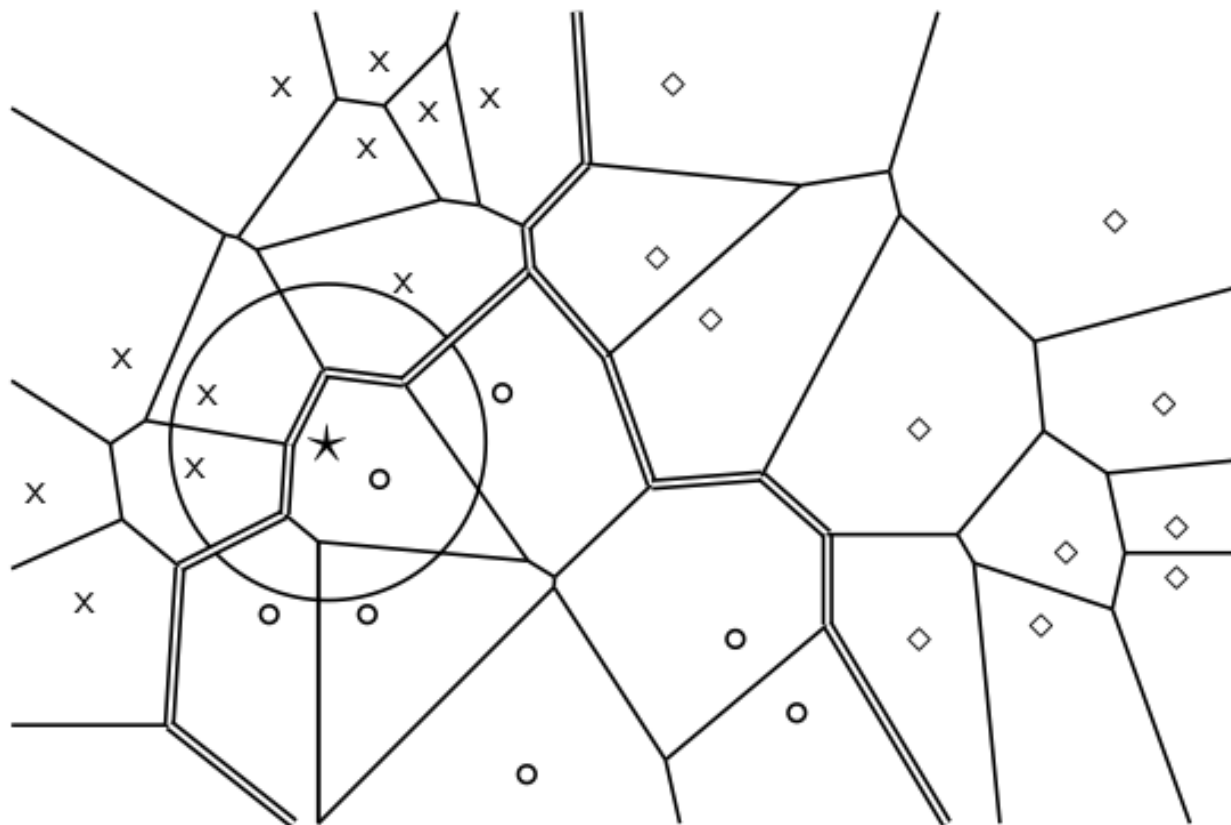


# Rocchio算法示意图 : $a_1 = a_2, b_1 = b_2, c_1 = c_2$



两类的边界由那些到  
两个类质心等距的点  
集组成

# kNN算法

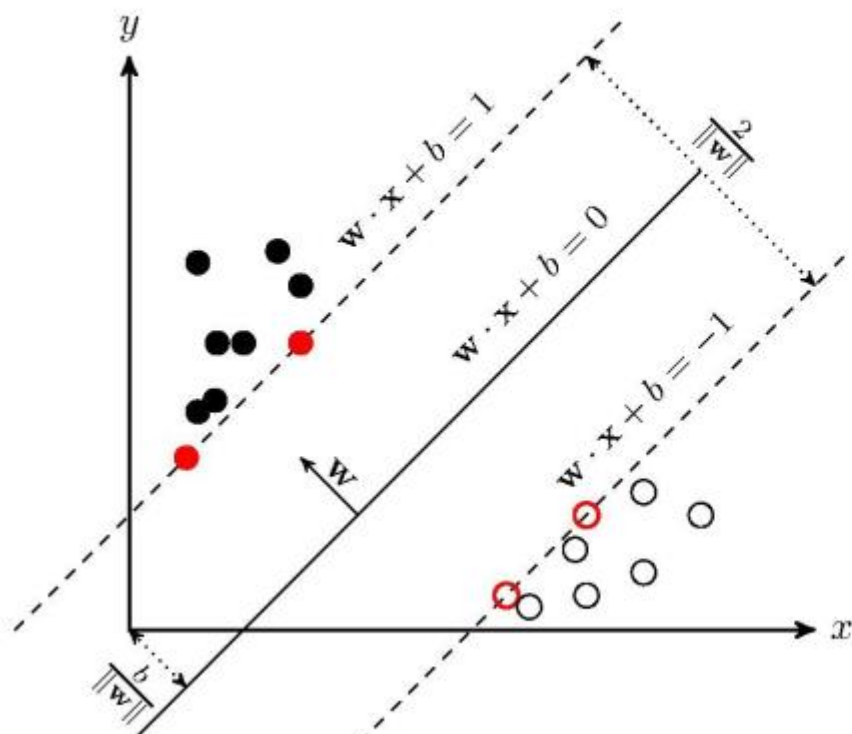


对于★ 对应的文档，在1NN 和 3NN下，分别应该属于哪个类？

# 超平面定义

在线性可分的情况下, 存在多个超平面(Hyperplane) 使得这两类被无误差的完全分开。超平面可以定义为:

$$W \bullet X + b = 0 \quad W, b \text{ 分别为超平面的法向量和截距}$$



$$\begin{aligned} w^T X_i + b &\geq +1, \Rightarrow y_i = +1 \\ w^T X_i + b &\leq -1, \Rightarrow y_i = -1 \end{aligned}$$

所有在上间隔边界上方的样本属于正类,  
在下间隔边界下方的样本属于负类。

两个间隔边界的距离  $d = \frac{2}{\|w\|}$  被定义为  
边距 (margin)

# 支持向量机 (SVM)

- 最优超平面是指两类的分类间隔 (Margin) 最大，即每类距离超平面最近的样本到超平面的距离之和最大。距离这个最优超平面最近的样本被称为支持向量 (Support Vector)。
- 优化问题：

目标函数：

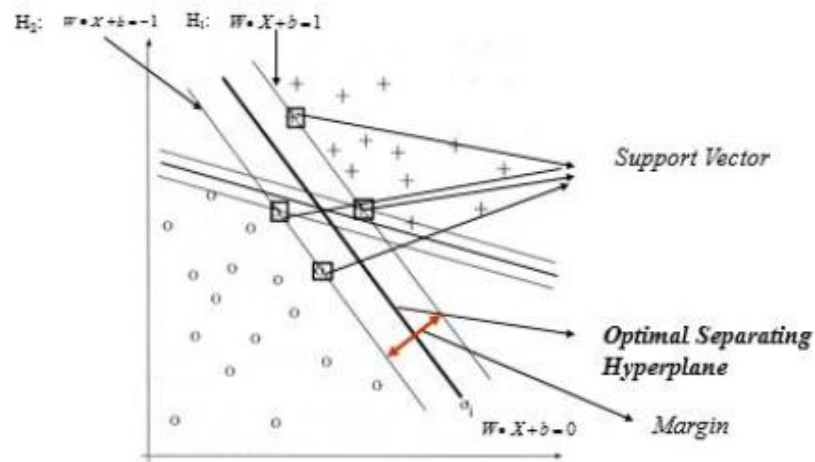
$$\frac{2}{\|W\|}$$

约束条件：

$$\left. \begin{array}{l} W \cdot X_1 + b \geq 1 \\ W \cdot X_2 + b \leq -1 \end{array} \right\} y_i [(W \cdot X_i) + b] - 1 \geq 0$$


将最大化问题转化为最小化问题：

$$\max_{w,b} \frac{2}{\|w\|} \longrightarrow \min_{w,b} \frac{1}{2} \|w\|^2$$



# 求解结果

- 上述二次优化问题，采用Lagrange方法求解，可得

$$f(X) = \text{sgn} \left( \sum_{i=1}^n \alpha_i^* y_i X \bullet \boxed{X_i} + b^* \right)$$


$\alpha^*$  不为零的这些训练点的输入为支持向量(Support Vector)

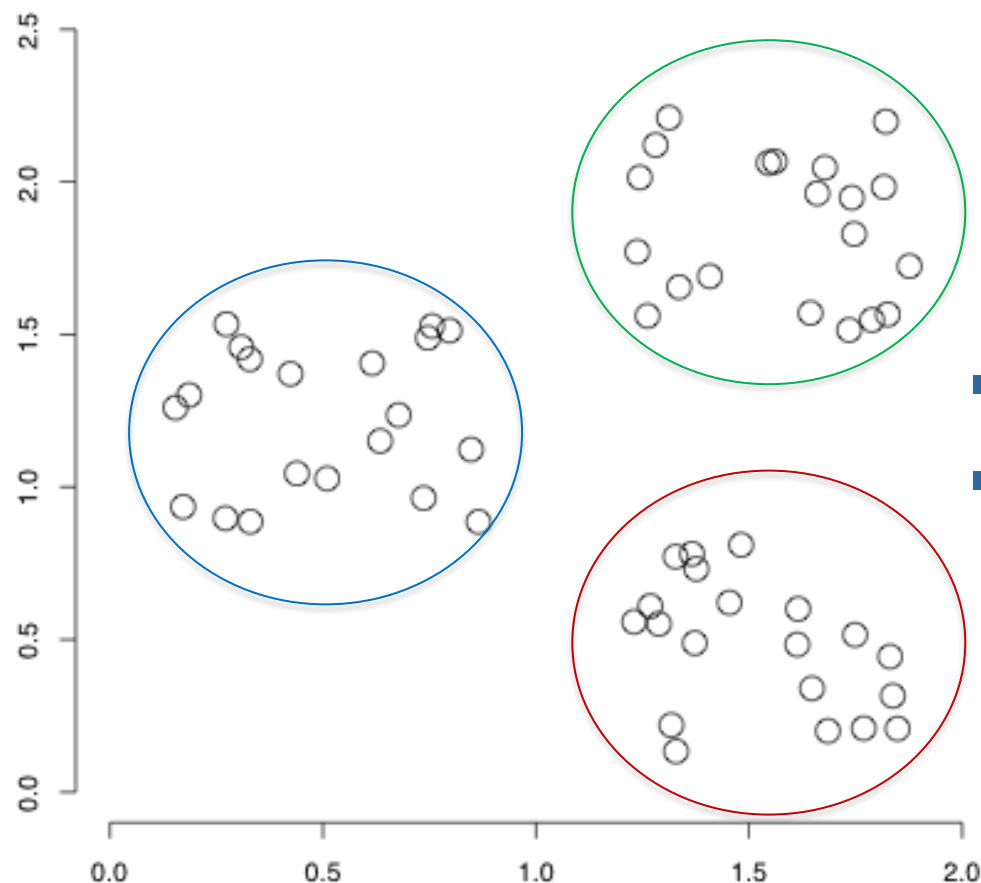
- 相当于每个类别中选出若干支持向量组成“投票委员会”，根据这些“委员”的加权投票(内积)结果得到最终的类别归属

# 提纲

- ① 上一讲回顾
- ② 聚类介绍
- ③ 聚类在IR中的应用
- ④ K-均值聚类算法
- ⑤ 聚类评价
- ⑥ 簇个数确定
- ⑦ 深度聚类

# 什么是聚类 (Clustering)

- 聚类是将一系列对象按照相似性聚团成子集或者簇(cluster)的过程



- 簇内文档之间彼此相似
- 簇间文档之间相似度不大



# 为什么要聚类？

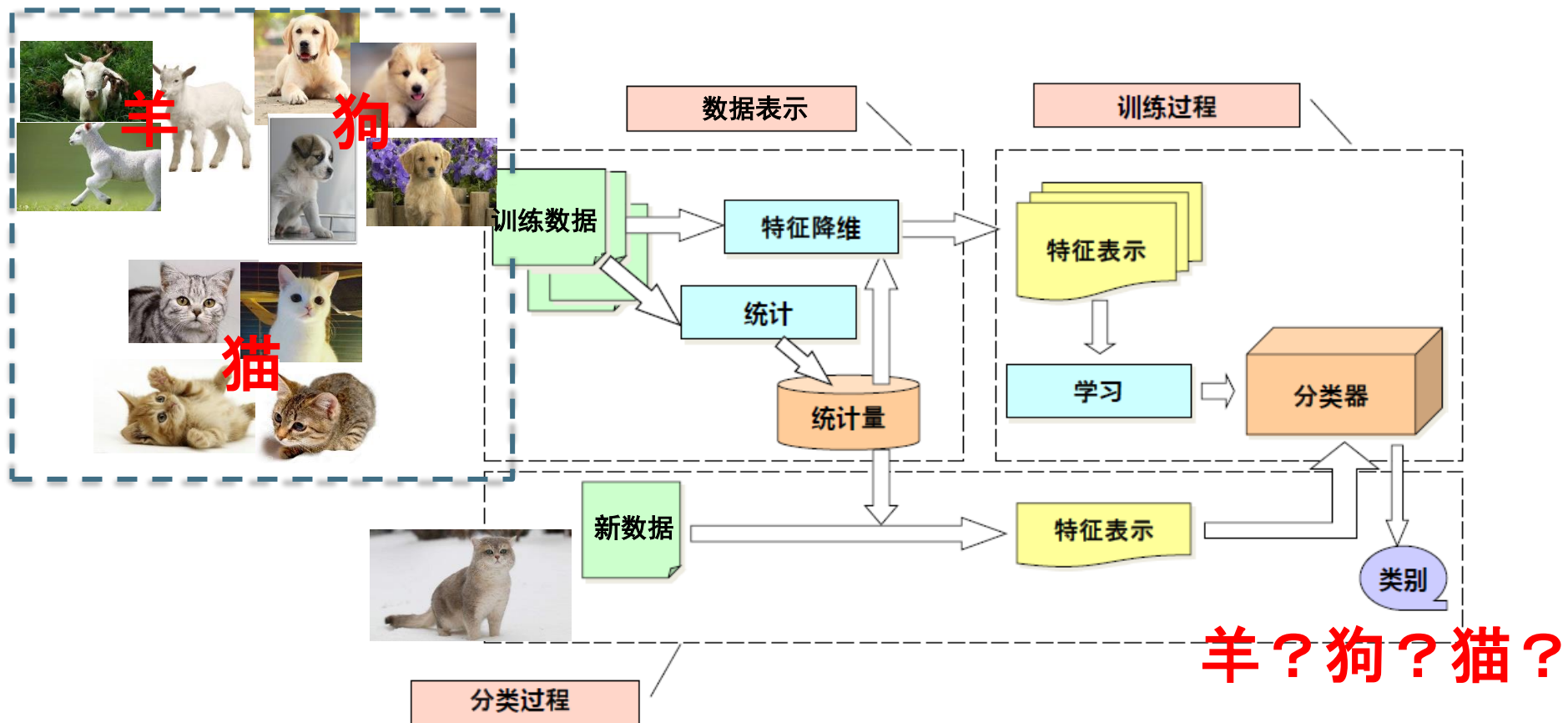
- 人类社会的固有现象：物以类聚、人以群分
  - 相似的对象往往聚集在一起
  - (相对而言)不相似的对象往往分开



