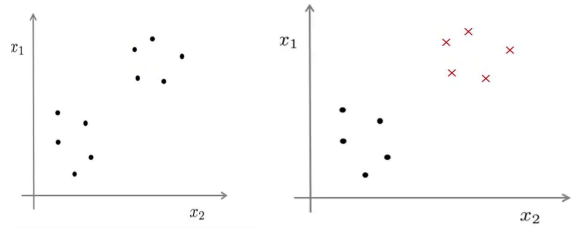
无监督学习
（Unsupervised learning-）

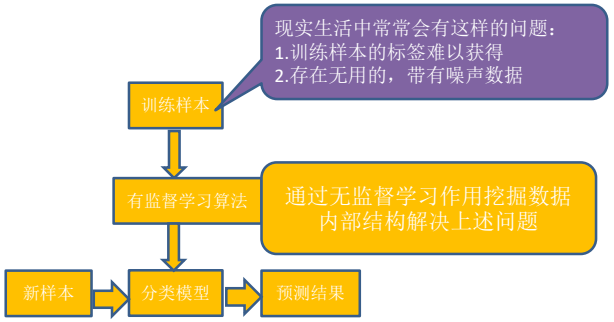
高琰

无监督学习



只给计算机训练数据，不给结果（标签），因此计算机无法准确地知道哪些数据具有哪些标签，只能凭借强大的计算能力分析数据的特征，发现数据本身的内部结构特点。

为什么要无监督学习？



无监督学习的应用背景

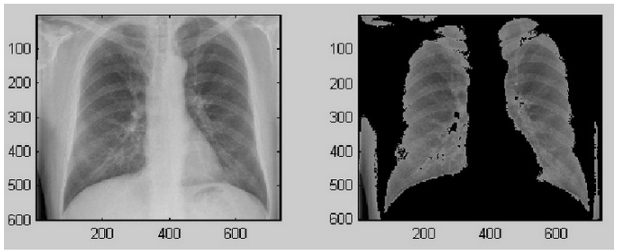
聚类

- 常见的应用背景包括：
  - 从庞大的样本集合中选出一些具有代表性的加以标注用于分类器的训练。
  - 先将所有样本自动分为不同的类别，再由人类对这些类别进行标注。
  - 在无类别信息情况下，寻找好的特征。

特征分析

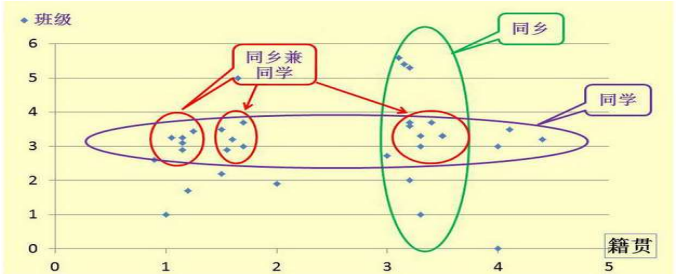
无监督学习的例子

- 尘肺分期自动判读中的肺野分割

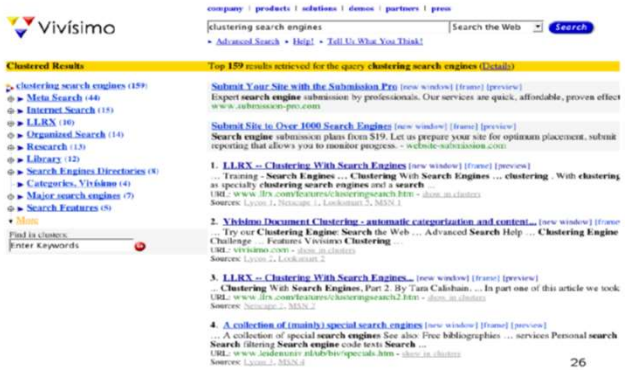


无监督学习的例子

- 客户分割（segmentation）是一种发现用户特性的方法。基于数据内部结构的分割将自然客户分组,从而给你一个客户信息的概况，这可以直接转化为增加客户的经营策略。



无监督学习的例子



无监督学习

- 下面例子，哪个使用的是无监督算法：
  - 给定邮件的标签（垃圾邮件或者是非垃圾邮件），学习一个垃圾邮件过滤器
  - 给定在网络上的一组新闻，将他们划分成相同的故事的新闻集合
  - 给定一组用户数据，自动地发现市场分割并且将顾客根据市场分割进行划分
  - 给定一组病人（标记了中风或是不中风）集合，学习预测新病人是否中风？

