信息检索
Information Retrieval

第八章 分类和聚类

引言

- 图书馆管理员面临的古老的问题
  - 将文档存储起来以供日后检索
  - 对于较大的集合，需要对文档进行标记
    - 为每个文档分配一个唯一的标识符
    - 无法通过主题来查找文档
- 通过主题对文档进行搜索
  - 按主题对文档进行分组
  - 对每个组用一个有意义的标签命名
  - 每个通过标签命名的组成为类别

引言

- 文本分类
  - 将文档与类别相关联的处理过程
  - *Text Classification*与*Text Categorization*（编目方法）等价
- 相关问题：将文档集合划分为没有标签的子集
  - 因为每个子集都没有标签，所以它不是类别
  - 这样每个子集被称为一个集群（cluster，簇）
  - 文档集合划分为子集的过程称为聚类（Clustering）
    - 通常将文本聚类看做是文本分类的一个简单的变体

引言

- 机器学习
  - 学习数据模式的算法
  - 学习的模式用于对新的数据进行预测
  - 学习算法分为
    - 有监督学习（Supervised Learning）
      - 有标注的训练数据
    - 无监督学习（unsupervised Learning）
      - 无训练数据
    - 半监督学习（Semi-supervised Learning）
      - 小规模的训练数据，大量无标注的数据

引言

- 文本分类问题
  - 分类器形式化定义如下：
    - D：文档集合
    - $C = \{c_1, c_2, \ldots, c_L\}$：类别的集合
    - 分类器是一个二元函数$F: D \times C \rightarrow \{0, 1\}$，
      - $[d_j, c_p] = 1$，如果d_j属于类别c_p
      - $[d_j, c_p] = 0$，否则

引言

- 分类模式
  - 2类问题，属于或不属于(binary)
  - 多类问题，多个类别(multi-class)，可拆分成2类问题
    - 一个文本可以属于多类(multi-label)
- 分类体系一般人工构造
  - 政治、体育、军事
  - 中美关系、恐怖事件
  - 很多分类体系：Reuters分类体系、中图分类

引言

TP312：计算机程序设计语言、算法语言

3：计算机；1：计算机软件；2：程序语言、算法语言

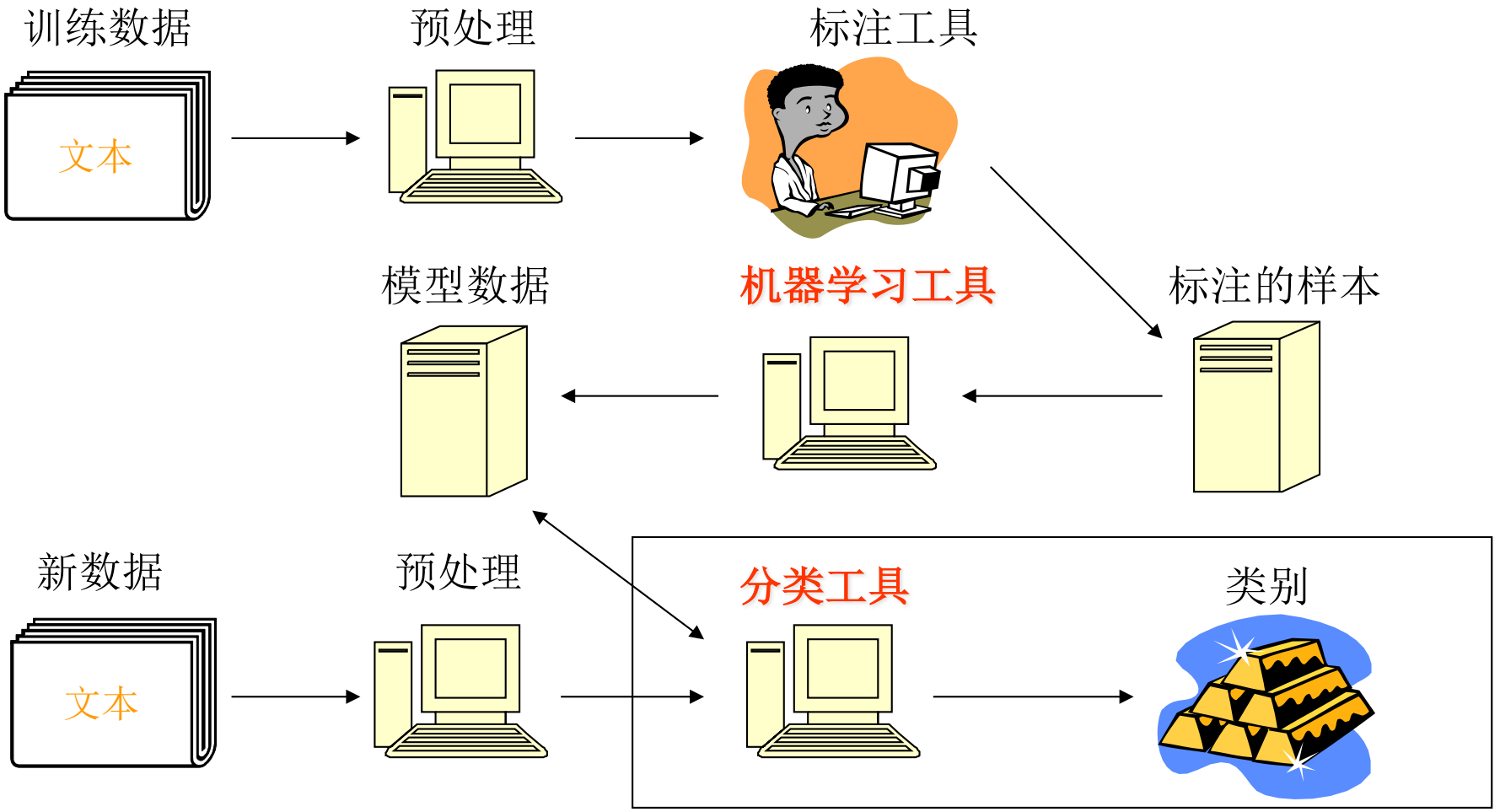
中图分类法

A类 马列主义、毛泽东思想
B类 哲学
C类 社会科学总论
D类 政治、法律
E类 军事
F类 经济
G类 文化、科学、教育、体育
H类 语言、文字
I类 文学
J类 艺术
K类 历史、地理
N类 自然科学总论
O类 数理科学和化学
P类 天文学、地球科学
Q类 生物科学
R类 医药、卫生
S类 农业科学
U类 交通运输
V类 航空、航天
X类 环境科学、劳动保护科学（安全科学）

TB类 一般工业技术
TD类 矿业工程
TE类 石油、天然气工业
TF类 冶金工业
TG类 金属学、金属工艺
TH类 机械、仪表工艺
TJ类 武器工业
TK类 动力工业
TL类 原子能技术
TM类 电工技术
TN类 无线电电子学、电信技术
TP类 自动化技术、计算技术
TQ类 化学工业
TS类 轻工业、手工业
TU类 建筑科学
TV类 水利工程