- 方便处理！

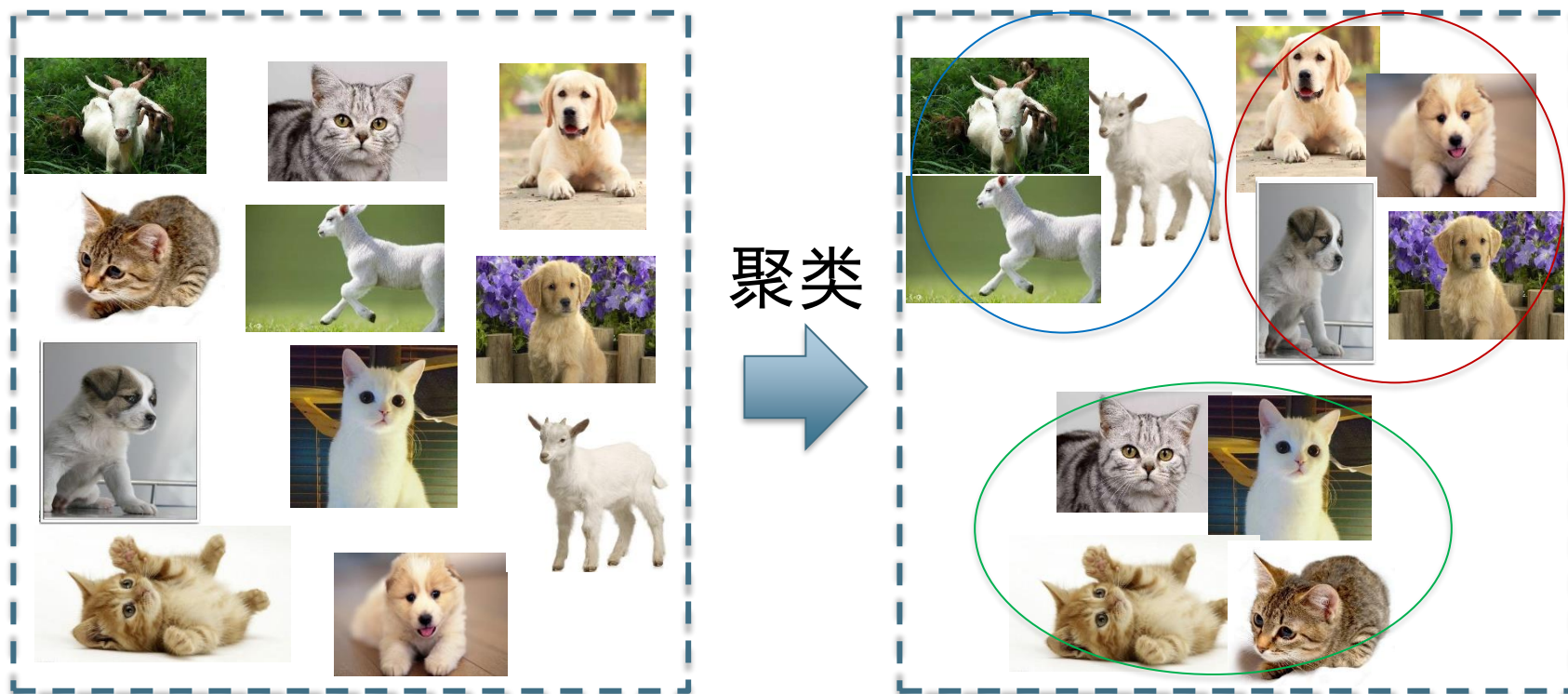


# 分类流程



- 分类是**有监督**机器学习的一种

# 聚类示例



- 聚类是一种最常见的**无监督**学习(unsupervised learning)方法
- 无监督意味着没有已标注好的数据集

# 分类 vs. 聚类

- 分类: 有监督的学习
- 聚类: 无监督的学习
- 分类: 类别事先人工定义好, 并且是学习算法的输入的一部分
- 聚类: 簇在没有人工输入的情况下从数据中推理而得
  - 但是, 很多因素会影响聚类的输出结果: 簇的个数、相似度计算方法、文档的表示方式, 等等

# 提纲

- ① 上一讲回顾
- ② 聚类介绍
- ③ 聚类在IR中的应用
- ④ K-均值聚类算法
- ⑤ 聚类评价
- ⑥ 簇个数确定
- ⑦ 深度聚类

# 聚类假设

**聚类假设：**在考虑文档和信息需求之间的相关性时，同一簇中的文档表现互相类似。

聚类在IR中的所有应用都直接或间接基于上述聚类假设

Van Rijsbergen的原始定义：“closely associated documents tend to be relevant to the same requests”（彼此密切关联的文档和同一信息需求相关）

# C. J. van Rijsbergen

C. J. "Keith" van Rijsbergen (Cornelis Joost van Rijsbergen) (born 1943) is a professor of computer science and the leader of the Glasgow Information Retrieval Group based at the University of Glasgow. He is **one of the founders of modern Information Retrieval** and the author of the seminal monograph Information Retrieval and of the textbook The Geometry of Information Retrieval.

In 2003 he was inducted as a Fellow of the Association for Computing Machinery. In 2004 he was awarded the Tony Kent Strix award. In 2006, he was awarded the Gerard Salton Award for Quantum haystacks.

[http://en.wikipedia.org/wiki/C.\\_J.\\_van\\_Rijsbergen](http://en.wikipedia.org/wiki/C._J._van_Rijsbergen)

<http://www.dcs.gla.ac.uk/~keith/>



# 思考

- 搜索引擎中哪些功能可能用到聚类？

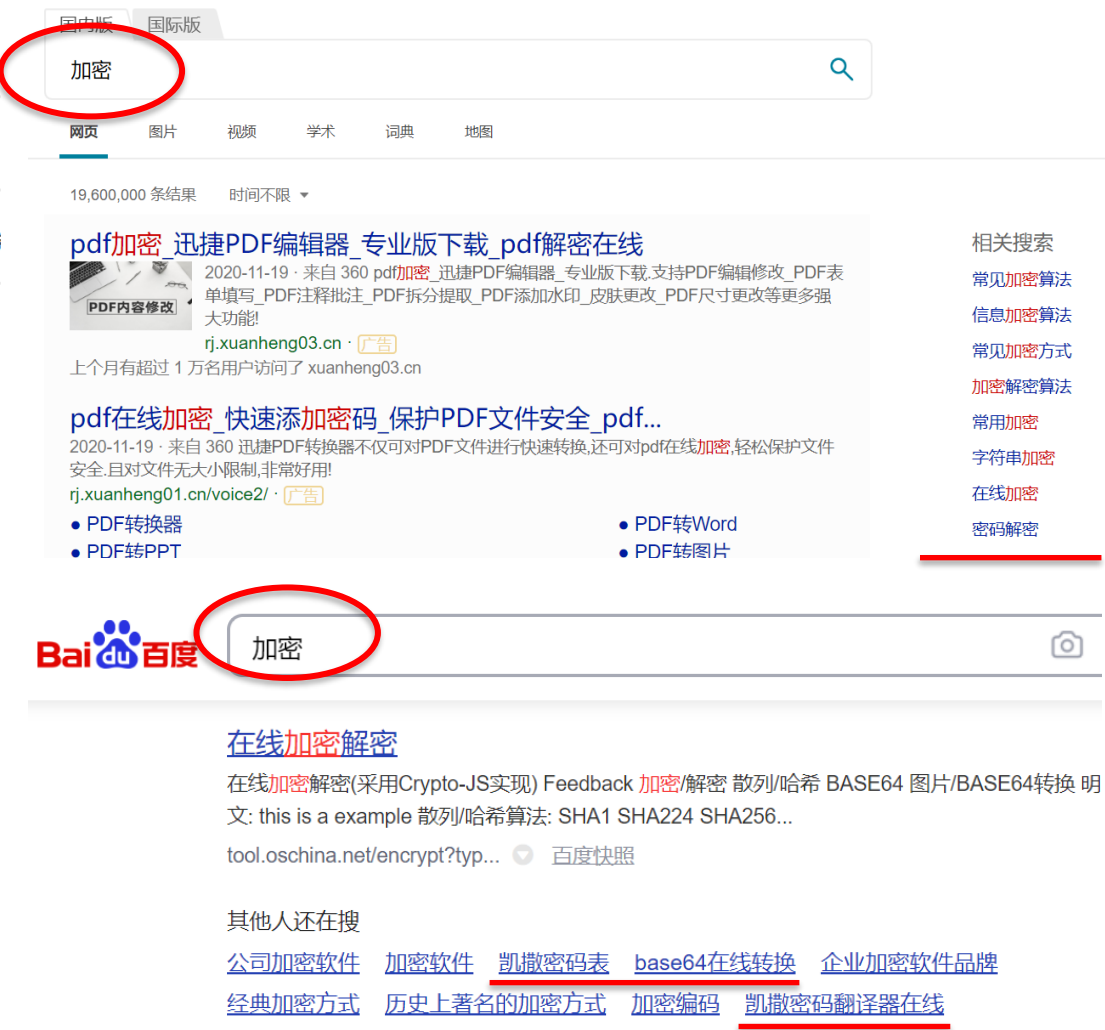


# 查询的聚类：发现用户的潜在意图



## 国科大的相关搜索

- 国科大官网
- 国科大排名
- 国科大图书馆
- 中科院研究生院
- 中国科学院大学邮箱
- 中国科学院大学邮箱登陆
- 中国科学院大学综合信息网
- 中国科学院大学招生信息网





# 搜索结果的聚类：更好地浏览

The screenshot shows the Vivísimo search engine interface. The search bar contains the query "jaguar", which is circled in red. To the right of the search bar is a dropdown menu set to "the Web" and a "Search" button. Above the search bar are links for "Advanced Search" and "Help". Below the search bar, a yellow banner displays "Top 208 results of at least 20,373,974 retrieved for the query jaguar (Details)".

On the left side, there is a "Clustered Results" panel. It lists various clusters with expandable arrows and counts: "jaguar (208)", "Cars (74)", "Club (34)", "Cat (23)", "Animal (13)", "Restoration (10)", "Mac OS X (8)", "Jaguar Model (6)", "Request (5)", "Mark Webber (5)", and "Maya (5)". This entire panel is circled in red. Below this panel is a "Find in clusters:" section with a text input field labeled "Enter Keywords" and a red search button.

On the right side, the main results area shows a list of search results. The first result is "Jag-lovers - THE source for all Jaguar information" with a description and source information. The second result is "Jaguar Cars" with a note that it is redirected to www.jaguar.com. The third result is "http://www.jaguar.com/" with source information. The fourth result is "Apple - Mac OS X" with a description and source information.

返回结果中前面的文档并没有覆盖jaguar作为动物的那个词义。但是用户很容易通过在Clustered Results面板中点击Cat簇得到该类结果

# 搜索结果的聚类：更好地浏览

The screenshot shows the dblp search interface. At the top, there's a navigation bar with 'home', 'browse', 'search', and 'about'. The dblp logo is on the left, and a search bar on the right contains the text 'wei zhang'. Below the search bar, the page title is 'Search dblp' with a subtext 'powered by CompleteSearch, courtesy of Hannah Bast, University of Freiburg'. The main content area is titled 'Author search results' and shows a list of 'Exact matches' and 'Likely matches'. The 'Exact matches' list includes several entries for 'Wei Zhang' with different IDs and affiliations. The 'Likely matches' list includes 'Weiming Zhang' and 'Weinan Zhang'. A red circle highlights the 'Exact matches' section. At the bottom, there's a 'Publication search results' section showing 'found 27,219 matches' and a 'Refine list' section with 'refine by author' and 'Wei Zhang (1,386)'.

home | browse | search | about

dblp computer science 50000000

wei zhang

[+] Search dblp [-] powered by CompleteSearch, courtesy of Hannah Bast, University of Freiburg

> Home

[+] Author search results [-]

Exact matches

- Wei Zhang — disambiguation page
- Wei Zhang 0001 University of New South Wales, Sydney, NSW, Australia
- Wei Zhang 0002 Virginia Commonwealth University, Compiler, Architecture, and Realtime Systems Lab, Richmond, VA, USA
- Wei Zhang 0003 Apple, Santa Clara, CA, USA
- Wei Zhang 0004 Peking University, Institute of Software, Beijing, China
- show all

Likely matches

- Weiming Zhang aka: Wei Ming Zhang
- Weinan Zhang 0001 Shanghai Jiao Tong University, John Hopcroft Center for Computer Science, China

show all 424 matches

[+] Publication search results [-]

found 27,219 matches

2020

[+] Refine list

refine by author

Wei Zhang (1,386)

# 基于聚类的检索：加快搜索速度



特朗普



百度一下

## 特朗普的最新相关信息

[终于服软!特朗普选择“接受现实”](#) 腾讯新闻

3分钟前

据CNN报道显示,特朗普现如今正在纠结到底要不要奋力一搏的时候,却突然面临了“选择困难”。

因为特朗普的孩子们突然向父亲提出矛盾建议,至此,特朗普家族的内部也突然...

[特朗普悲愤当初看错人,自己一夜白头,英法日却火...](#) 腾讯新闻

1小时前

[美媒披露:特朗普私下知道自己输了,拖延过渡只为...](#) 环球时报

1小时前

[佩洛西连任,特朗普卸任,这两个冤家终于分出了胜负](#) 网易

1小时前

[让特朗普失望了,亲密盟友第一个站出来力挺中方,...](#) 网易

36分钟前

## 特朗普(美国第45任总统) - 百度百科



生日：1946年6月14日

代表作品：做生意的艺术，学徒

简介：[特朗普](#)一般指唐纳德·特朗普。唐纳德·特朗普（Donald Trump，1946年6月14日- ），出生于美国纽约，祖籍德国巴伐利亚，德裔...

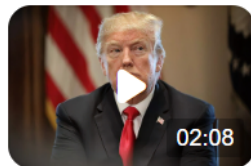
[人物经历](#) [为政举措](#) [商业成就](#) [个人作品](#) [个人荣誉](#) [更多>](#)

[baike.baidu.com/](https://baike.baidu.com/)

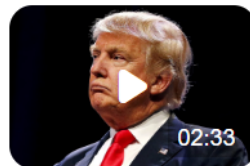
## 特朗普 - 视频大全 - 高清在线观看



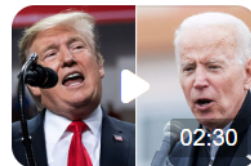
已列最后关头 特朗普



两艘把士的?特朗普



该来的还是来了?于



万万没想到 特朗普

# 文档聚类用于提高召回率

- 为提高搜索召回率:
  - 可以实现将文档集中的文档进行聚类
  - 当文档 $d$ 和查询匹配时，也返回包含 $d$ 的簇所包含的其它文档
  - 我们希望通过上述做法，在输入查询“car”时，也能够返回包含“automobile”的文档
  - 由于聚类算法会把包含“car”的文档和包含“automobile”的文档聚在一起
  - 两种文档都包含诸如“parts”、“dealer”、“mercedes”和“road trip”之类的词语

# 全局浏览的例子: Google News



新闻

添加栏目

个性化谷歌新闻

更新时间: 5分钟前

中国版 (China)

焦点报道

国际/港台

内地

财经

娱乐

科技

互联网

体育

社会

汽车

房产

教育

热门报道

所有内容

新闻标题

焦点报道



事业单位公开招聘 既要招到合适的人 又要公平公正

凤凰网 - 58分钟前

中广网北京 11月25日消息 (记者侯艳) 据中国之声《央广新闻》报道, 国务院法制办昨天 (24日) 公布《事业单位人事管理条例 (征求意见稿) 》, 面向社会各界征求意见, 其中有关公开招聘的内容引人关注。...