数据类型

- 二元变量（Binary variables）
- 区间标度变量（Interval-scaled variables ）
- 标称型，序数型和比例型变量（Nominal, ordinal and ratio variables ）
- 混合类型变量（Variables of mixed types）

数据类型

二元变量

	不浮出水面是否能生存	是否有脚蹼	属于鱼类
1	是	是	是
2	是	是	是
3	是	否	否
4	否	是	否
5	否	是	否

数据类型

标称型与区间标度变量

标称型

区间标度

Vehicle	Top speed km/h	Color	Air resistance	Weight Kg
V1	220	red	0.30	130
V2	230	black	0.32	
V3	260	red	0.29	
V4	140	gray	0.35	
V5	155	blue	0.33	950
V6	130	white	0.40	600
V7	100	black	0.50	3000
V8	105	red	0.60	2500
V9	110	gray	0.55	3500

数据类型

- 序数型变量

  其可能的值之间具有有意义的序或秩评定，但是相继值之间的差是未知的。序数属性通常用于等级评定调查。如：比赛名次

- 比率型变量（ratio variables）

  具有固定零点的数值属性，即一个值是另一个的倍数（比率）。比率值也是有序的，如：速度，重量等

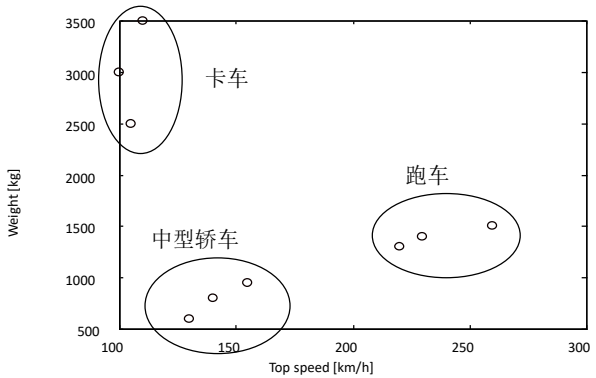
聚类（Clustering）

- 聚类定义：
  - 在一堆的数据中寻找一种”自然分组”(k组)。聚类中的组叫做簇（Cluster）希望同簇的样本较为相似，而不同簇的样本间有明显不同。

Vehicle Example

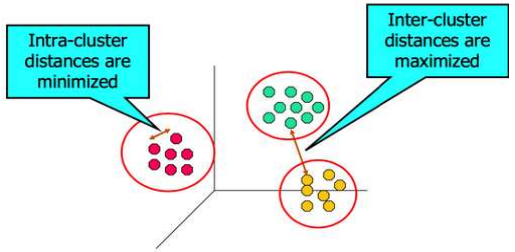
Vehicle	Top speed km/h	Colour	Air resistance	Weight Kg
V1	220	red	0.30	1300
V2	230	black	0.32	1400
V3	260	red	0.29	1500
V4	140	gray	0.35	800
V5	155	blue	0.33	950
V6	130	white	0.40	600
V7	100	black	0.50	3000
V8	105	red	0.60	2500
V9	110	gray	0.55	3500

Vehicle Clusters



什么是一个好的聚类方法?

- 一个好的聚类要要具备以下两个特点：
  - 高的簇内相似性（High intra-cluster similarity）
  - 低的簇间相似性 （Low inner-cluster similarity）



相似性(similarity)

相似性通常很难去定义，只能凭主观确定

差异性表示

- 与**相似性**相对应的就是**差异性**(dissimilarity或者说distance)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

数据矩阵　　差异矩阵

区间标度变量差异度计算

- 数据标准化
  - 计算绝对偏差的平均值:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|)$$

  其中 $m_f = \frac{1}{n}(x_{1f} + x_{2f} + \ldots + x_{nf})$.

  - 计算标准度量值 (*z-score*) $z_{if} = \frac{x_{if} - m_f}{s_f}$
- 计算距离

距离计算公式

- 常用的距离度量方法有:

  明考斯基距离（*Minkowski distance*）:

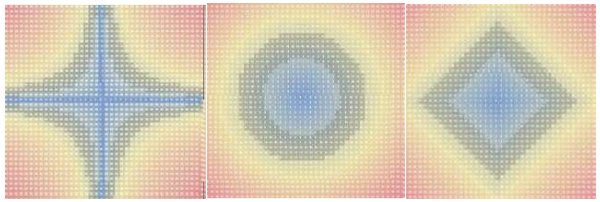
$$d(i,j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \ldots + |x_{ip} - x_{jp}|^q)}$$

  其中 $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ 和 $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ 是两个p维的数据对象, q是一个正整数。