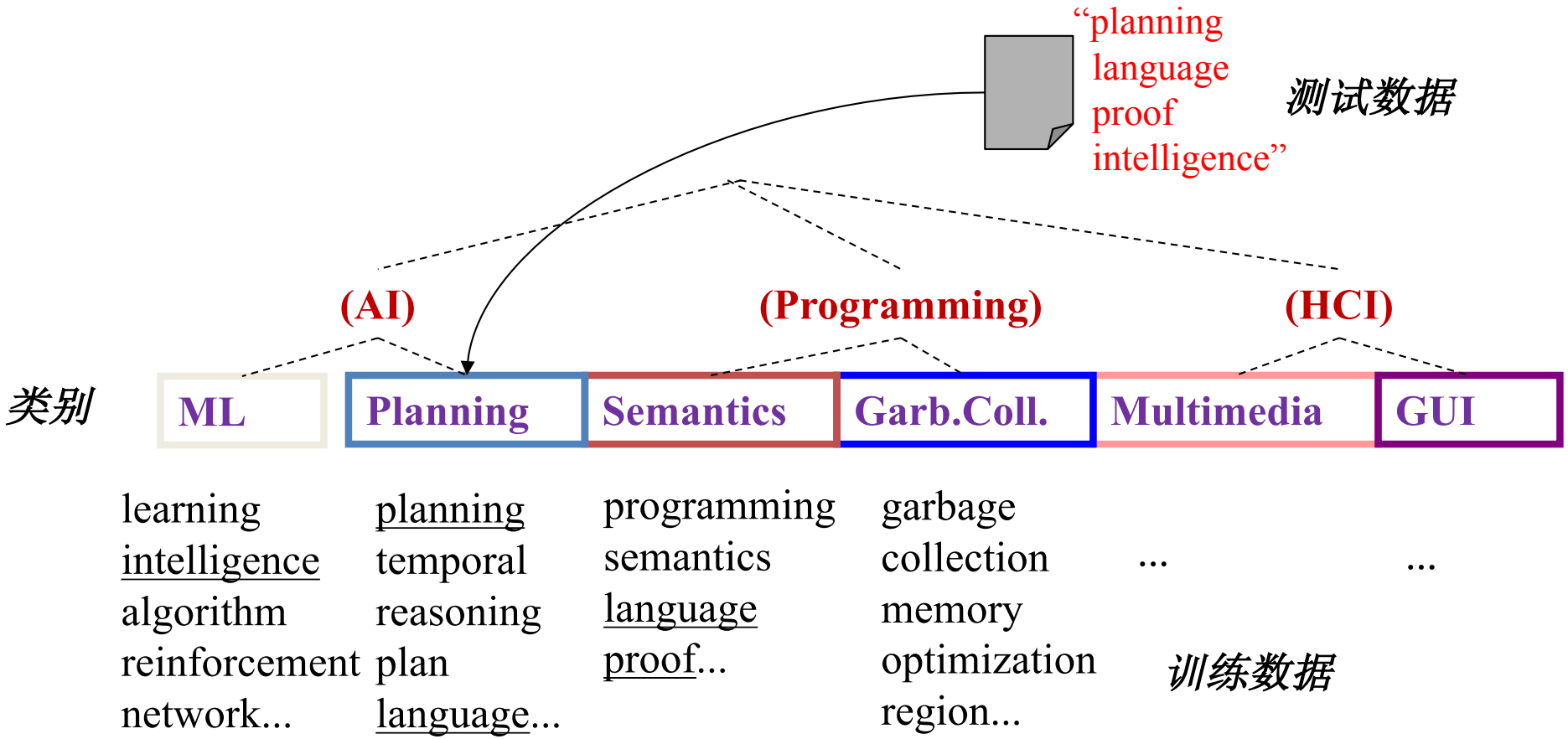
引言

- 分类系统的流程



引言

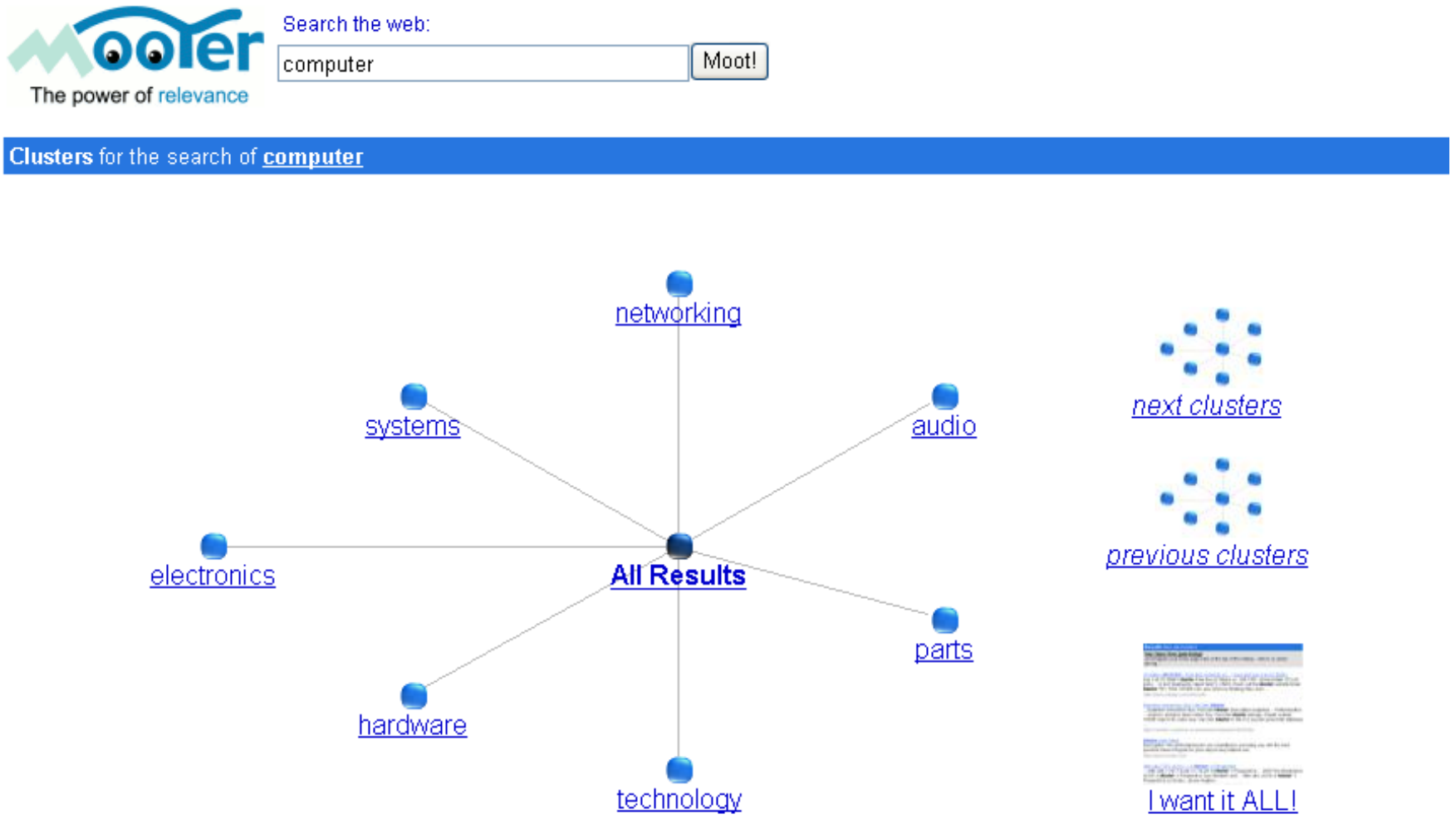
- 文本分类示例


"planning
language
proof
intelligence"
测试数据
(AI)
(Programming)
(HCI)
类别
ML
Planning
Semantics
Garb.Coll.
Multimedia
GUI
learning
intelligence
algorithm
reinforcement
network...
planning
temporal
reasoning
plan
language...
programming
semantics
language
proof...
garbage
collection
memory
optimization
region...
...
...
训练数据