我国拟规范事业单位工资福利 规范薪资与社保 搜狐

事业单位公开招聘: 既要招到合适的人 又要公平公正 中国新闻网

新浪网 - 新华网 - 南方报业

此专题所有 1,023 篇报道 >

叙利亚反政府军呼吁外国军队发动空袭 加速政府垮台 搜狐 - 5分钟前

专题报道(600篇) >

速冻食品新国标下月施行 金黄葡萄球菌允许检出 新民网 - 17分钟前

专题报道(2,050篇) >

大连四把火处理结果中石油董事长被处分 14人法办 凤凰网 - 43分钟前

专题报道(755篇) >

销售员坚持投注获双色球二等奖 搜狐 - 20分钟前

专题报道(689篇) >

中国海军西太训练引关注 日媒称系有意对日施压 环球网 - 1小时前

专题报道(449篇) >

住建部研究中心主任陈淮解读房地产 凤凰网 - 19分钟前

专题报道(3,029篇) >

人民日报谈四川藏区多起年轻僧人自焚事件(1) 中华网 - 10分钟前

成都一中学选拔19名“尖子生”与校长共进晚餐 网易 - 13分钟前

午评: 金融地产“变脸”拖累大盘反弹 沪指缩量跌0.36% 和讯网 - 39分钟前 - 专题报道(92篇) >

陈浩民就袭胸门痛哭道歉 阿娇陈冠希领衔玩道歉的明星(图) 新民网 - 20分钟前 - 专题报道(530篇) >

传Facebook手机最早明年4月上市 或免费 凤凰网 - 34分钟前 - 专题报道(84篇) >

阿里财报解读: 良无限价值远超一般交易平台 搜狐 - 36分钟前 - 专题报道(464篇) >

坏事也能变好事 韦迪: 未来中超要恢复亚冠满额 搜狐 - 29分钟前 - 专题报道(200篇) >

大四男生校内强奸同学未遂杀人 曾获学校奖学金 新浪网 - 28分钟前 - 专题报道(102篇) >

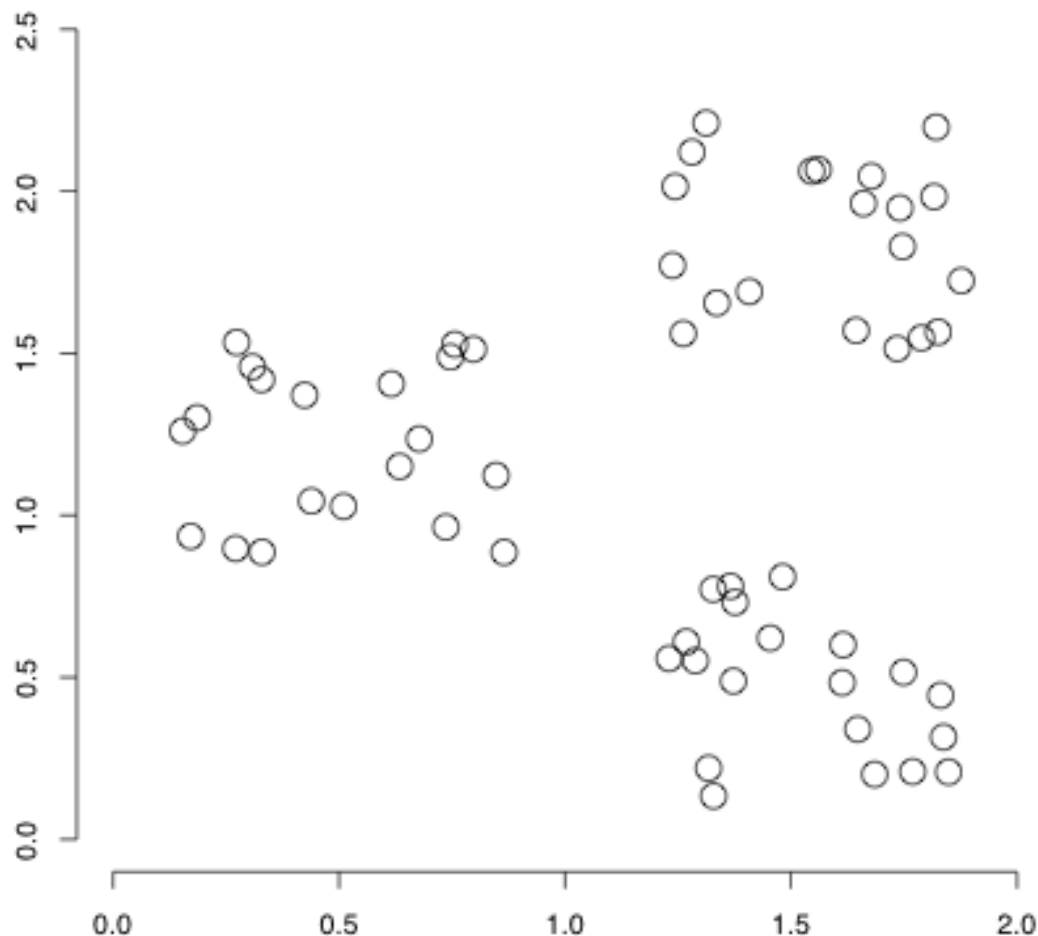
通用中国甘文维: 宝骏630月销量已达5000辆(图) 搜狐 - 46分钟前 - 专题报道(145篇) >

# 全局浏览的例子: Google News

The screenshot shows the Google News interface for the 'Science' category. The left sidebar contains navigation options like 'Top stories', 'For you', 'Following', 'Saved searches', 'COVID-19', 'U.S.', 'World', 'Your local news', 'Business', 'Technology', 'Entertainment', 'Sports', 'Science' (highlighted), and 'Health'. At the bottom of the sidebar are links for 'Language & region', 'Settings', and 'Get the Android app'. The main content area features a search bar at the top and a 'Science' header with 'Follow' and 'Share' buttons. Below the header are tabs for 'Latest', 'Environment', 'Outer space', 'Physics', 'Genetics', and 'Wildlife'. Three news articles are displayed, each with a 'View Full Coverage' link circled in red:

- New Climate Satellite Measures Sea Level Rise** (NPR • 32 minutes ago)
  - SpaceX will launch a new NASA satellite -- and land with a boom this weekend (CNET • 12 hours ago)
  - [View Full Coverage](#)
- SpaceX's plans for a reusable Dragon spacecraft fleet detailed by Gwynne Shotwell** (Teslarati • 5 hours ago)
  - NASA video relives this week's spectacular SpaceX mission to the ISS (Digital Trends • 9 hours ago)
  - [View Full Coverage](#)
- Star collision helps solve 16-year-old mystery of the Blue Ring Nebula** (CNET • 14 hours ago)
  - The dazzling Blue Ring Nebula puzzled scientists for 16 years -- and now they finally understand why (Space.com • 21 hours ago)
  - [View Full Coverage](#)

# 一个具有清晰簇结构的数据集



提出一个算法来  
寻找该例中的簇  
结构



# 聚类的要求

- 一般目标：将相关文档放到一个簇中，将不相关文档放到不同簇中
  - 如何对上述目标进行形式化？
- 簇的数目应该合适，以便与聚类的数据集相吻合
  - 一开始，我们假设给定簇的数目为 $K$ 。
  - 后面会介绍确定 $K$ 的半自动的方法
- 聚类的其它目标
  - 避免非常小和非常大的簇
  - 定义的簇对用户来说很容易理解
  - 其它.....



# 扁平聚类 vs. 层次聚类

- 扁平算法
  - 通过一开始将全部或部分文档随机划分为不同的组
  - 通过迭代方式不断修正
  - 代表算法：K-均值聚类算法
- 层次算法
  - 构建具有层次结构的簇
  - 自底向上(Bottom-up)的算法称为凝聚式(agglomerative)算法
  - 自顶向下(Top-down)的算法称为分裂式(divisive)算法

# 硬聚类 vs. 软聚类

- 硬聚类(Hard clustering): 每篇文档仅仅属于一个簇
  - 很普遍并且相对容易实现
- 软聚类(Soft clustering): 一篇文档可以属于多个簇
  - 对于诸如浏览目录之类的应用来说很有意义
  - 比如, 将 胶底运动鞋 (sneakers) 放到两个簇中:
    - 体育服装(sports apparel)
    - 鞋类(shoes)
  - 只有通过软聚类才能做到这一点
- 本节课关注扁平的硬聚类算法
- 有关软聚类和层次聚类参考《信息检索导论》第16.5节、第17章和第18章

# 扁平算法

- 扁平算法将 $N$ 篇文档划分成 $K$ 个簇
- 输入：给定一个文档集合及聚类结果簇的个数 $K$
- 输出：寻找一个划分将这个文档集合分成 $K$ 个簇，该结果满足某个最优划分准则
- 全局优化：穷举所有的划分结果，从中选择最优的那个划分结果
  - 无法处理
- 高效的启发式方法:  $K$ -均值聚类算法

# 提纲

- ① 上一讲回顾
- ② 聚类介绍
- ③ 聚类在IR中的应用
- ④ K-均值聚类算法
- ⑤ 聚类评价
- ⑥ 簇个数确定
- ⑦ 深度聚类

# K-均值聚类算法

- 或许是最著名的聚类算法
- 算法十分简单，但是在很多情况下效果不错
- 是文档聚类的默认或基准算法

# 聚类中的文档表示

- 向量空间模型
- 同基于向量空间的分类一样，这里我们也采用欧氏距离的方法来计算向量之间的相关性...
- ...欧氏距离与余弦相似度差不多等价(如果两个向量都基于长度归一化，那么欧氏距离和余弦相似度是等价的)
- 然而，质心向量通常都没有基于长度进行归一化

# K-均值聚类算法

- K-均值聚类算法中的每个簇都定义为其质心向量
- 质心向量的定义：

$$\vec{\mu}(\omega) = \frac{1}{|\omega|} \sum_{\vec{x} \in \omega} \vec{x}$$

其中  $\omega$  代表一个簇

- 划分准则：使得所有文档到其所在簇的质心向量的距离平方和最小
- 通过下列两步来实现目标优化：
  - **重分配(reassignment)**: 将每篇文档分配给离它最近的簇
  - **重计算(recomputation)**: 重新计算每个簇的质心向量

# K-均值聚类算法

$K$ -MEANS( $\{\vec{x}_1, \dots, \vec{x}_N\}, K$ )

```

1   $(\vec{s}_1, \vec{s}_2, \dots, \vec{s}_K) \leftarrow \text{SELECTRANDOMSEEDS}(\{\vec{x}_1, \dots, \vec{x}_N\}, K)$ 
2  for  $k \leftarrow 1$  to  $K$ 
3  do  $\vec{\mu}_k \leftarrow \vec{s}_k$ 
4  while stopping criterion has not been met
5  do for  $k \leftarrow 1$  to  $K$ 
6      do  $\omega_k \leftarrow \{\}$ 
7      for  $n \leftarrow 1$  to  $N$ 
8          do  $j \leftarrow \arg \min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$ 
9               $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$  (reassignment of vectors)
10         for  $k \leftarrow 1$  to  $K$ 
11             do  $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$  (recomputation of centroids)
12 return  $\{\vec{\mu}_1, \dots, \vec{\mu}_K\}$ 
    
```

重分配

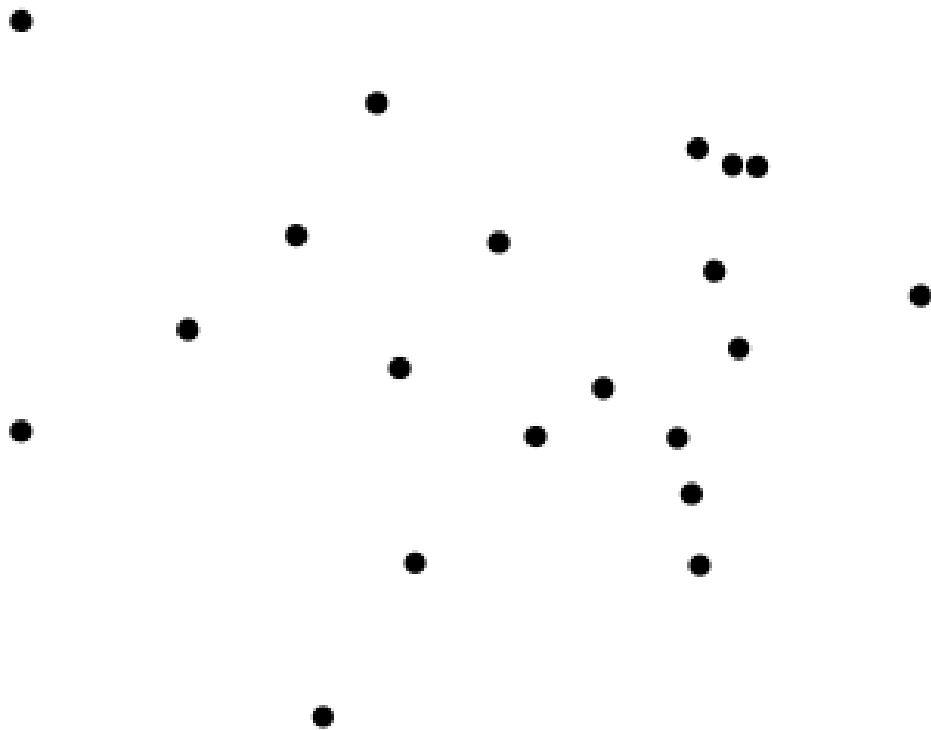
重计算



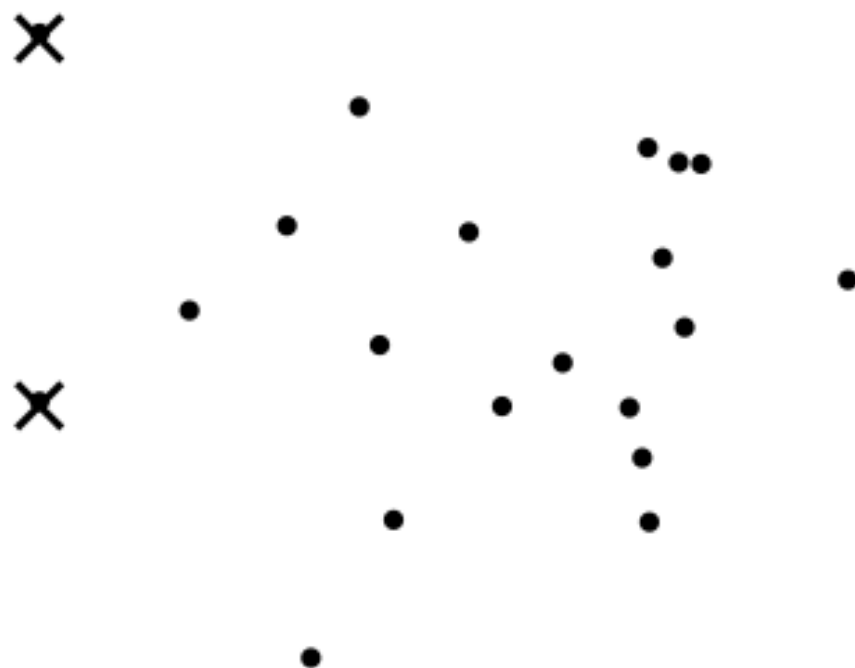
# K-均值算法的stopping criterion

- K-均值聚类算法可以采用如下终止条件。
- 当迭代一个**固定次数I**后停止。该条件能够限制聚类算法的运行时间，但有些情况下，由于迭代次数不足，聚类结果的质量并不高。
- 当文档到簇的分配结果**不再改变**后停止。除了某些情况下会使算法陷入局部最优外，该停止条件通常会产出较好的聚类结果，但是运行时间不宜太久。
- 当**质心向量**  $\vec{\mu}_k$  **不再改变**后停止。这**等价于**文档到簇的分配结果不再改变。
- 思考：为什么等价？