  当q = 1时, *d* 称为**曼哈坦距离**（Manhattan distance）

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{ip} - x_{jp}|$$

各种距离的图示



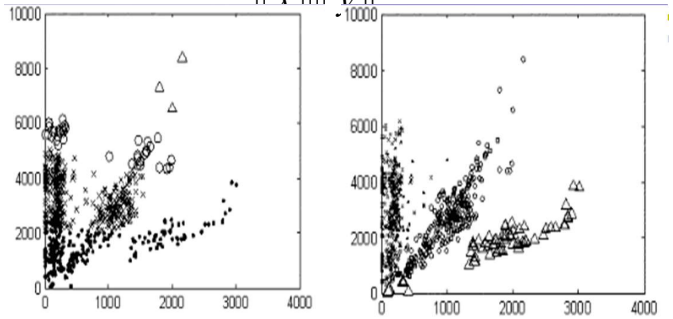
明考斯基距离
λ 在 0-1之间

欧几里得距离

曼哈坦距离

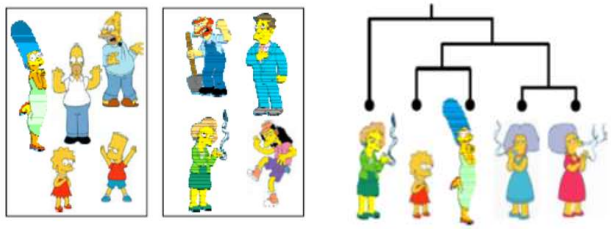
区间标度变量余弦相似度

$$\mathrm{sim}(x,y)=\frac{xy}{\|x\|\|y\|}$$



聚类算法分类1

- 划分聚类(Partitional Clustering)
- 层次聚类(Hierarchical Clustering)



聚类算法分类2

- 互斥聚类(Exclusive clustering)每个对象都指派到单个簇。
- 重叠聚类(Overlapping clustering)用来反映一个对象同时属于多个簇(类)这一事实。
- 模糊聚类(Fuzzy clustering)每个对象以一个0(绝对不属于)和1(绝对属于)之间的隶属权值属于每个簇。

聚类分类3

- 完全聚类(Complete clustering)将每一个对象都分配到某一个簇(cluster)
- 部分聚类(Partial clustering)有些对象没有被聚类，比如一些噪声(noise)或者离群值(outliers)

聚类（Clustering）

- 在一堆的数据中寻找一种”自然分组”(k组)。聚类中的组叫做簇（Cluster）希望同簇的样本较为相似，而不同簇的样本间有明显不同。

k-均值(k-means)算法

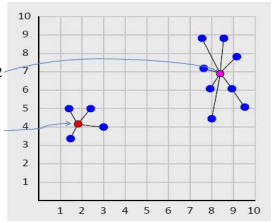
- 1.算法描述
- 2. 划分的准则函数
- 3. 算法的优缺点
- 4. 算法的扩展变形
- 5. 算法的应用

1.划分的准则函数

- **误差平方和：**

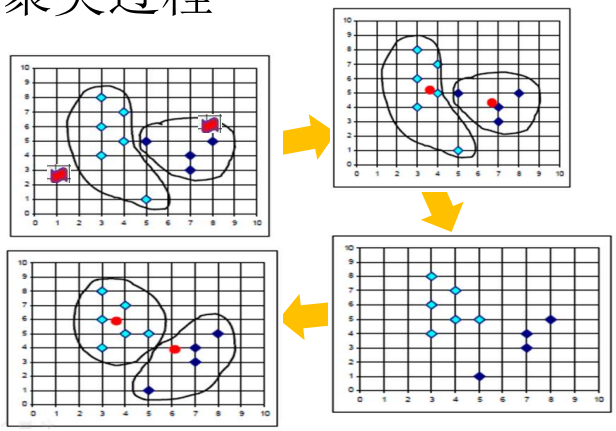
$$J_e = \sum_{i=1}^{k} \sum_{x \in C_i} |x - m_i|^2$$

$$m_i = \frac{1}{n_i} \sum_{x \in C_i} x$$



K-均值算法是获得准则函数Je最小的划分

2、聚类过程



2.算法描述

- **给定k，算法的处理流程如下:**

  第一步：随机的把所有对象分配到k个非空的簇中；

  第二步：计算每个簇的平均值，并用该平均值代表相应的簇中心；

  第三步：将每个对象根据其与各个簇中心的距离，重新分配到与它距离最近的簇中；

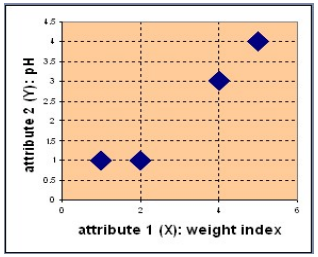
  第四步：重复2, 3直到k个簇的中心点不再发生变化或准则函数J_e收敛。

  O (kn)