主要内容

- 聚类方法
- 特征选择方法
- 文本分类方法
- 文本分类的评价

聚类方法


Search the web:
computer
Moot!
The power of relevance
Clusters for the search of computer
networking
systems
audio
electronics
All Results
parts
hardware
technology
next clusters
previous clusters
I want it ALL!

聚类方法

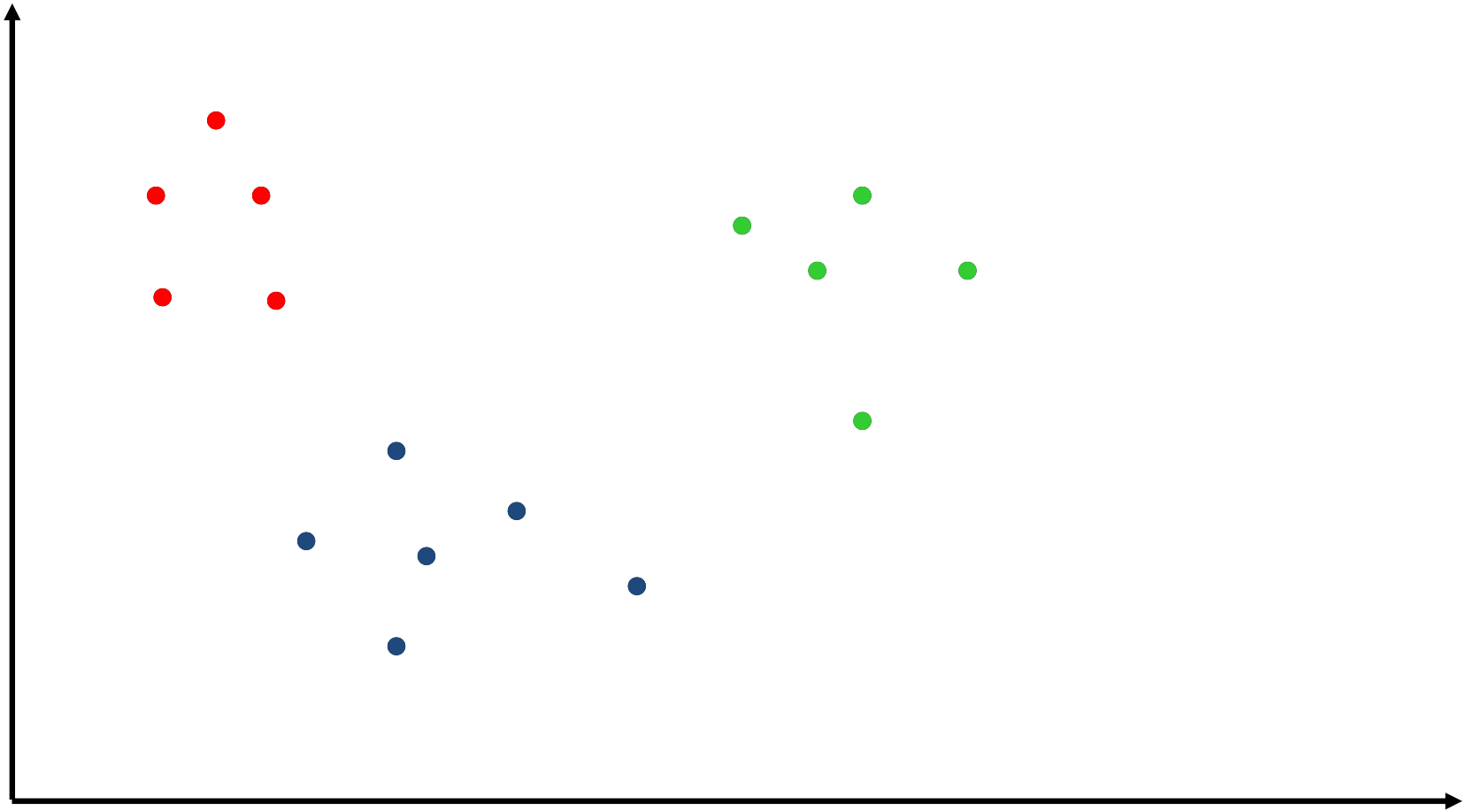

bbmao
超搜索
网页 求职 会员收藏
计算机
去抓
bbmao Google YAHOO! Bai百度 Sogou iASK爱问 结果说明
分类
所有结果[提示]
+ 考试 (25)
+ 大学 (19)
- 电脑 (19)
- 教育 (19)
+ 软件 (23)
- 培训 (11)
- 北京 (16)
- 学校 (7)
- 图书 (8)
- 计算机网 (10)
- 安全 (11)
- 国家 (11)
- 上海 (6)
- 科技 (13)
- 电子 (10)
- 科学 (6)
- 深圳 (6)
- 华南 (4)
- 管理 (10)
- 工作 (9)
更多...
意见反馈
预览网页
收藏网页
浏览快照
搜索 计算机第 1 - 10 项。
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
发现好东西了？点击 '收藏' 把它收藏到 bbmao，共享给大家，也方便自己找回。
|><| 太平洋电脑网PConline.com.cn->IT世界由此精彩
中国最具知名度和影响力的IT门户网站，根据权威网站流量数据机构ALEXA及中国互联网实验 室数据显示，PConline的日均浏览量排在中国IT类网站第一，全球网站50强
PConline下 设今日报价、IT新闻、数码世界手机、笔记本、硬件资讯、软件资讯、下载、通讯、 ...
www.pconline.com.cn - Google 1
电脑报——发行量第一的计算机报
论坛登录. 无安全提问，母亲的名字，爷爷的名字，父亲出生的城市，您其中一位老师的名字，您个人计算机的型号，您最喜欢的餐馆名称，驾驶执照的最后四位数字. 宣传文
首页 ▪ 新闻评论 ▪ 整机外设 ▪ 硬件评测 ▪ 软件网络 ▪ 数字娱乐 ▪ 消费电子 ...
www.cpcw.com - Google 2，搜狗 4
电脑之家PChome | 科技引领生活
中国最具知名度和影响力的IT门户网站，热门社区，模特热图，生活自拍，全球网站50强内。 今日报价、IT新闻、数码世界、手机、笔记本、硬件资讯、软件资讯、软件下
载、通讯、 渠道商情、解决方案、招聘培训、产品调查和二手等频道，并拥有最大型的IT产品资料 ...
www.pchome.com - Google 3
没有找到你要的东东吗？请点击左边的分类！
欢迎访问中国计算机报网站！
包括该报的即日和过去的报道的全文浏览和搜索功能。
www.ciw.com.cn - Google 4，百度 5，雅虎 2，搜狗 3
天极Yesky_全球中文IT第一门户
提供新闻、游戏、IT、影视和生活资讯及相关产品销售。
www.yesky.com - Google 5
网上订购戴尔计算机，超强配置免费升级
现在登录戴尔网站订购台式机，即可享受超值现金优惠，还有多重免费升级配置，让您惊喜不断！ 欢迎拨打24小时免费电话：800-858-0488。手机拨打 400-889-716
收市话费用
sf.baidu.com/baidu.php?url=&e=dj0xJms9NDU3OTU5MyZz... - 百度 1