# 例子

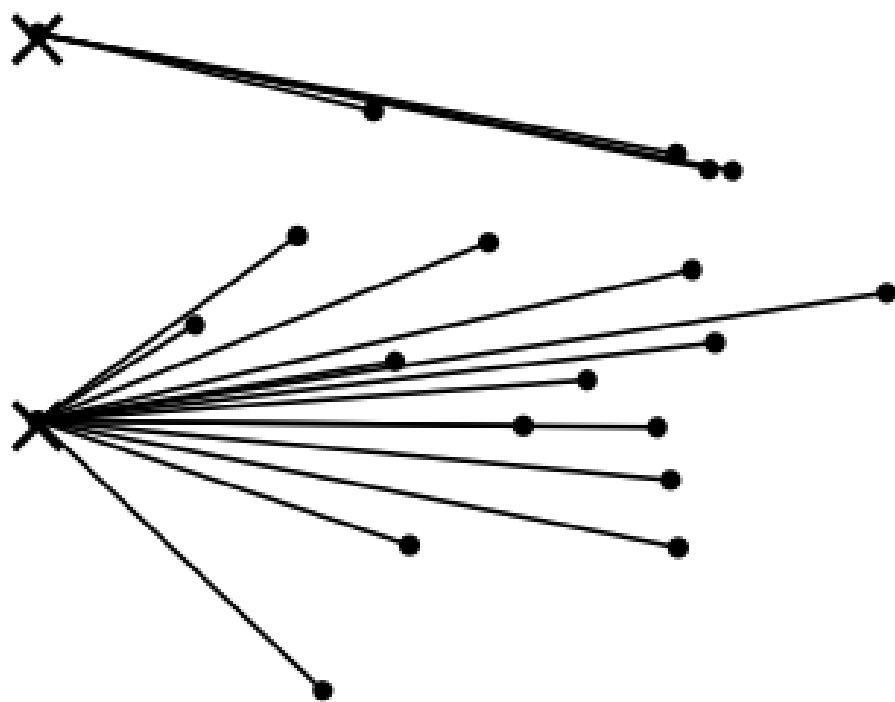


# 例子：随机选择两个种子( $K=2$ )



- (i) 猜猜最后划分的两个簇是什么？
- (ii) 计算簇的质心向量

例子：将文档分配给离它最近的质心向量(第一次)







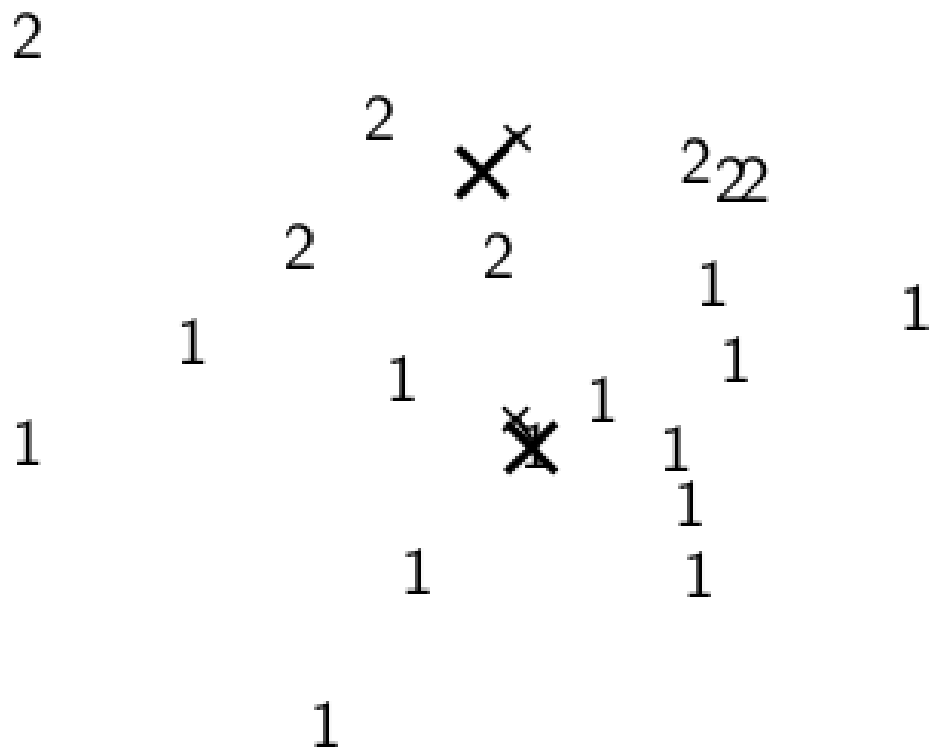
例子：将文档分配给离它最近的质心向量(第二次)



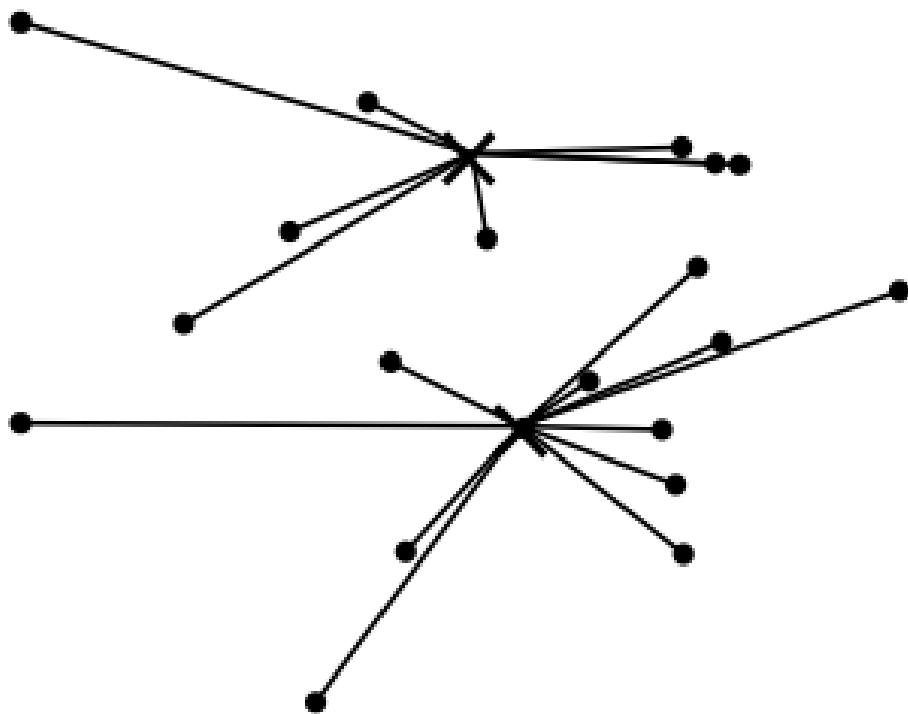




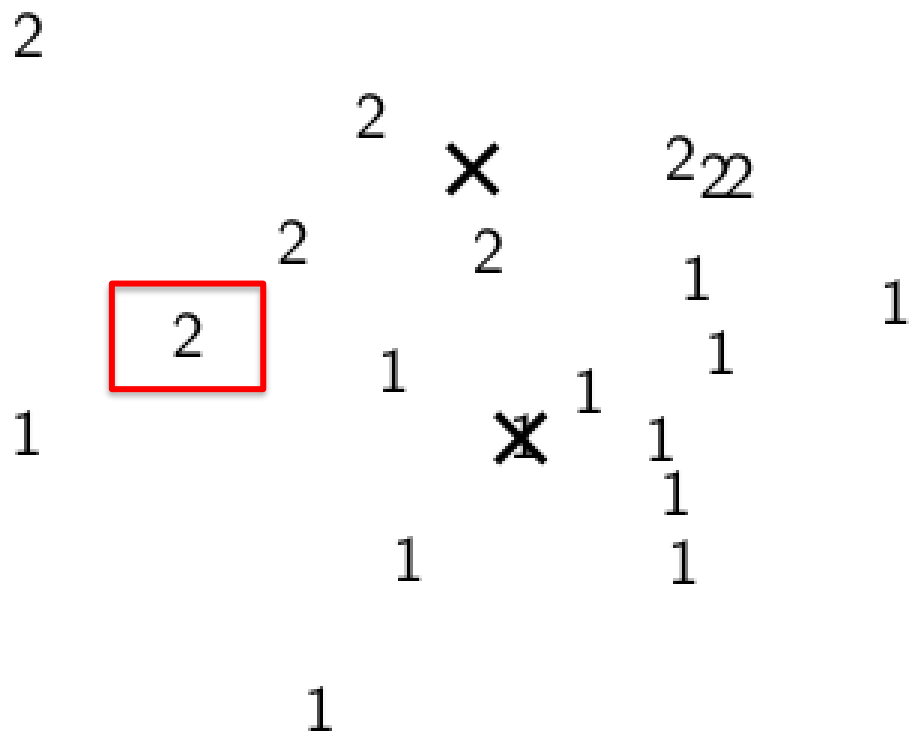
## 例子：重新计算质心向量



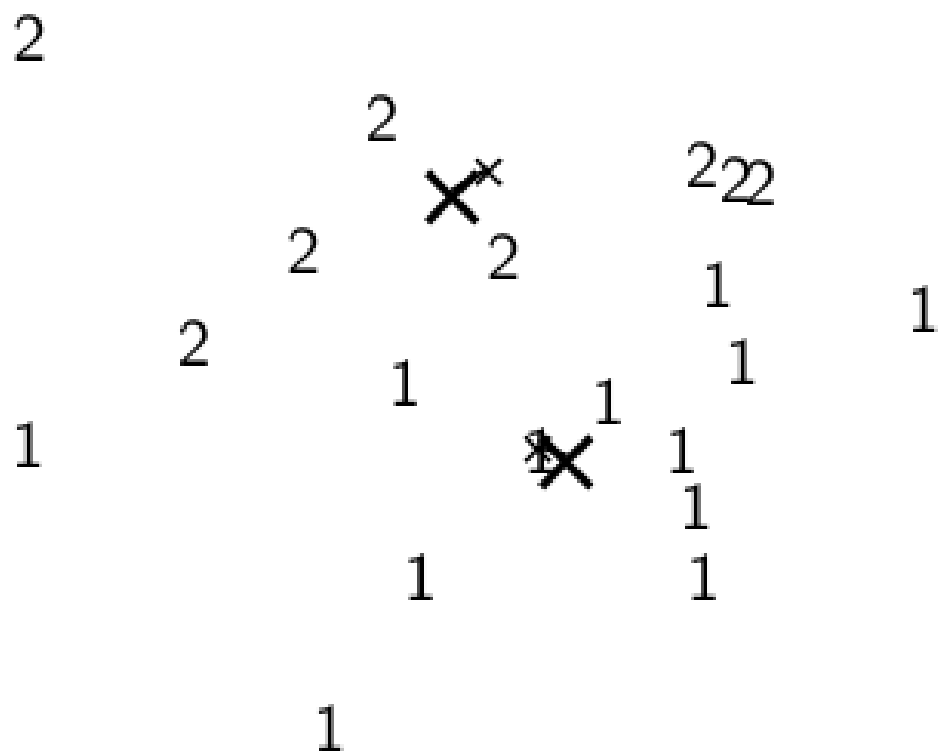
## 例子：再重新分配(第三次)



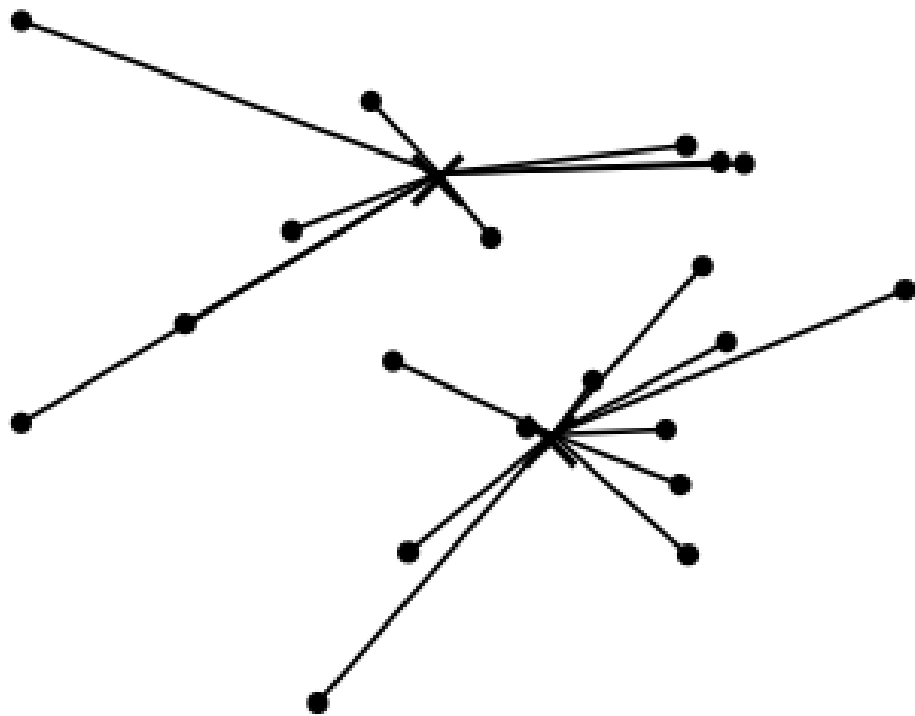
## 例子：分配结果



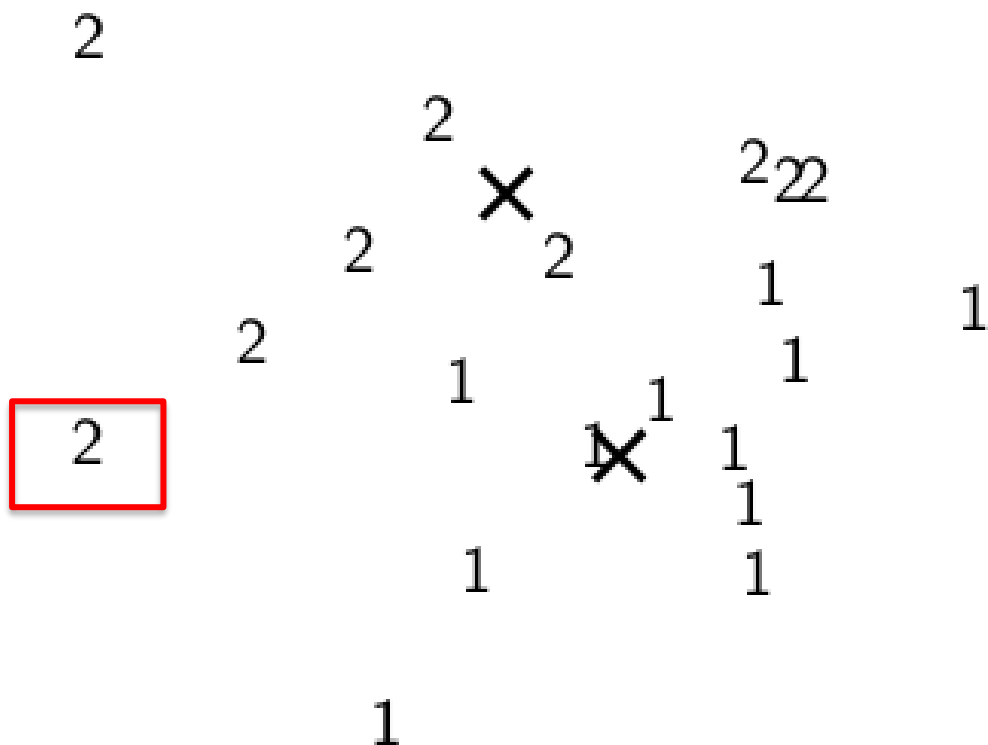
## 例子：重新计算质心向量



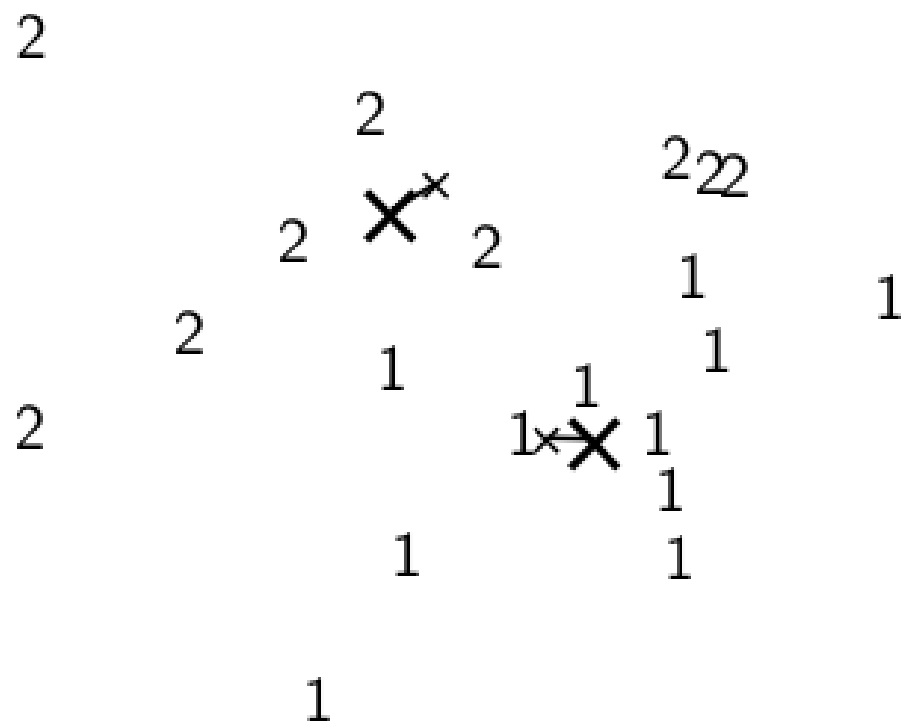
## 例子：再重新分配(第四次)