- **时间复杂度:** O(tkn)

Example :

Object	Feature 1 (X): weight index	Feature 2 (Y): pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

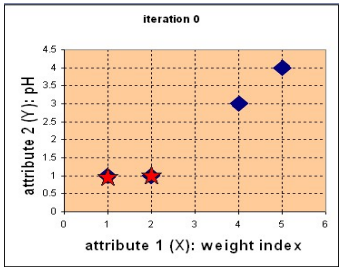


Example

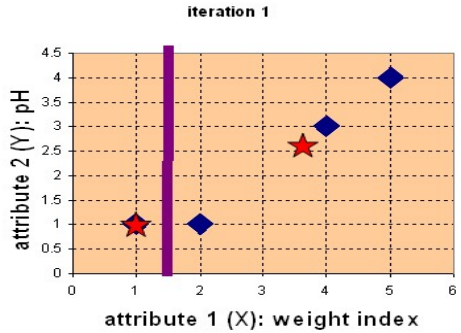
A B C D

$$\begin{pmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{pmatrix}$$

- $m_1=(1,1)$
- $m_2=(2,1)$

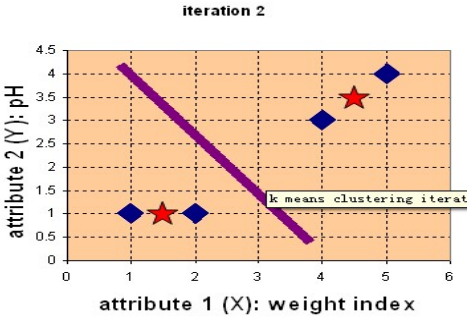


迭代1



迭代2

- m_1=(1,1) m_2=(11/3,8/3)
-



思考？

- 第3次迭代结果？
- 第3次迭代后是否还要继续迭代?

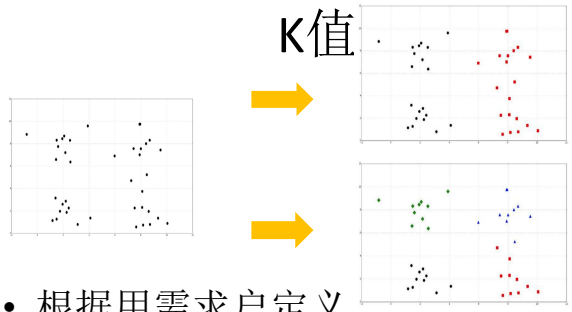
思考？

- A(2,5),B(2,10),C(8,4),D(5,8),E(4,9)
- 初始中心为A,E

第一次迭代后的中心？

最后的簇划分？

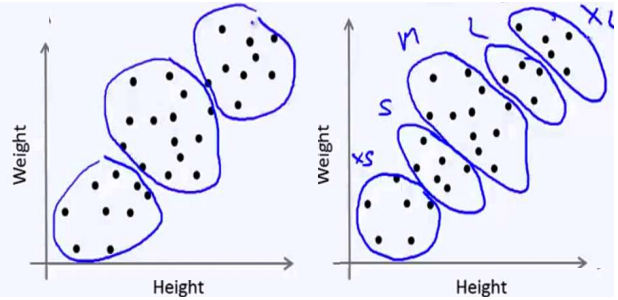
K值



- 根据用需求户定义
- 肘部法则

K值的选择

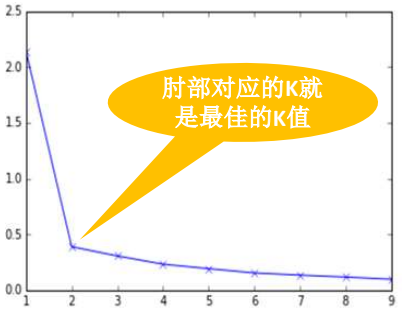
- 根据用需求户定义



K值的选择

- 肘部法则

把不同K值的成本函数值Je画出来。
随着K值的增大，Je会减小；每个类包含的样本数会减少，于是样本离其重心会更近。
但是，随着K值继续增大，Je的的变化趋缓。K值增大过程中，Je下降幅度最大的K位置对应的K值就是肘部。