聚类方法

- 聚类
  - 将无标记的样本划分到聚类的各个子集中
    - 类内样本非常相似
    - 类间样本非常不同
  - 通过无监督的方法发现新类别

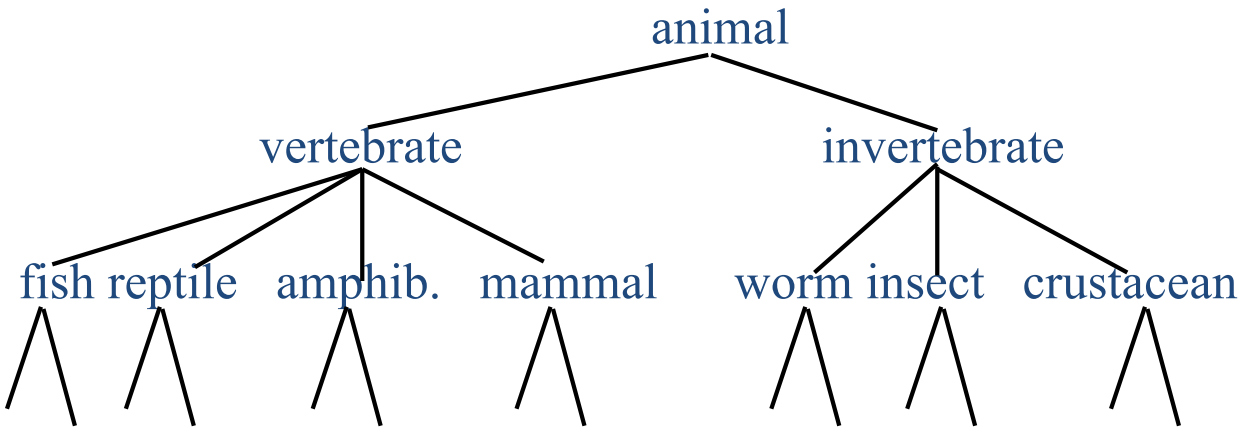
聚类方法

- 聚类样例



聚类方法

- 聚类方法—层次聚类（Hierarchical Clustering）
  - 在无标注的样本集合中建立树状层次分类结构
    - 会聚(bottom-up)
      - 以每个样本独自一类开始，迭代合并到越来越大的类中
    - 分裂 (partitional, top-down)
      - 将所有样本不断划分到类别中



聚类方法

- 聚类方法—层次聚类（Hierarchical Clustering）
  - 层次聚类的一般过程
    1. 输入
       - 包括*N*个文档的集合
       - *n*×*n*相似度（距离）矩阵
    2. 将每个文档分配到一个簇中（Cluster）
       - 会生成*N*个簇，每个簇中包含一个文档
    3. 找到两个距离最近的簇
       - 将这两个簇合并成一个簇，簇的数量减少为*N-1*
    4. 从新计算新生成的簇和已有的簇之间的距离
    5. 重复步骤*3*和*4*，直到产生一个包含*N*个文档的簇

聚类方法

- 聚类方法—层次聚类（Hierarchical Clustering）
  - 步骤4提到了需要计算两个簇之间的相似性（距离）的概念
  - 用于计算簇之间距离的方法
    - 单链接（Single-Link）
    - 全链接（Complete-Link）
    - 平均链接（Average-Link）

聚类方法

- $dist(c_p, c_r)$：两个簇c_p和c_r之间的距离
- $dist(d_j, d_l)$：两个文档d_j和d_l之间的距离

聚类方法—层次聚类（Hierarchical Clustering）

单链接（Single-Link）

$$dist(c_p, c_r) = \min_{\forall d_j \in c_p, d_l \in c_r} dist(d_j, d_l)$$

- 两个簇的距离为两个簇中最相似的文本之间的距离

全链接（Complete-Link）

$$dist(c_p, c_r) = \max_{\forall d_j \in c_p, d_l \in c_r} dist(d_j, d_l)$$

- 两个簇的距离为两个簇中相似度最小的两个文本之间的距离

平均链接（Average-Lin

$$dist(c_p, c_r) = \frac{1}{n_p + n_r} \sum_{d_j \in c_p} \sum_{d_l \in c_r} dist(d_j, d_l)$$

- 两个簇的距离为两个簇中文档之间的平均相似度

聚类方法

- 聚类方法—非层次聚类（non-Hierarchical Clustering）
  - 非层次聚类的一般过程
    - 需要确定期望的类别数k
    - 随机选择k个种子
    - 进行初始聚类
    - 迭代，将样例重新划分
    - 直到样例所属的类别不再改变

聚类方法

- 聚类方法—非层次聚类（non-Hierarchical Clustering）
  - *K-Means*算法
    - 假设给定的样例是一个实值向量
      - 文档可以表示成向量
    - 基于质心或者类别c中的样本均值进行聚类

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

    - 根据样例与当前类别质心的相似度重新划分类别
      - 余弦相似度、欧氏距离.....

聚类方法

- 聚类方法—非层次聚类（non-Hierarchical Clustering）
  - *K-Means*算法执行过程

令 d为两个样例的距离度量

选择 k 个随机样例$\{s_1, s_2, \ldots s_k\}$ 作为种子

直到聚类收敛或满足停止策略:

对每个样例 x_i:

将 x_i 分配到 c_j ， $d(x_i, s_j)$ 是最小的

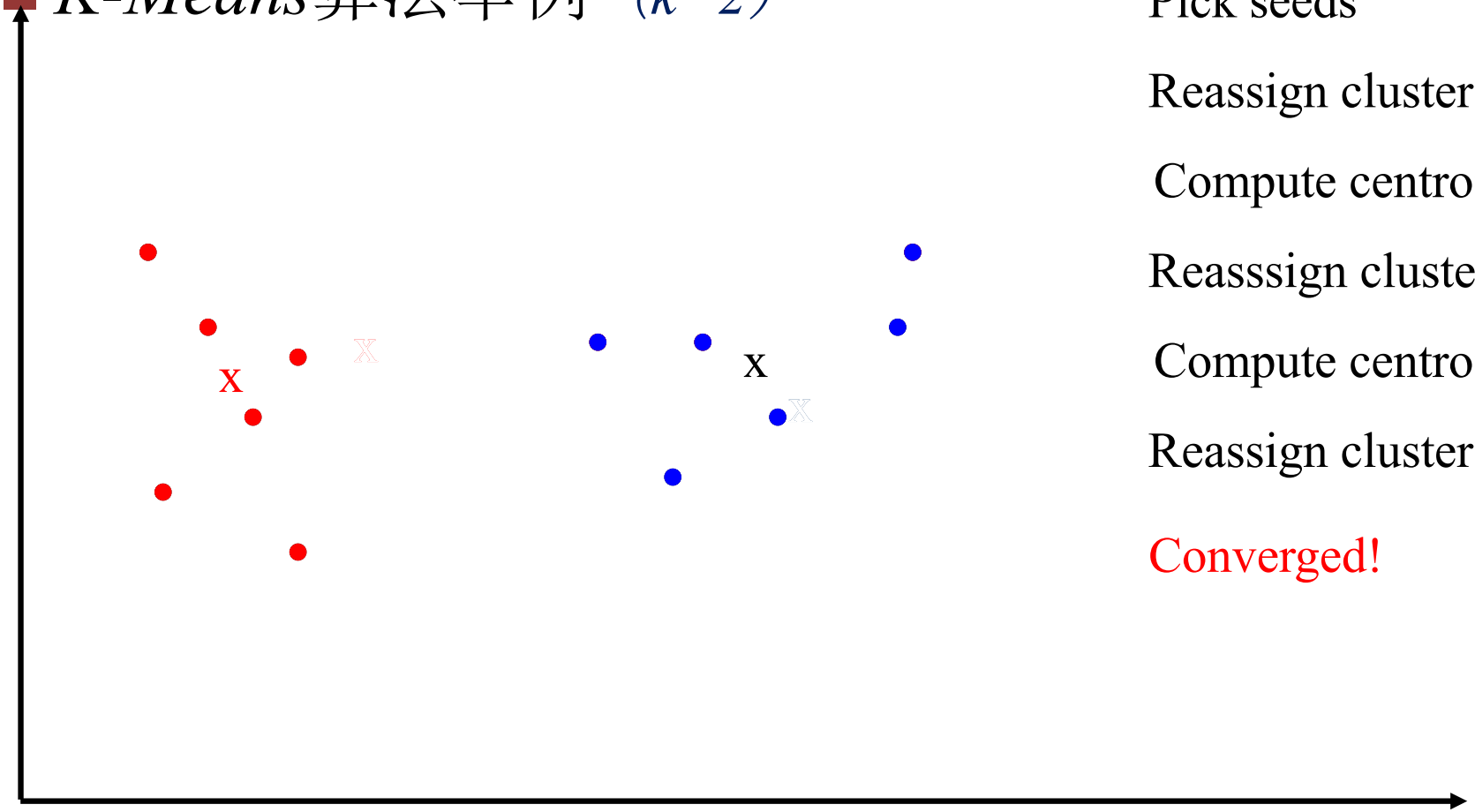
(*Update the seeds to the centroid of each cluster*)

对每个类 c_j

$s_j = \mu(c_j)$

聚类方法

- 聚类方法—非层次聚类（non-Hierarchical Clustering）
  - *K-Means*算法举例 *(k=2)*



聚类方法

- 聚类方法—非层次聚类（non-Hierarchical Clustering）
  - *K-Means*算法
    - 种子的选择
      - 聚类结果与随机种子的选择是相关的
      - 随机选择的种子可能会导致收敛很慢或者收敛到局部最优
      - 采用启发式方法或其他方法选择好的种子

聚类方法

- 层次聚类方法（HAC）和 K-Means算法可以直接应用于文本聚类中
  - 可以使用归一化、基于TF/IDF权重的向量以及余弦相似度等方法
- 文本聚类技术的应用:
  - 在检索阶段，加入同一类别的其他文本作为初始检索结果，提高召回率
  - 检索结果进行聚类，可以提供给用户更好的组织形式
  - 自动生成的层次聚类结果为用户提供方便，也可以根据聚类结果生成多文档文摘

主要内容

- 聚类方法
- 特征选择方法
- 文本分类方法
- 文本分类的评价

特征选择方法

- 较大的特征空间可能会使文档分类器在实际中无法工作
- 经典解决方案
  - 选择表示文档的所有特征的一个子集，称为特征选择（特征提取）
    - 减少文档表示的维度
    - 减少过度拟合（只适用于训练集，难以泛化）

特征选择方法

- 特征选择
  - 依赖于词项在文档和类别中出现的频率
    - D_t：训练文档集合
    - N_t：训练文档集合D_t中文档的数目
    - t_i：训练文档集合D_t中包含词项k_i的文档数目
    - $C = \{c_1, c_2, \ldots\ldots, c_L\}$：类别的集合
    - $T : D_t \times C \rightarrow [0, 1]$：训练集中函数的结果（实际值）
- 词项—类别偶然事件表（Term-Class Incidence Table）

特征选择方法

- 词项—类别偶然事件表（Term-Class Incidence Table）

	c_p中的文档数目	不在c_p中的文档数目	合计
包含词项k_i的文档数目	$n_{i,p}$	n_i-$n_{i,p}$	n_i
不包含词项k_i的文档数目	n_p-$n_{i,p}$	N_t-n_i-(n_p-$n_{i,p}$)	N_t-n_i
所有文档数目	n_p	N_t-n_p	N_t

- $n_{i,p}$：包含词项k_i的文档分到了类别c_p中的数目
- n_i-$n_{i,p}$：包含词项k_i的文档没有分到类别c_p中的数目
- n_p：类别c_p中文档的总数
- n_p-$n_{i,p}$：类别c_p中不包含词项k_i的文档数目

	c_p中的文档数目	不在c_p中的文档数目	合计
包含词项k_i的文档数目	$n_{i,p}$	n_i-$n_{i,p}$	n_i
不包含词项k_i的文档数目	n_p-$n_{i,p}$	N_t-n_i-(n_p-$n_{i,p}$)	N_t-n_i
所有文档数目	n_p	N_t-n_p	N_t

- 词项—类别偶然事件表（Term-Class Incidence Table）
  - $k_i \in d_j$的概率：$P(k_i) = \frac{n_i}{N_t}$
  - $k_i \notin d_j$的概率：$P(\overline{k_i}) = \frac{N_t - n_i}{N_t}$
  - $d_j \in c_p$的概率：$P(c_p) = \frac{n_p}{N_t}$
  - $d_j \notin c_p$的概率：$P(\overline{c_p}) = \frac{N_t - np}{N_t}$
  - $k_i \in d_j \; and \; d_j \in c_p$的概率：$P(k_i, c_p) = \frac{n_{i\,p}}{N_t'}$
  - $k_i \notin d_j \; and \; d_j \in c_p$的概率：$P(\overline{k_i}, c_p) = \frac{n_p - n_{i\,p}}{N_t'}$
  - $k_i \in d_j \; and \; d_j \notin c_p$的概率：$P(k_i, \overline{c_p}) = \frac{n_i - n_{i\,p}}{N_t'}$
  - $k_i \notin d_j \; and \; d_j \notin c_p$的概率：$P(\overline{k_i}, \overline{c_p}) = \frac{N_t - n_i - (n_p - n_{i\,p})}{N_t'}$