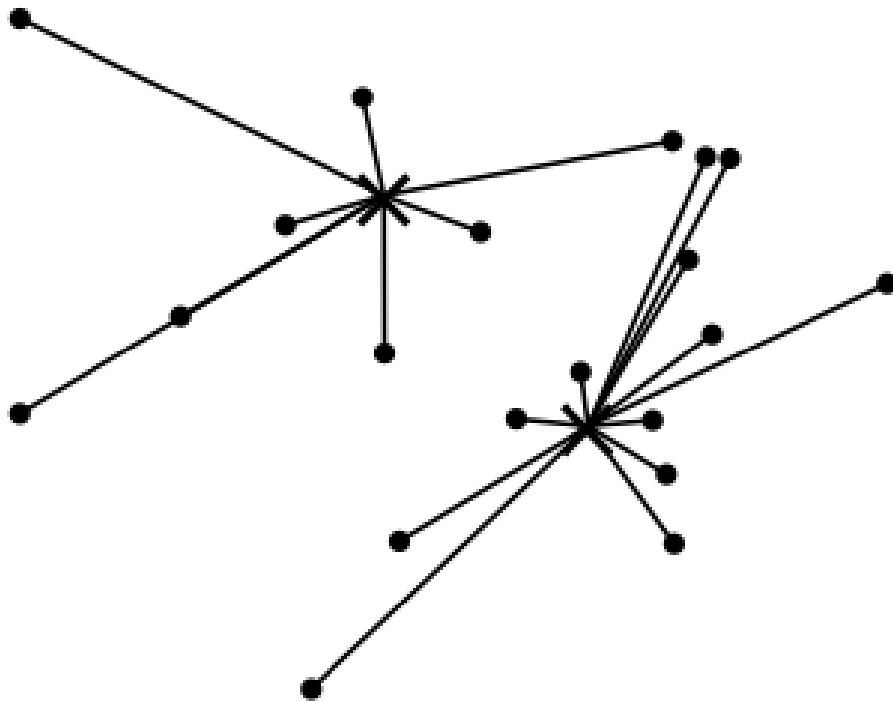
## 例子：分配结果



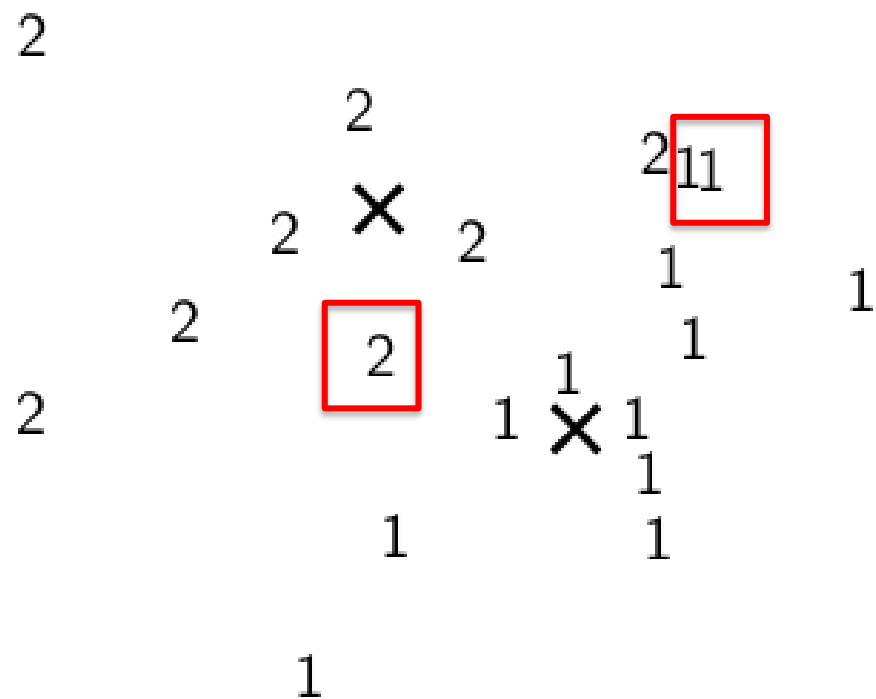
## 例子：重新计算质心向量



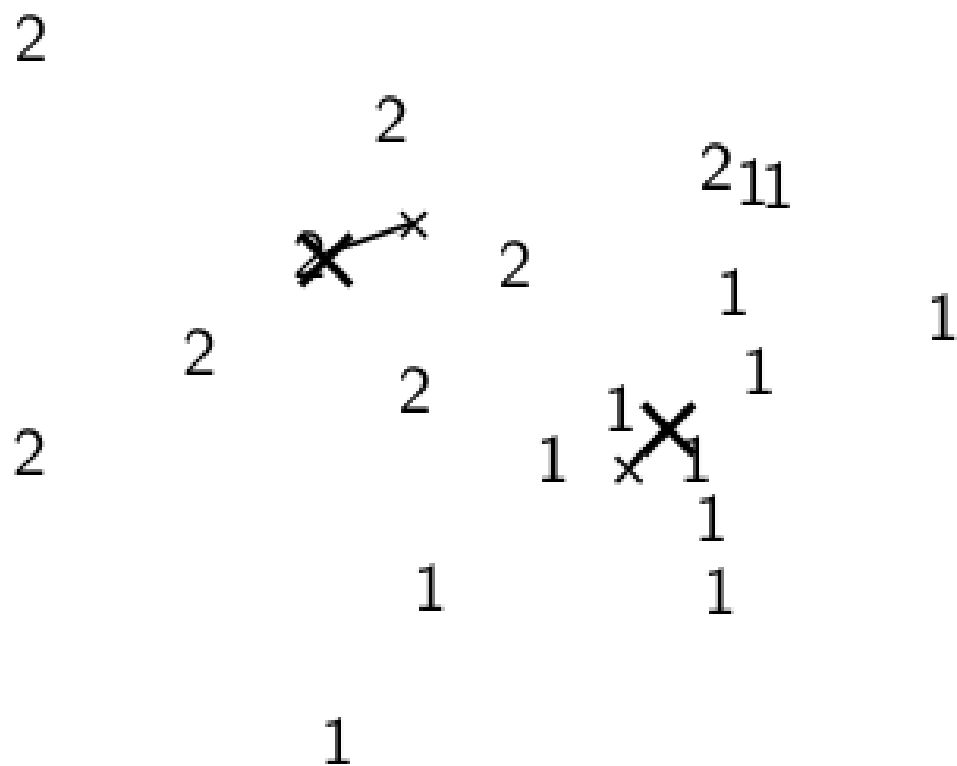
## 例子：重新分配(第五次)



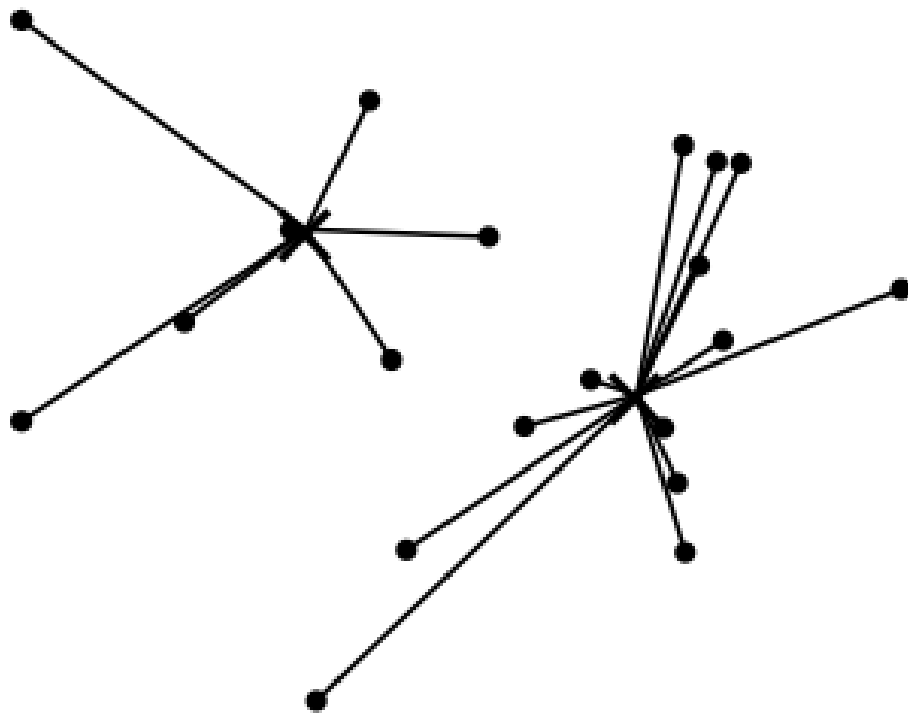




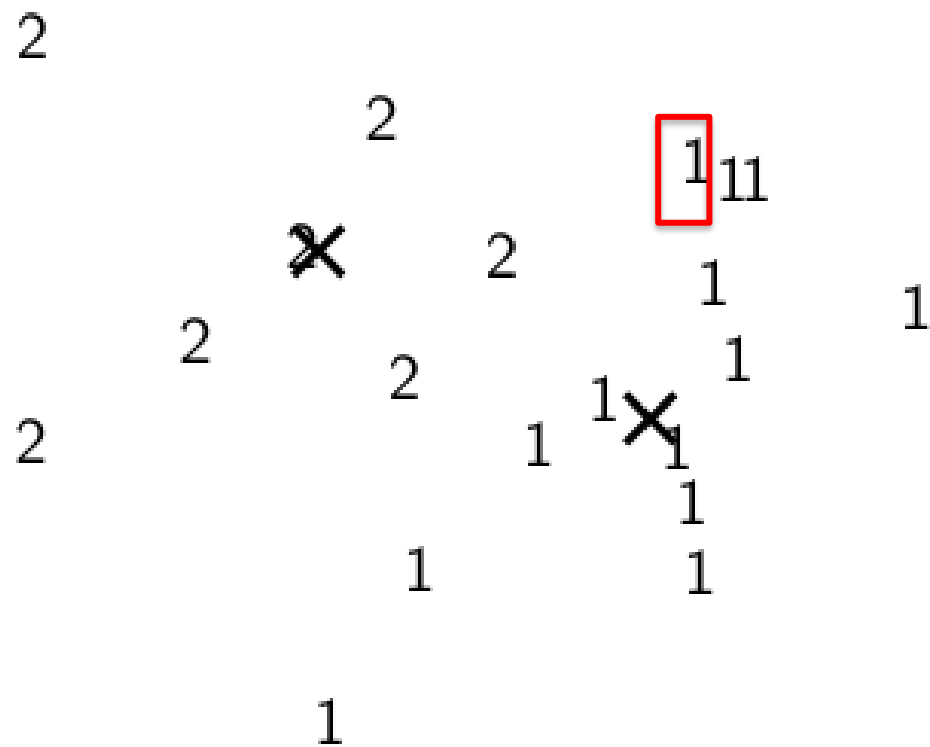
## 例子：重新计算质心向量



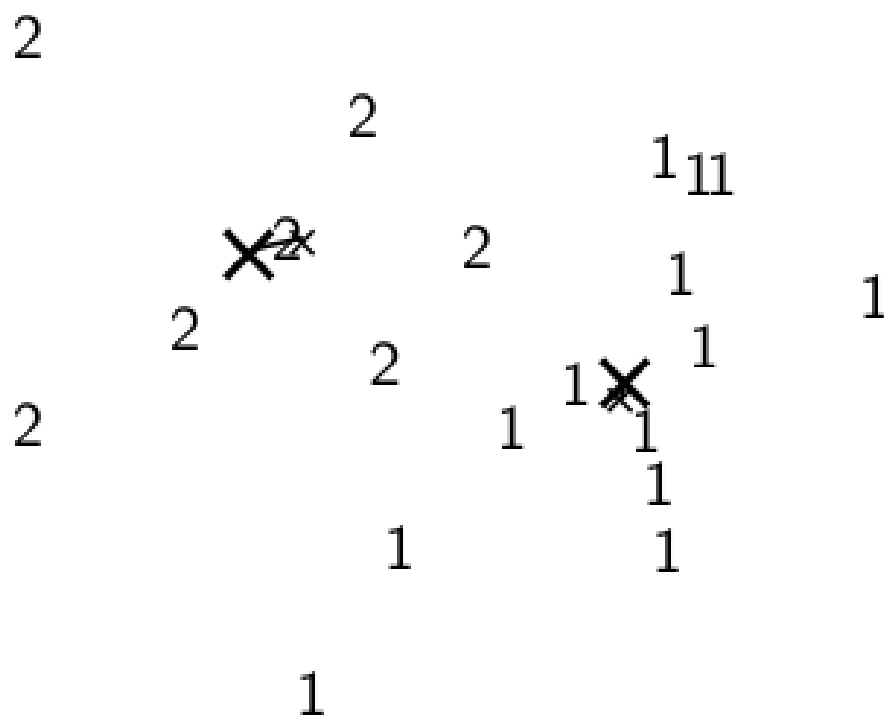
## 例子：重新分配(第六次)