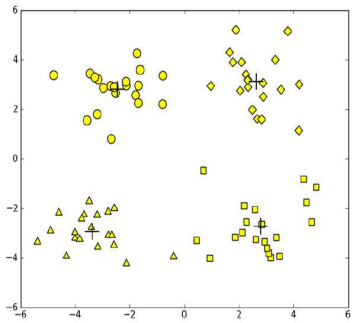


K均值实战

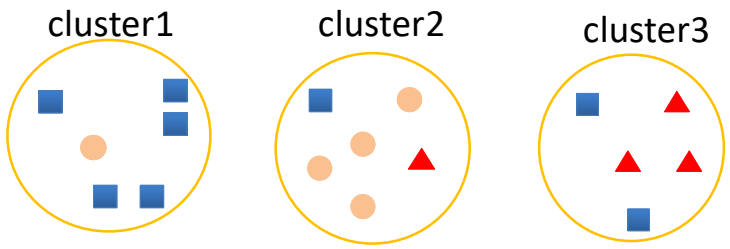
- 主要子函数
  - 读入数据： loadDataSet(filename)
  - 随机选择簇中心: randCent(dataSet,k)
  - 计算每个点间的距离: distElucd(vecA.vecB)

- kmeans函数：
  - myCentroids, clustAssing = kMeans(dataSet,k,distMeans,createCent=randCent)

```
» datMat=mat(kMeans.loadDataSet('testSet.txt'))
» myCentroids, clustAssing = kMeans.kMeans(datMat,4)
```



评价指标



$$\text{Purity}\ (\text{W},\ \text{C}) = \frac{1}{\text{N}} \sum_{\text{k}} \max_{\text{j}} \left| w_k \cap c_j \right|$$

评价指标

- Purity

  Purity方法的优势是方便计算，值在0～1之间，完全错误的聚类方法值为0，完全正确的方法值为1。同时，Purity方法的缺点也很明显它无法对退化的聚类方法给出正确的评价，设想如果聚类算法把每篇文档单独聚成一类，那么算法认为所有文档都被正确分类，那么purity值为1！而这显然不是想要的结果。

评价指标

- RI：一种用排列组合原理来对聚类进行评价的手段

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

其中：TP指被聚在一类的文档被正确分类了，TN是不应该聚在一类的文档被正确分开了，FP指不应该放在一类的文档被错误的放在一类，FN只不应该分开的文档被错误地分开了

F-measure

- 基于上述RI方法衍生出的一个方法

$$F_\beta = \frac{(\beta^2 + 1)\, PR}{\beta^2 \cdot (P + R)}$$

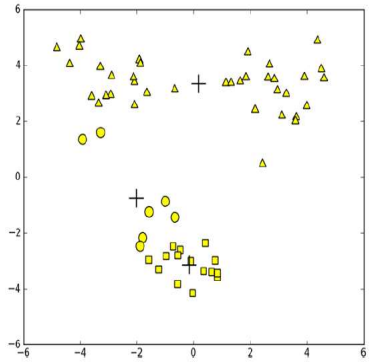
- RI方法有个特点就是把准确率和召回率看得同等重要，事实上有时候我们可能需要某一特性更多一点，这时候就适合F值方法

实战

- http://blog.csdn.net/cjianwyr/article/details/54907144

算法的优缺点

- 优点：容易实现。
- 缺点：
  - 受初始中心点影响，可能收敛局部最优
  - 适用数据类型：数值型数据



改进算法

- K-modes
- 二分-k均值

k-modes算法

- k-modes算法把k-means算法扩展到可分类数据
- k-modes算法中的中心定义：
  - 根据可分类属性值出现的频率更新聚类中心，聚类中出现频率最高的属性值被选为聚类中心，即modes（类模式）。
- k-modes算法的距离计算方法：
  - 假设***X***，***Y***是数据集中的两个对象，它们用*m*维属性描述，则这两个对象之间的相异度为：

$$d(X,Y)=\sum_{j=1}^{m}\delta(x_j,y_j)$$

$$\text{当}x_j=y_j,\delta(x_j,y_j)=0;\text{当}x_j\neq y_j,\delta(x_j,y_j)=1$$

二分K-均值

- 目的：克服只适应于数值型数据
- 伪代码：
  - 将所有点看成一个簇
  - 当簇的数目小于k时
  - 对于每个簇
    - 计算总误差
    - 在给定的簇上面进行K-均值聚类（k=2）
    - 计算将该簇一分为二之后的总误差（SSE）
  - 选择使得误差最小的那个簇进行划分

二分K-均值

- 目的：克服收敛于局部最小的缺陷
- 伪代码：
  - 将所有点看成一个簇
  - 当簇的数目小于k时
  - 对于每个簇
    - 计算总误差
    - 在给定的簇上面进行K-均值聚类（k=2）
    - 计算将该簇一分为二之后的总误差（SSE）
  - 选择使得误差最小的那个簇进行划分