特征选择方法

- 特征选择—文档频率（Document Frequency）
  - $DF(\mathrm{t_k})$：n_i / N_t
    - n_i：出现词项的文档数；N_t：文本集合中文档总数
  - 特征选择过程
    - 设定文档频率DF的上界阈值∂_u和下界阈值∂_l
    - 统计训练数据集合中词项的文档频率
      - $\forall DF(\mathrm{t_k}) < \partial_l$：词项$t_k$在训练集中出现的频率过低，不具有代表性，因此从特征空间中去掉
      - $\forall DF(\mathrm{t_k}) > \partial_u$：词项$t_k$在训练集中出现的频率过高，不具有区分度，因此从特征空间中去掉
    - 最终选择的词项：$\partial_l \le DF(\mathrm{t_k}) \le \partial_u$

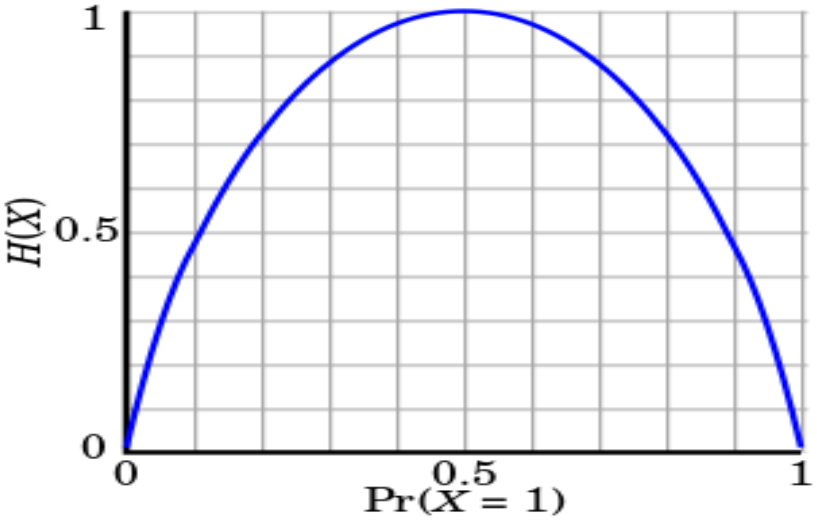
特征选择方法

- 特征选择—文档频率（Document Frequency）
  - 基于文档频率的特征选择方法特点
    - 方法简单、易实现
    - 理论依据不严密
      - 某些词项虽然出现频率低，但却含有较多的信息，对分类有帮助
        - 如：信息检索中“Polite Policy”（礼貌策略）

特征选择方法

- 特征选择—TF-IDF
  - $w_{i,j}$： 词项k_i在文档d_j中的TF-IDF权重
  - K_{th}： TF-IDF权重的阈值
  - 特征选择过程
    - 保留所有的词项k_i（ $w_{i,j} \geq K_{th}$）
    - 低于阈值的其他词项抛弃掉
  - TF-IDF特征选择方法的缺点
    - 文档频率大的词项无用，这是不完全正确的
    - 没有体现词项的位置信息
      - Web网页中，HTML具有结构特征，不同位置的特征词应该赋予不同的系数

特征选择方法



- 特征选择—信息增益（Infor
  - 对训练集中所有词语量化其重要程度
  - 信息熵（Entropy）
    - 假设有一个变量X，它的取值有n种
      - $x_1, x_2, \ldots\ldots, x_n$
      - 每一种取值的概率分别为$P_1, P_2, \ldots\ldots, P_n$
    - X的熵定义为：$H(X) = -\sum_{i=1}^{n} P_i \times log_2 P_i$
      - X的取值可能性越多，携带的信息量越大
        - 假设X代表一个问题，这个问题有10中答案，我们需要较多的信息才能准确地知道答案

特征选择方法

- 特征选择—信息增益（Information Gain）
  - 信息熵（Entropy）（续）
    - 在文本分类中，类别C是变量，可能的取值
      - $c_1, c_2, \ldots\ldots, c_n$
      - 每一个类别出现的概率分别为$P(c_1), P(c_2), \ldots\ldots, P(c_n)$
    - 整个分类系统的信息熵为

$$H(C) = -\sum_{i=1}^{n} P(c_i) \times log_2 P(c_i)$$

      - $P(c_i)$表示C_i类中包含文档数量占全部文档数量的比例

特征选择方法

在分类系统中，特征t确定后：

$$H(C|T) = P_t \times H(C|t) + P_{\bar{t}} \times H(C|\bar{t})$$

- 特征选择—信息增益（Information Gain）
  - 条件熵
    - 给定一个特征词t，系统的信息量是多少？
      - 假设有一个变量X，它的取值有n种
        - $x_1, x_2, \ldots\ldots, x_n$
        - 如何计算条件熵？每个值都可以作为一个条件
          - 计算n个值，然后取均值（不是简单的求平均数）
          - 而是要用每个值出现的概率来算平均（一个值出现的可能性比较大，计算出来的信息量占的比重要大一些）
    - 条件熵计算如下

$$H(C|X) = P_1 \times H(C|X = x_1) + P_2 \times H(C|X = x_2) + \cdots + P_n \times H(C|X = x_n)$$

特征选择方法

- 特征选择—信息增益（Information Gain）
  - 信息增益

$$IG(T) = H(C) - H(C|T) = -\sum_{i=1}^{n} P(c_i)\times log_2 P(c_i) - P_t \times H(C|t) - P_{\bar{t}} \times H(C|\bar{t})$$

$$= -\sum_{i=1}^{n} P(c_i)\times log_2 P(c_i) + P_t \sum_{i=1}^{n} P(c_i|t)\times log_2 P(c_i|t) + P_{\bar{t}} \times \sum_{i=1}^{n} P(c_i|\bar{t})\times log_2 P(c_i|\bar{t})$$

- $P(c_i)$表示C_i类中包含文档数量占全部文档数量的比例
- P_t表示包含词项t的文档数量占全部文档数量的比例
- $P_{\bar{t}}$表示不包含词项t的文档数量占全部文档数量的比例
- $P(c_i|t)$包含词项t的c_i类中文档数量占全部文档集合中包含词项t的文档数量的比例

特征选择方法

- 特征选择—信息增益（Information Gain）
  - 信息增益方法的特点
    - 在文本集合中，分布类别比较广泛的词语信息增益都比较高
      - t属于类别C_i的不确定性比较大，t包含的信息量也比较大
      - 当t确定后，如果是一个好的特征，条件熵会比较小，信息增益就会增大
    - 信息增益在计算过程中，获得的是整个训练语料的特征词，不能区别不同类别间特征词的权重
      - 每个类别需要有自己的特征集合，有些特征对这个类别有区分度，对其他类别可能毫无用处