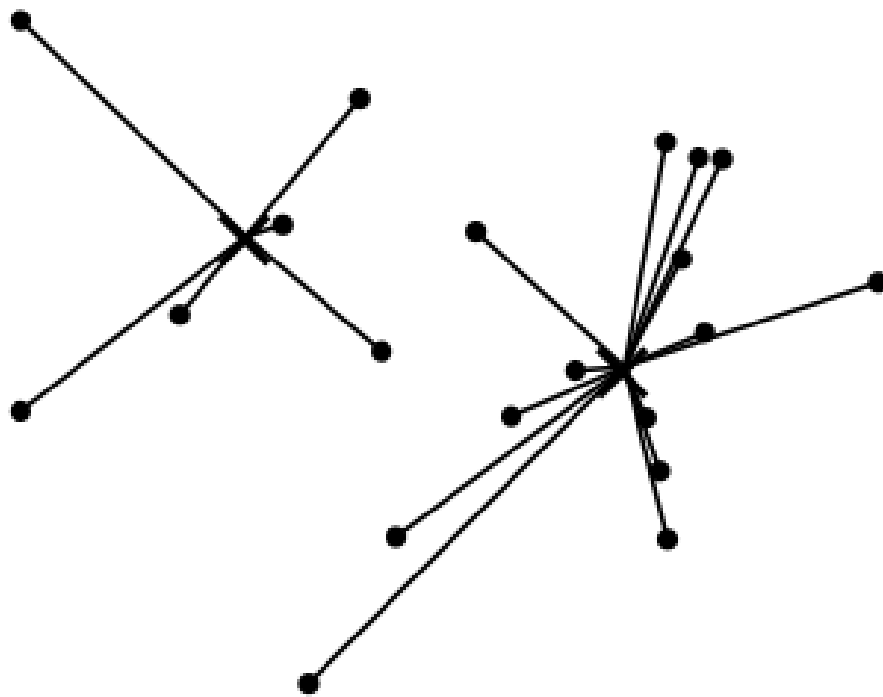
## 例子：分配结果

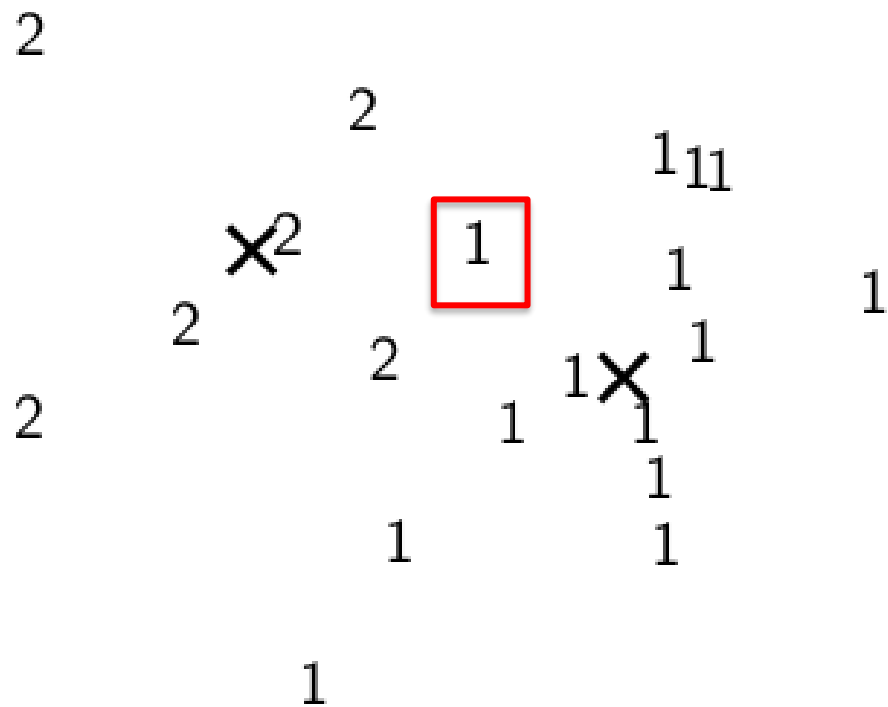


## 例子：重新计算质心向量

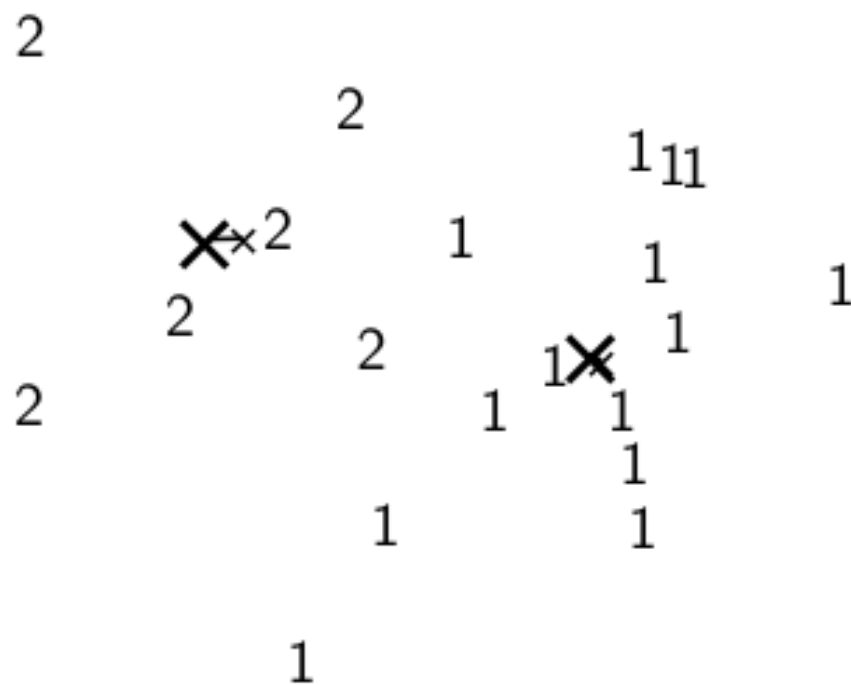


## 例子：重新分配(第七次)



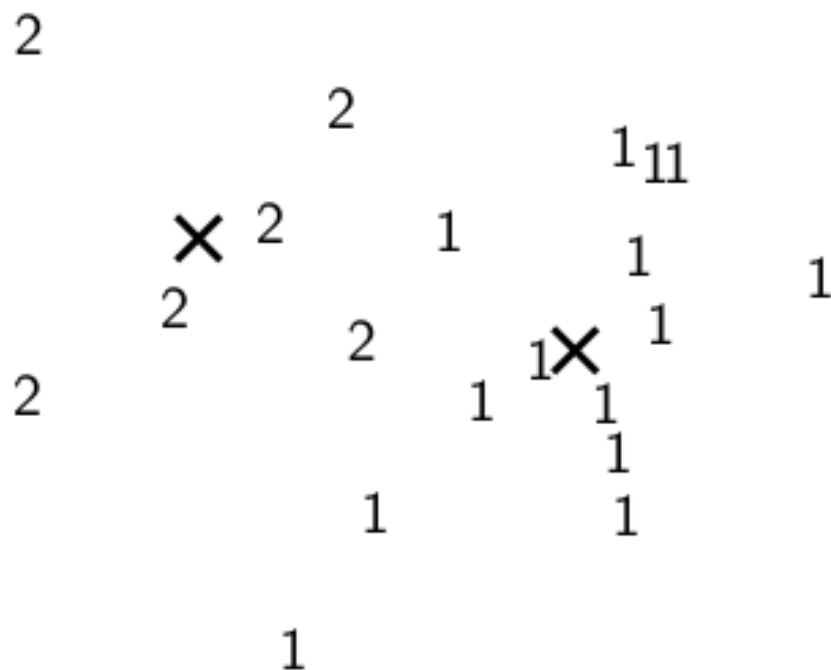


## 例子：重新计算质心向量





# 质心向量和分配结果最终收敛



# K-均值聚类算法的时间复杂度

$K$ -MEANS( $\{\vec{x}_1, \dots, \vec{x}_N\}, K$ )

```

1   $(\vec{s}_1, \vec{s}_2, \dots, \vec{s}_K) \leftarrow \text{SELECTRANDOMSEEDS}(\{\vec{x}_1, \dots, \vec{x}_N\}, K)$ 
2  for  $k \leftarrow 1$  to  $K$ 
3  do  $\vec{\mu}_k \leftarrow \vec{s}_k$ 
4  while stopping criterion has not been met
5  do for  $k \leftarrow 1$  to  $K$ 
6      do  $\omega_k \leftarrow \{\}$ 
7      for  $n \leftarrow 1$  to  $N$ 
8          do  $j \leftarrow \arg \min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$ 
9               $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$  (reassignment of vectors)
10     for  $k \leftarrow 1$  to  $K$ 
11         do  $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$  (recomputation of centroids)
12 return  $\{\vec{\mu}_1, \dots, \vec{\mu}_K\}$ 
    
```

$O(KNM)$

$O(NM)$

# K-均值聚类算法的时间复杂度

- 计算两个向量的距离的时间复杂度为  $O(M)$ .
- 重分配过程:  $O(KNM)$  (需要计算  $KN$  个文档-质心的距离)
- 重计算过程:  $O(NM)$  (在计算质心向量时, 需要累加簇内的文档向量)
- 假定迭代次数的上界是  $I$
- 整体复杂度:  $O(IKNM)$ ~线性
- 但是, 上述分析并没有考虑到实际中的最坏情况
- 在一些非正常的情况下, 复杂度可能会比线性更糟

# K-均值聚类中的目标函数

- 一个衡量质心对簇中文档的代表程度的指标是RSS (Residual Sum of Squares, 残差平方和)
- $RSS =$  所有簇上的文档向量到(最近的)质心向量的距离平方和的总和

$$RSS = \sum_{k=1}^K RSS_k \quad RSS_k = \sum_{x \in \omega_k} |\vec{x} - \vec{\mu}(\omega_k)|^2$$

- RSS是K-均值算法的目标函数，我们的目的是要让这个函数取最小值
- 由于N是固定的，最小化RSS就等价于最小化平方距离，而平方距离度量的正是质心对文档的代表能力

# K-均值聚类算法一定会收敛: 证明

- 每次重新分配之后RSS会下降
  - 这是因为每个向量都被移到离它最近的质心向量所代表的簇中
- 每次重新计算之后RSS也会下降
  - 参见下一页幻灯片
- 可能的聚类结果是有穷的
- 因此：一定会收敛到一个固定点
- 当然，这里有一个假设就是假定出现了等值的情况，算法都采用前后一致的方法来处理(比如，某个向量到两个质心向量的距离相等)

# 重新计算之后RSS也会下降的证明

$$\text{RSS} = \sum_{k=1}^K \text{RSS}_k$$

$$\text{RSS}_k(\vec{v}) = \sum_{\vec{x} \in \omega_k} \|\vec{v} - \vec{x}\|^2 = \sum_{\vec{x} \in \omega_k} \sum_{m=1}^M (v_m - x_m)^2$$

$$\frac{\partial \text{RSS}_k(\vec{v})}{\partial v_m} = \sum_{\vec{x} \in \omega_k} 2(v_m - x_m) = 0$$

$$v_m = \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} x_m$$

这正好是基于每个向量分量来计算的质心的定义。因此，当将旧质心替换为新质心时，我们让 $\text{RSS}_k$ 极小化。重新计算之后，作为 $\text{RSS}_k$ 之和的RSS一定也会下降。

# $K$ -均值聚类算法一定是收敛的

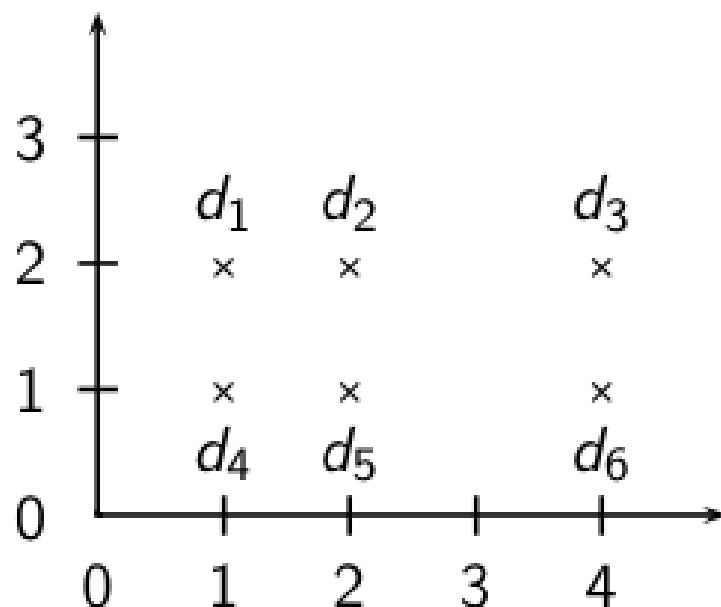
- 但是不知道达到收敛所需要的时间!
- 如果不太关心少许文档在不同簇之间来回交叉的话, 收敛速度通常会很快 ( $< 10$ - $20$ 次迭代)
- 但是, 完全的收敛需要多得多的迭代过程

# K-均值聚类算法的最优性

- 收敛并不意味着会达到全局最优的聚类结果!
- 这是K-均值聚类算法的最大缺点之一
- 如果开始的种子选的不好，那么最终的聚类结果可能会非常糟糕



# 有关收敛性的课堂练习：次优的聚类结果



对于种子 $d_2$ 和 $d_5$ ，K-均值算法最后收敛为 $\{\{d_1, d_2, d_3\}, \{d_4, d_5, d_6\}\}$   
 对种子 $d_2$ 和 $d_3$ ，收敛结果为 $\{\{d_1, d_2, d_4, d_5\}, \{d_3, d_6\}\}$ ，这是 $K=2$ 时的全局最优值

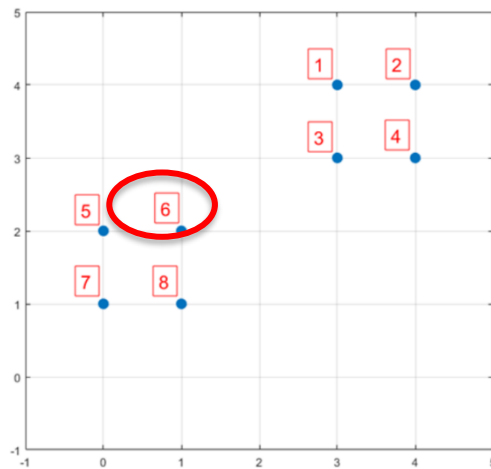
- $K=2$ 情况下的最优聚类结果是什么？
- 对于任意的种子 $d_i$ 、 $d_j$ ，我们是否都会收敛于该聚类结果？
- $d_2$ 和 $d_3$ 为种子， $d_2$ 和 $d_5$ 为种子，聚类结果分别是怎样的？

# K-均值聚类算法的初始化

- 种子的随机选择只是K-均值聚类算法的一种初始化方法之一
- 随机选择不太鲁棒：可能会获得一个次优的聚类结果
- 一些确定初始质心向量的更好办法：
  - 非随机地采用某些启发式方法来选择种子(比如，过滤掉一些离群点，或者寻找具有较好文档空间覆盖度的种子集合)
  - 采用层级聚类算法寻找好的种子
  - 选择  $i$  (比如  $i = 10$ ) 次不同的随机种子集合，对每次产生的随机种子集合运行K-均值聚类算法，最后选择具有最小RSS值的聚类结果
  - 为每个簇选出 $i$ 个随机向量，将它们的质心向量作为该簇的种子向量