特征选择方法

- 特征选择—互信息（Mutual Information）
  - 度量两个变量之间的相关性
    - 词项t_k在类别c_i中出现频率较高，而在其他类别中出现频率较低，则t_k与c_i的互信息较大
  - 互信息
    - $I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}$
      - 如果x和y相互独立，则互信息为0
      - 如果x和y相关性较大，互信息也较大
    - 应用到文本分类特征选择
    - $I(U;C) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0.1\}} P(U = e_t, C = e_c) log \frac{P(U=e_t, C=e_c)}{P(U=e_t)P(C=e_c)}$

$log \frac{P(x|y)}{P(x)}$

$$I(U;C)=\sum_{e_t\in\{0,1\}}\sum_{e_c\in\{0.1\}}P(U=e_t,C=e_c)log\frac{P(U=e_t,C=e_c)}{P(U=e_t)P(C=e_c)}$$

● 特征选择—互信息 (Mutual Information)

$$I(U;C)=\frac{N_{11}}{N}log_2\frac{NN_{11}}{N_{1.}N_{.1}}\quad e_t=1,\ e_c=1$$

$$+\frac{N_{01}}{N}log_2\frac{NN_{01}}{N_{0.}N_{.1}}\quad e_t=0,\ e_c=1$$

$$+\frac{N_{10}}{N}log_2\frac{NN_{10}}{N_{1.}N_{.0}}\quad e_t=1,\ e_c=0$$

$$+\frac{N_{00}}{N}log_2\frac{NN_{00}}{N_{0.}N_{.0}}\quad e_t=0,\ e_c=1$$

$e_t=1$，$e_c=1$
包含词项t的文档属于类别c的概率
$P(t,c)=N_{11}/N$
$P(t)=N_{1.}/N$　　$P(c)=N_{.1}/N$
因此得到

$$\frac{N_{11}}{N}log_2\frac{NN_{11}}{N_{1.}N_{.1}}$$

其他项同理

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

N_{xy}表示的是$x=e_t$和$y=e_c$情况下对应的文档数目

特征选择方法

- 特征选择—互信息（Mutual Information）
  - 特征选择互信息方法存在的问题
    - “低频词依赖现象”
      - 训练集中出现很少的词，互信息值很大
        - 只在某个类别中的部分文档中出现
      - 这些词可能是输入错误的词或者系统分词错误的词语
        - 对未标注样本进行测试时，很难找到这样的词语

特征选择方法

- 特征选择—卡方检验（Chi Square, χ^2）
  - 假设检验的方法
    - 假设两个变量确实是独立的
    - 观察实际值（观察值）与理论值（“如果两者确实独立”的情况下应该有的值）的偏差程度
      - 如果偏差足够小，则认为是测量手段不够精确导致或者偶然发生的，两者确实是独立的
      - 如果偏差大到一定程度，使得这样的误差不太可能是偶然产生或者测量不精确所致，则认为两者实际上是相关的

特征选择方法

- 特征选择—卡方检验（Chi Square, χ^2）

① 观察值A和理论值E之间的偏差，就是二者的差

- 将多个观察值和理论值的偏差求和
- $\chi^2 = \sum_{i=1}^{k}(A_i - E_i)$

② 差值有正有负，会相互抵消。本来有偏差，会变成没有偏差，因此，加上平方后再求和

- $\chi^2 = \sum_{i=1}^{k}(A_i - E_i)^2$

③ 上述公式中，如果均值为500，相差5是很小的（1%）；如果均值为20，相差5比较大（25%）。归一化

- $\chi^2 = \sum_{i=1}^{k}\frac{(A_i - E_i)^2}{E_i}$ 最终的公式

特征选择	属于“体育”	不属于“体育”	总计
包含“篮球”	A	B	A+B
不包含“篮球”	C	D	C+D
总数	A+C	B+D	N

- 特征选择—卡方检验（Chi Square, χ^2）
  - 计算四格表中E_{11}（包含“篮球”且属于“体育”类）
    - 观察值：A
    - 理论值
      - 假设两个变量时独立的，也就是“篮球”和“体育”类没有关系（“篮球”在文档集合中接近等概率出现）
        - 即：（$A+B$）/N
        - 理论值为（$A+C$）（（$A+B$）/N）
    - 可以得到D_{11}的方差

$$D11 = \frac{\left(A - \frac{(A+C)(A+B)}{N}\right)^2}{\frac{(A+C)(A+B)}{N}}$$

特征选择方法

- 特征选择—卡方检验（Chi Square, χ^2）
  - χ^2(篮球，体育) $= D_{11} + D_{12} + D_{21} + D_{22}$
  - χ^2(篮球，体育) $= \dfrac{N(AD-BC)^2}{(A+C)(A+B)(B+D)(C+D)}$
  - 如果确定了文档集合和某一个类别
    - N、$A+C$、$B+D$的值就是确定的
      - 无论分类结果如何，属于该类别和不属于该类别的文档数目是固定的
    - χ^2(篮球，体育) $= \dfrac{(AD-BC)^2}{(A+B)(C+D)}$

特征选择方法

- 特征选择—卡方检验（Chi Square, χ^2）
  - 特征选择卡方检验存在的问题
    - “低频词缺陷”
      - A和B的值是怎么得出来的
        - 统计文档中是否出现词t，却不管t在该文档中出现了几次
        - 这会使得他对低频词有所偏袒（因为它夸大了低频词的作用）

主要内容

- 聚类方法
- 特征选择方法
- 文本分类方法
- 文本分类的评价

文本分类方法

- 文本分类方法—贝叶斯分类
  - 基于概率理论的学习和分类方法
  - 贝叶斯理论在概率学习及分类中充当重要角色
  - 仅使用每类的先验概率不能对待分的文本提供信息
  - 分类是根据给定样本描述的可能的类别基础上产生的后验概率分布

文本分类方法

- 文本分类方法—贝叶斯分类
  - 贝叶斯理论

$$P(H|E) = \frac{P(H \cap E)}{P(E)} \qquad P(E|H) = \frac{P(H \cap E)}{P(H)}$$

$$P(H \cap E) = P(E|H)P(H)$$

得到：

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

文本分类方法

- 文本分类方法—贝叶斯分类
  - 设各个类别的结合为 $\{c_1, c_2, \ldots\ldots, c_n\}$
  - 设E为实例的表述（文本）
  - 确定E的类别

$$P(c_i|E) = \frac{P(c_i)P(E|c_i)}{P(E)}$$

  - $P(E)$ 可以根据下式确定

$$\sum_{i=1}^{n} P(c_i|E) = \sum_{i=1}^{n} \frac{P(c_i)P(E|c_i)}{P(E)} = 1$$

$$P(E) = \sum_{i=1}^{n} P(c_i)P(E|c_i)$$

文本分类方法

$$P(c_i|E) = \frac{P(c_i)P(E|c_i)}{P(E)}$$

- 文本分类方法—贝叶斯分类
  - 需要计算：
    - 先验概率：$P(c_i)$ （某个类别c_i出现的概率）
    - 条件概率：$P(E|c_i)$ （给定类别c_i文本E出现的概率）
  - $P(c_i)$容易从数据中获得
    - 假设文档集合D中，属于类别c_i的样例数为n_i
    - 则有：$P(c_i) = n_i / |D|$
  - 假设样例的特征是关联的
    - $E = w_1 \wedge w_2 \wedge \cdots \wedge w_m$
    - 估计所有的$P(E|c_i)$