# K-means++： K-means初始化方法的改进

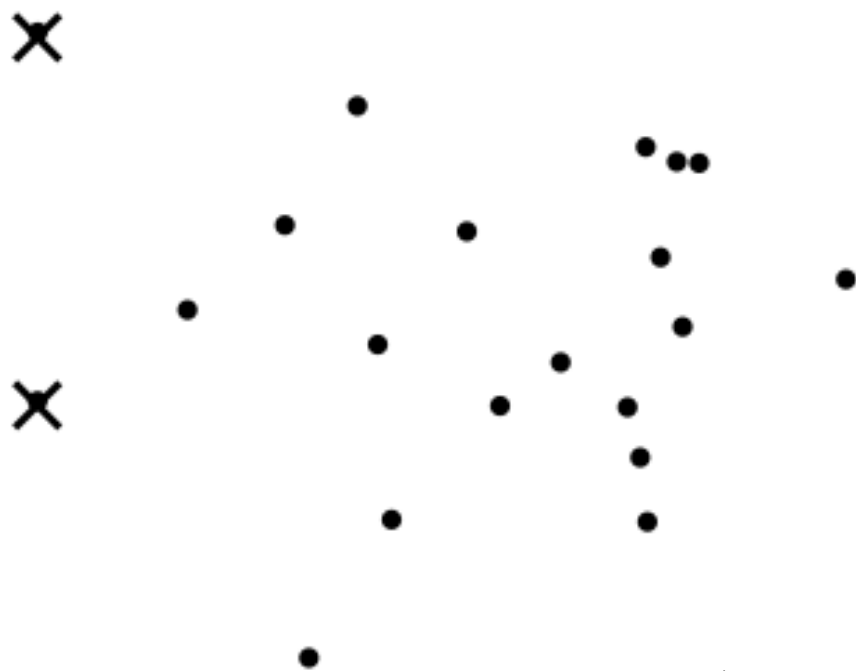
- 种子选择原则：聚类中心尽可能离的远
- 步骤一：随机选取一个样本作为第一个聚类中心  $c_1$ ；
- 步骤二：计算每个样本与当前已有聚类中心最短距离（即与最近一个聚类中心的距离），用  $D(x)$ 表示；接着计算每个样本被选为聚类中心的概率  $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$  最后，用轮盘法选出下一个聚类中心；
- 步骤三：重复步骤二，直至选出  $k$  个聚类中心。



序号	①	②	③	④	⑤	⑥	⑦	⑧
$D(x)$	$2\sqrt{2}$	$\sqrt{13}$	$\sqrt{5}$	$\sqrt{10}$	1	0	$\sqrt{2}$	1
$D(x)^2$	8	13	5	10	1	0	2	1
$P(x)$	0.2	0.325	0.125	0.25	0.025	0	0.05	0.025
Sum	0.2	0.525	0.65	0.9	0.925	0.925	0.975	1

# K-中心点算法 (K-medoids)

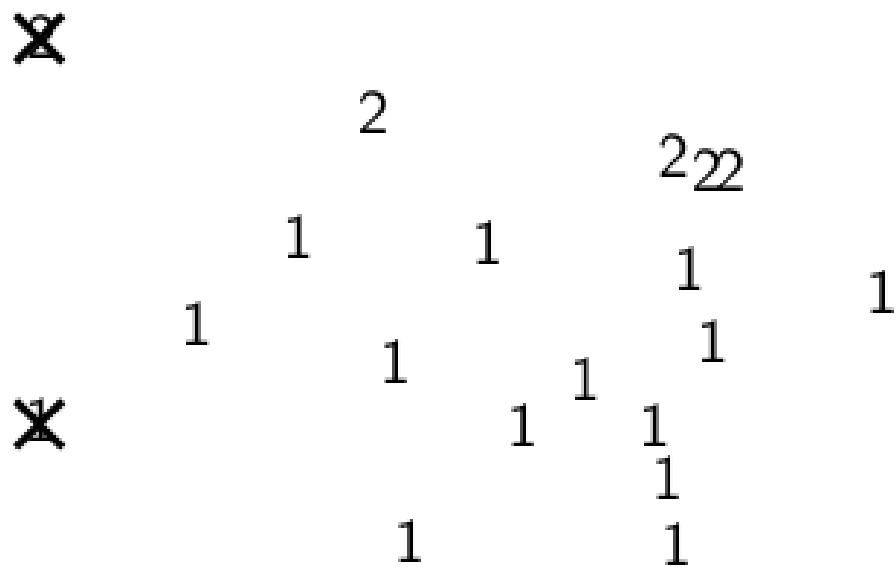
- 重计算阶段不同于K-均值算法
- 从当前 cluster 中**选取**这样一个**点**作为质心，它到其他所有（当前 cluster 中的）点的距离之和最小



例子：随机选择两个种子( $K=2$ )

# K-中心点算法 (K-medoids)

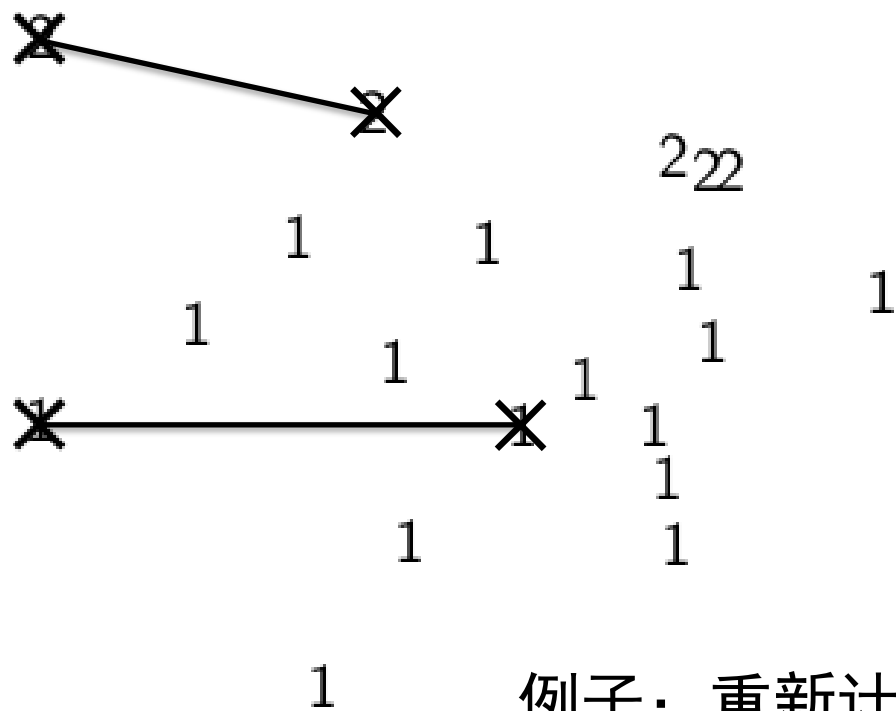
- 重计算阶段不同于K-均值算法
- 从当前 cluster 中**选取**这样一个**点**作为质心，它到其他所有（当前 cluster 中的）点的距离之和最小



1 例子：分配后的簇(第一次)

## K-中心点算法 (K-medoids)

- 重计算阶段不同于K-均值算法
- 从当前 cluster 中选取这样一个点作为质心，它到其他所有（当前 cluster 中的）点的距离之和最小



- 对噪声鲁棒性好
- 时间复杂度高，计算质心的时间复杂度为 $O(N^2M)$

### 例子：重新计算质心向量

# 提纲

- ① 上一讲回顾
- ② 聚类介绍
- ③ 聚类在IR中的应用
- ④ K-均值聚类算法
- ⑤ 聚类评价
- ⑥ 簇个数确定
- ⑦ 深度聚类

# 怎样判断聚类结果的好坏？

- 内部准则(Internal criteria)
  - 一个内部准则的例子： K-均值聚类算法的RSS值
- 但是内部准则往往不能评价聚类在应用中的实际效用
- 替代方法： 外部准则(External criteria)
  - 按照用户定义的分类结果来评价，即对一个分好类的数据集进行聚类，将聚类结果和事先的类别情况进行比照，得到最后的评价结果



# 外部准则

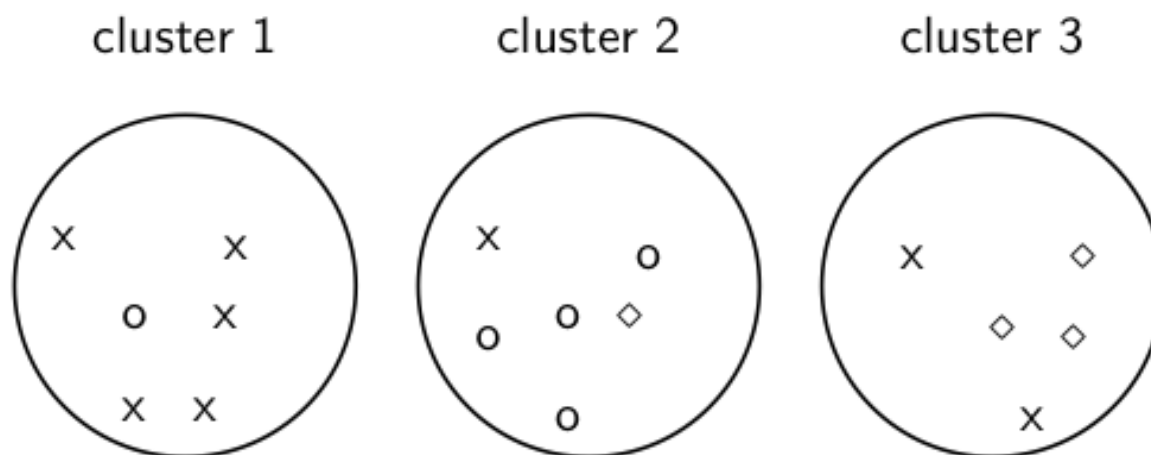
- 基于已有标注的标准数据集(如Reuters语料库)来进行聚类评价
- 目标：聚类结果和给定分类结果一致
- (当然，聚类中我们并不知道最后每个簇的标签，而只是关注如何将文档分到不同的组中)
- 一个评价指标：纯度([purity](#))

# 外部准则: 纯度

$$\text{purity}(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

- $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$  是簇的集合
- $C = \{c_1, c_2, \dots, c_J\}$  是类别的集合
- 对每个簇  $\omega_k$ : 找到一个类别  $c_j$ , 该类别包含  $\omega_k$  中的元素最多, 为  $n_{kj}$  个, 也就是说  $\omega_k$  的元素最多分布在  $c_j$  中
- 将所有  $n_{kj}$  求和, 然后除以所有的文档数目

# 纯度计算的例子



为计算纯度

$$\max_j |\omega_1 \cap c_j| = 5 \quad (\text{class } x, \text{ cluster } 1);$$

$$\max_j |\omega_2 \cap c_j| = 4 \quad (\text{class } o, \text{ cluster } 2);$$

$$\max_j |\omega_3 \cap c_j| = 3 \quad (\text{class } \diamond, \text{ cluster } 3)$$

$$\text{纯度为 } (1/17) \times (5 + 4 + 3) \approx 0.71.$$

# 兰迪指数(Rand index)

- 定义:  $RI = \frac{TP+TN}{TP+FP+FN+TN}$

- 考虑所有两个文档之间(文档对)的关系, 可以得到 2x2 的列联表:

	same cluster	different clusters
same class	true positives (TP)	false negatives (FN)
different classes	false positives (FP)	true negatives (TN)

- 总的文档对数目为  $TP+FN+FP+TN$
- 对于N篇文档, 总共有  $\binom{N}{2}$  个文档对
- 例子: 上例中,  $\binom{17}{2} = 136$
- 每个文档对要么为positive或negative (聚类算法要么将这两个文档放在同一簇中, 要么放在不同簇中)...
- ... 聚类结果要么 “true” (correct) 要么 “false” (incorrect): 即聚类的结果要么正确要么不正确

## 兰迪指数：例子

回到上例，三个簇中分别包含6、6、5个点，因此处于同一簇的文档对的个数为：

$$TP + FP = \binom{6}{2} + \binom{6}{2} + \binom{5}{2} = 40$$

其中，簇1中的x对，簇2中的o对，簇3中的◇对，以及簇3中的x对，都是真正例：

$$TP = \binom{5}{2} + \binom{4}{2} + \binom{3}{2} + \binom{2}{2} = 20$$

于是,  $FP = 40 - 20 = 20$ 。类似地，可以计算出FN和TN。

# 兰迪指数

$$RI = \frac{TP+TN}{TP+FP+FN+TN}$$

	same cluster	different clusters	
same class	TP = 20	FN = 24	RI is then
different classes	FP = 20	TN = 72	

$$(20 + 72)/(20 + 20 + 24 + 72) \approx 0.68.$$

# 另外两种聚类评价指标

- 归一化互信息(Normalized mutual information, NMI)
  - 聚类结果包含多少分类信息？
  - 单点簇 (簇个数 = 文档个数) 具有最大的MI
  - 因此，需要给予簇和类的熵进行归一化
- F 值
  - 类似兰迪指数，但是正确率和召回率可以加权平均。

# 另外两种聚类评价指标

## 归一化互信息

$$NMI(\Omega, C) = \frac{I(\Omega, C)}{[H(\Omega) + H(C)] / 2}$$

$$\begin{aligned} I(\Omega, C) &= \sum_k \sum_j P(\omega_k \cap c_j) \log \frac{P(\omega_k \cap c_j)}{P(\omega_k)P(c_j)} \\ &= \sum_k \sum_j \frac{|\omega_k \cap c_j|}{N} \log \frac{N |\omega_k \cap c_j|}{|\omega_k| |c_j|} \circ \end{aligned}$$

$$\begin{aligned} H(\Omega) &= -\sum_k P(\omega_k) \log P(\omega_k) \\ &= -\sum_k \frac{|\omega_k|}{N} \log \frac{|\omega_k|}{N} \circ \end{aligned}$$

一种能在聚类质量和簇数目之间维持均衡的指标



# 另外两种聚类评价指标

## F值

- 可以使用8.3 节讨论的F 值来度量聚类结果，并通过设置 $\beta > 1$  以加大对FN 的惩罚，此时实际上也相当于赋予召回率更大的权重。

$$P = \frac{TP}{TP+FP} \quad R = \frac{TP}{TP+FN} \quad F_{\beta} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

# 聚类评价结果比较

	purity	NMI	RI	$F_5$
lower bound	0.0	0.0	0.0	0.0
maximum	1.0	1.0	1.0	1.0
value for example	0.71	0.36	0.68	0.46

所有指标都从0 (非常差的聚类结果) 到 1 (完美聚类)

# 提纲

- ① 上一讲回顾
- ② 聚类介绍
- ③ 聚类在IR中的应用
- ④ K-均值聚类算法
- ⑤ 聚类评价
- ⑥ 簇个数确定
- ⑦ 深度聚类

# 簇个数确定

- 在很多应用中，簇个数  $K$  是事先给定的
  - 比如，可能存在对  $K$  的外部限制
  - 例子：在“分散-集中”应用中，在显示器上(上世纪90年代)很难显示超过10-20个簇
- 如果没有外部的限制会怎样？是否存在正确的簇个数？
- 一种办法：定义一个优化准则
  - 给定文档，找到达到最优情况的  $K$  值
  - 能够使用的最优准则有哪些？
  - 我们不能使用前面所提到的RSS或到质心的平均平方距离等准则，因为它们会导致  $K = N$  个簇

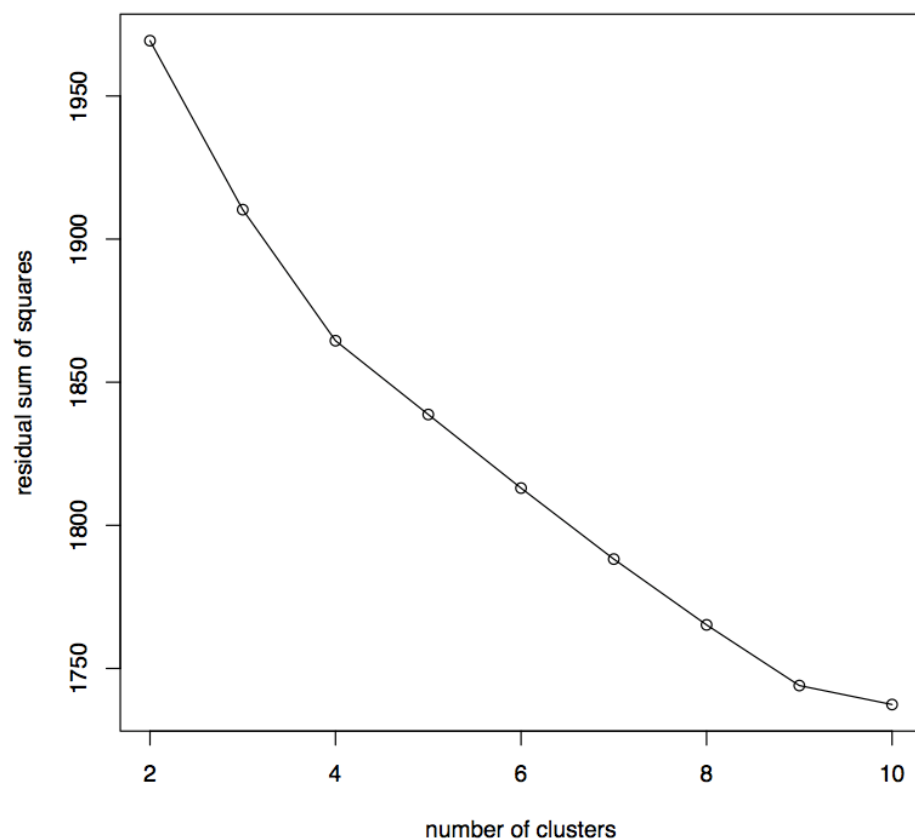
# 简单的目标函数 (1)

- 基本思路:
  - 从1个簇开始 ( $K = 1$ )
  - 不断增加簇 (= 不断增大  $K$ )
  - 对每个新的簇增加一个惩罚项
- 在惩罚项和RSS之间折中
- 选择满足最佳折中条件的  $K$

## 简单的目标函数 (2)

- 给定聚类结果，定义文档的代价为其到质心向量的(平方)距离（失真率）
- 定义全部失真率  $RSS(K)$  为所有文档代价的和
- 然后：对每个簇一个惩罚项  $\lambda$
- 于是，对于具有  $K$  个簇的聚类结果，总的聚类惩罚项为  $K\lambda$
- 定义聚类结果的所有开销为失真率和总聚类惩罚项的和：
- $RSS(K) + K\lambda$
- 选择使得  $(RSS(K) + K\lambda)$  最小的  $K$  值
- 当然，还要考虑较好的  $\lambda$  值 ...

# 在曲线中寻找拐点



拐点的结果具有一定的代表性

本图中两个拐点：4 和 9

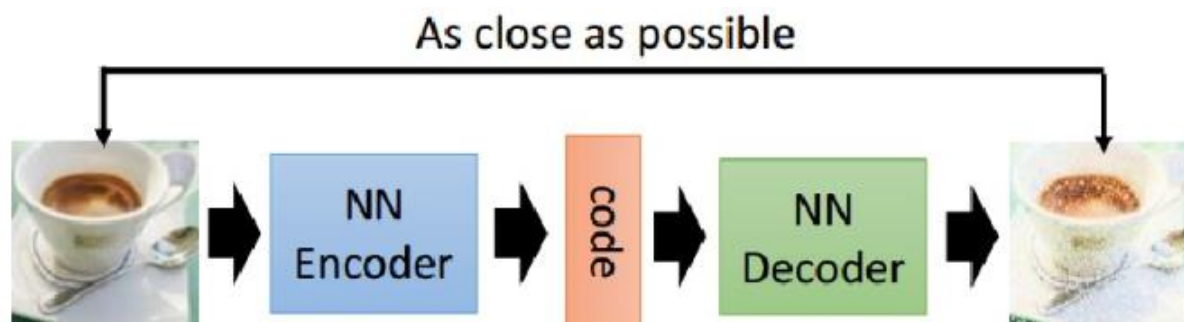
# 提纲

- ① 上一讲回顾
- ② 聚类介绍
- ③ 聚类在IR中的应用
- ④ K-均值聚类算法
- ⑤ 聚类评价
- ⑥ 簇个数确定
- ⑦ 深度聚类**



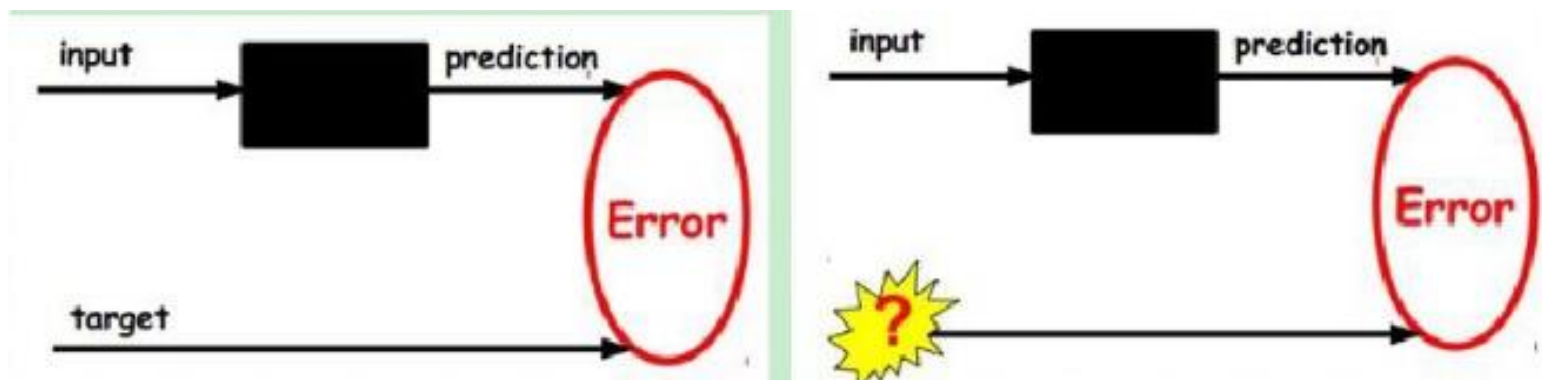
# 自编码器 (AutoEncoder, AE)

- 自编码器是人工神经网络的一种类型，使用**无监督**的方式学习高效的数据值编码。
- 自编码器的目的是学习一组数据的表示（编码），通常用于降维
- 自编码器是一种尽可能复现输入信号的神经网络



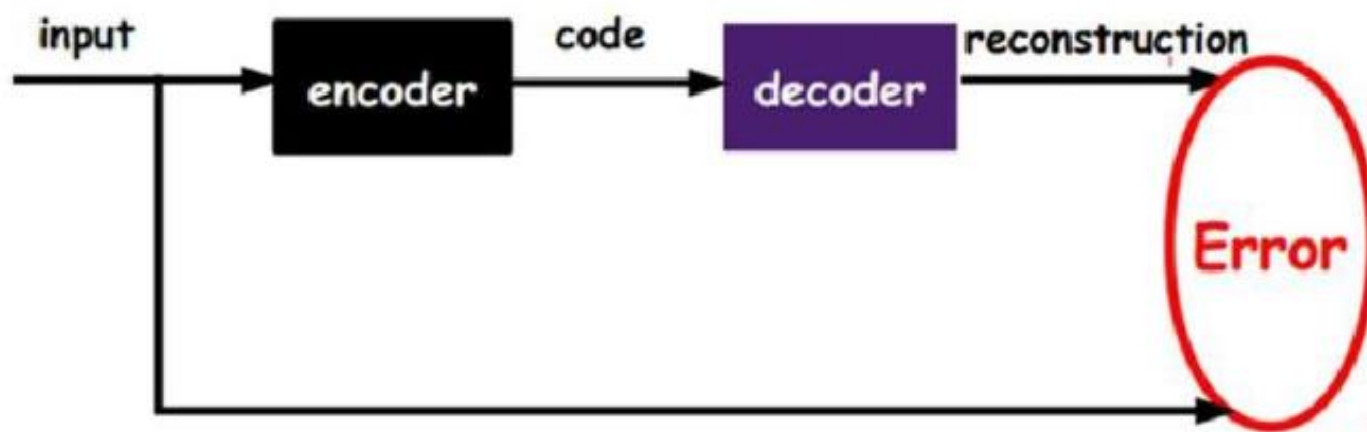
# 自编码器 (AutoEncoder, AE)

- 给定无标签数据，用无监督学习方式去学习特征



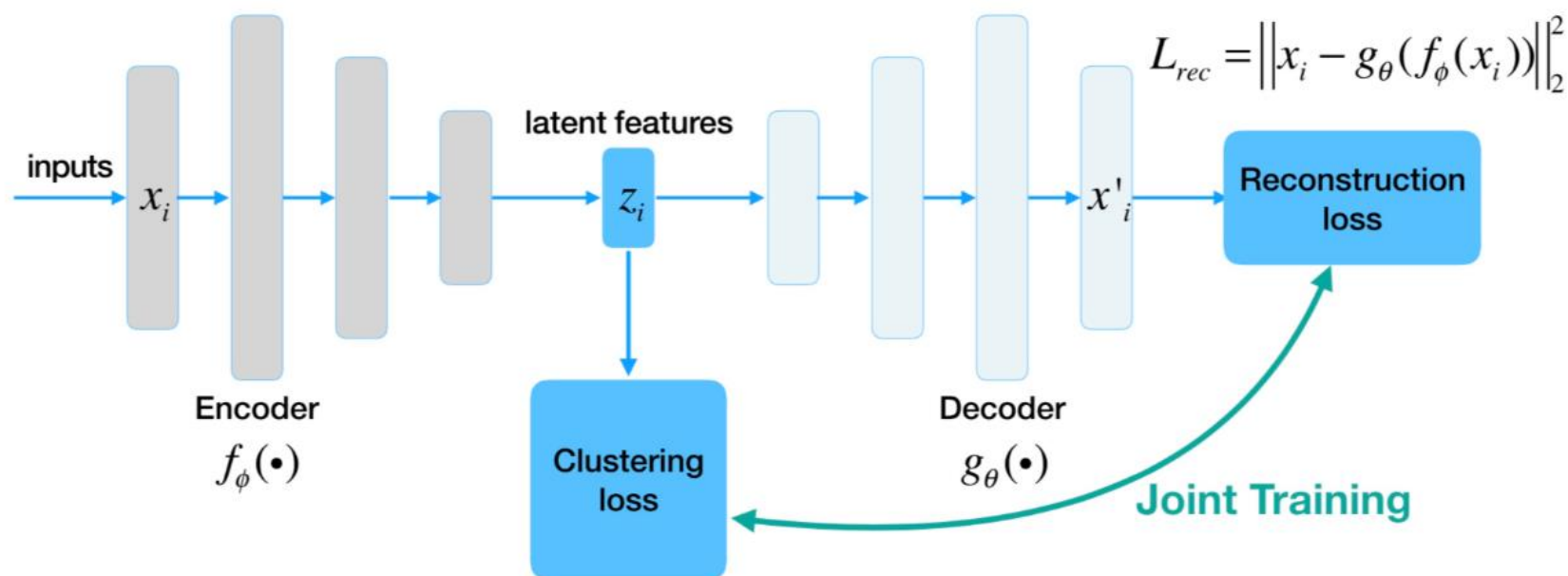
- 思考：对于无标签数据，调整参数的依据是什么？

# 自编码器 (AutoEncoder, AE)



优化目标是**重构误差**最小，来调整encoder和decoder的参数

# 基于AE的深度聚类



- 代表性算法：
  - Deep Clustering Network (DCN), ICML 2017;
  - Deep Embedding Network (DEN), ICPR 2014;
  - Deep Subspace Clustering Networks (DSC-Nets), NIPS 2017;
  - Deep Multi-Manifold Clustering (DMC), AAAI 2017;
  - Deep Embedded Regularized Clustering (DEPICT), ICCV 2017;
  - Deep Continuous Clustering (DCC), arxiv 2018
- $$\min_{W, Z, M, \{s_i\}} \sum_{i=1}^N \left( \ell(g(f(x_i)), x_i) + \frac{\lambda}{2} \|f(x_i) - Ms_i\|_2^2 \right)$$

# DeepCluster



This ECCV 2018 paper, provided here by the Computer Vision Foundation, is the author-created version.

The content of this paper is identical to the content of the officially published ECCV 2018

LNCS version of the paper as available on SpringerLink: <https://link.springer.com/conference/eccv>

## Deep Clustering for Unsupervised Learning of Visual Features

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze

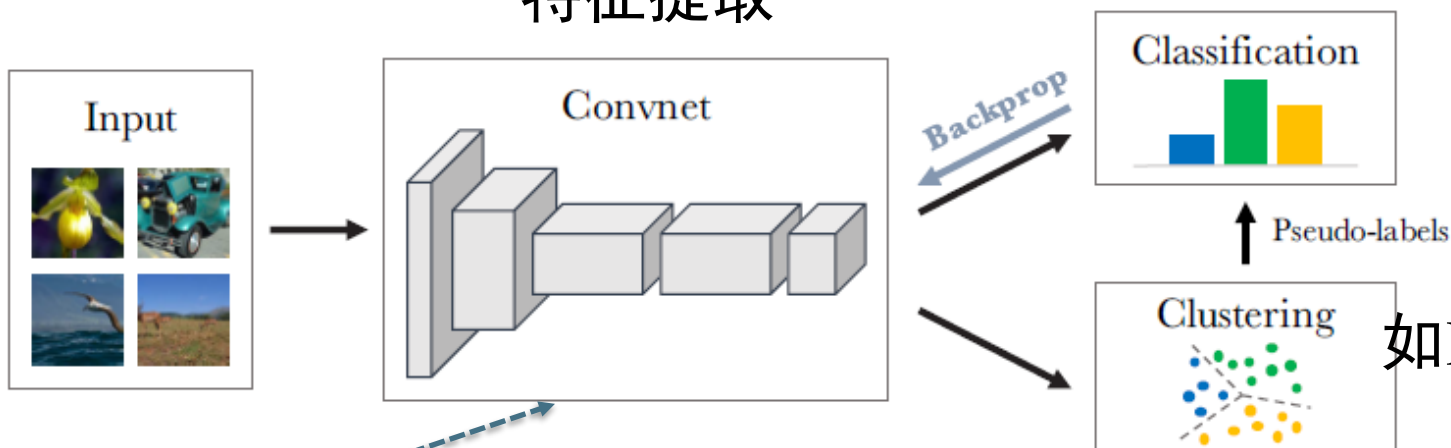
Facebook AI Research

`{mathilde,bojanowski,ajoulin,matthijs}@fb.com`

**Abstract.** Clustering is a class of unsupervised learning methods that has been extensively applied and studied in computer vision. Little work has been done to adapt it to the end-to-end training of visual features on large-scale datasets. In this work, we present DeepCluster, a clustering method that jointly learns the parameters of a neural network and the cluster assignments of the resulting features. DeepCluster iteratively groups the features with a standard clustering algorithm,  $k$ -means, and uses the subsequent assignments as supervision to update

# DeepCluster

## 特征提取



如K-means

网络损失

$$\min_{\theta, W} \frac{1}{N} \sum_{n=1}^N \ell(gw(f_{\theta}(x_n)), y_n)$$

聚类损失

$$\min_{C \in \mathbb{R}^{d \times k}} \frac{1}{N} \sum_{n=1}^N \min_{y_n \in \{0,1\}^k} \|f_{\theta}(x_n) - Cy_n\|_2^2$$

# 本讲小结

- 聚类的概念(What is clustering?)
- 聚类在IR中的应用
- $K$ -均值( $K$ -Means)聚类算法
- 聚类评价
- 簇(cluster)个数(即聚类的结果类别个数)确定

# 参考资料

- 《信息检索导论》 第16章
- <http://ifnlp.org/ir>
  - $K$ -均值聚类算法的例子
  - Keith van Rijsbergen有关聚类假设的论述
  - Bing/Carrot2/Clusty: 搜索结果聚类
- <https://www.cnblogs.com/yixuan-xu/p/6272208.html>
- Clustering for Unsupervised Learning of Visual Features, ECCV 2018
- Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering, ICML 2017



# 课后练习

---

- 习题16-4
- 在K-均值算法中，为什么对同一概念car使用不同词项来表示的文档最后可能会被归入到同一个簇中？
- 习题16-5
- K-均值算法的两个停止条件为：(i) 文档的分配不再改变；(ii)簇质心不再改变。请问这两个条件是否等价？