文本分类方法

- 文本分类方法—贝叶斯分类
  - 假定样例的特征是独立的（文本中词之间是独立的），则

$$P(E|c_i) = P(w_1 \wedge w_2 \wedge \cdots \wedge w_m|c_i) = \prod_{j=1}^{m} P(w_j|c_i)$$

  - 因此，只需要知道每个特征和类别的$P(w_j|c_i)$

文本分类方法

- 文本分类方法—贝叶斯分类
  - 朴素贝叶斯算法（训练过程）

设V为文档集合D所有词词表

对每个类别 $c_i \in C$

D_i是文档D中类别c_i的文档集合

$P(c_i) = |D_i| / |D|$ （计算类别c_i出现的先验概率）

设 n_i为D_i中词的总数

对每个词 $w_j \in V$

令 n_{ij} 为D_i中w_{ij}的数量

$P(w_j | c_i) = (n_{ij} + 1) / (n_i + |V|)$

文本分类方法

- 文本分类方法—贝叶斯分类
  - 朴素贝叶斯算法（测试过程）
    - 给定一个测试文档X
    - 设n为X中词的个数
    - 返回的类别：

$$\underset{c_i \in C}{argmax}\ P(c_i)\prod_{i=1}^{n} P(w_i|c_i)$$

      - w_i是X中第i个词

文本分类方法

- 文本分类方法—贝叶斯分类
  - 举例（1）
    - $C = \{allergy, cold, well\}$
    - e_1 = sneeze; e_2 = cough; e_3 = fever
    - 当前实例是：$E = \{sneeze, cough, \neg fever\}$

Prob	Well	Cold	Allergy
$P(c_i)$	0.9	0.05	0.05
$P(sneeze\|c_i)$	0.1	0.9	0.9
$P(cough\|c_i)$	0.1	0.8	0.7
$P(fever\|c_i)$	0.01	0.7	0.4

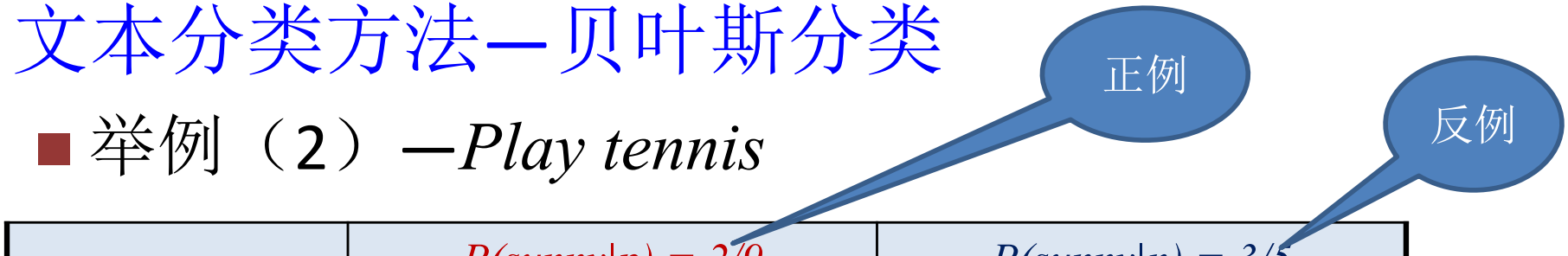
文本分类方法

Prob	Well	Cold	Allergy
$P(c_i)$	0.9	0.05	0.05
$P(sneeze|c_i)$	0.1	0.9	0.9
$P(cough|c_i)$	0.1	0.8	0.7
$P(fever|c_i)$	0.01	0.7	0.4

- 文本分类方法—贝叶斯分类
  - 举例（1）
    - 当前实例是：$E = \{sneeze, cough, \neg fever\}$
  - 参数计算
    - $P(well|E)=(0.9*0.1*0.1*0.99)/P(E)=0.0089/P(E)$
    - $P(cold|E)=(0.05*0.9*0.8*0.3)/P(E)=0.01/P(E)$
    - $P(allergy|E)=(0.05*0.9*0.7*0.6)/P(E)=0.019/P(E)$
    - $P(E)=0.0089+0.01+0.019=0.0379$
    - $P(well|E)=0.23$ $P(cold|E)=0.26$ $P(allergy|E)=0.50$

文本分类方法

- 文本分类方法—贝叶斯分类
  - 举例（2）—*Play tennis*



outlook	*P(sunny|p) = 2/9*	*P(sunny|n) = 3/5*
	P(overcast|p) = 4/9	*P(overcast|n) = 0*
	P(rain|p) = 3/9	*P(rain|n) = 2/5*
temperature	*P(hot|p) = 2/9*	*P(hot|n) = 2/5*
	P(mild|p) = 4/9	*P(mild|n) = 2/5*
	P(cool|p) = 3/9	*P(cool|n) = 1/5*
humidity	*P(high|p) = 3/9*	*P(high|n) = 4/5*
	P(normal|p) = 6/9	*P(normal|n) = 2/5*
wind	*P(true|p) = 3/9*	*P(true|n) = 3/5*
	P(false|p) = 6/9	*P(false|n) = 2/5*

文本分类方法

outlook	*P(sunny|p) = 2/9*	*P(sunny|n) = 3/5*
	P(overcast|p) = 4/9	*P(overcast|n) = 0*
	P(rain|p) = 3/9	*P(rain|n) = 2/5*
temperature	*P(hot|p) = 2/9*	*P(hot|n) = 2/5*
	P(mild|p) = 4/9	*P(mild|n) = 2/5*
	P(cool|p) = 3/9	*P(cool|n) = 1/5*
humidity	*P(high|p) = 3/9*	*P(high|n) = 4/5*
	P(normal|p) = 6/9	*P(normal|n) = 2/5*
wind	*P(true|p) = 3/9*	*P(true|n) = 3/5*
	P(false|p) = 6/9	*P(false|n) = 2/5*

- 文本分类方法—贝叶斯
  - 举例（2）—*Play tennis*
    - 给定一个实例$X=\{rain, hot, high, false\}$
      - $P(X|p)\cdot P(p)=P(rain|p)\cdot P(hot|p)\cdot P(high|p)\cdot P(false|p)\cdot P(p)$

        $=3/9\cdot 2/9\cdot 3/9\cdot 6/9\cdot 9/14=0.010582$
      - $P(X|n)\cdot P(n)=P(rain|n)\cdot P(hot|n)\cdot P(high|n)\cdot P(false|n)\cdot P(n)$

        $=2/5\cdot 2/5\cdot 4/5\cdot 2/5\cdot 5/14 = 0.018286$
    - 实例X被分到n类，即“不适合打网球”

文本分类方法

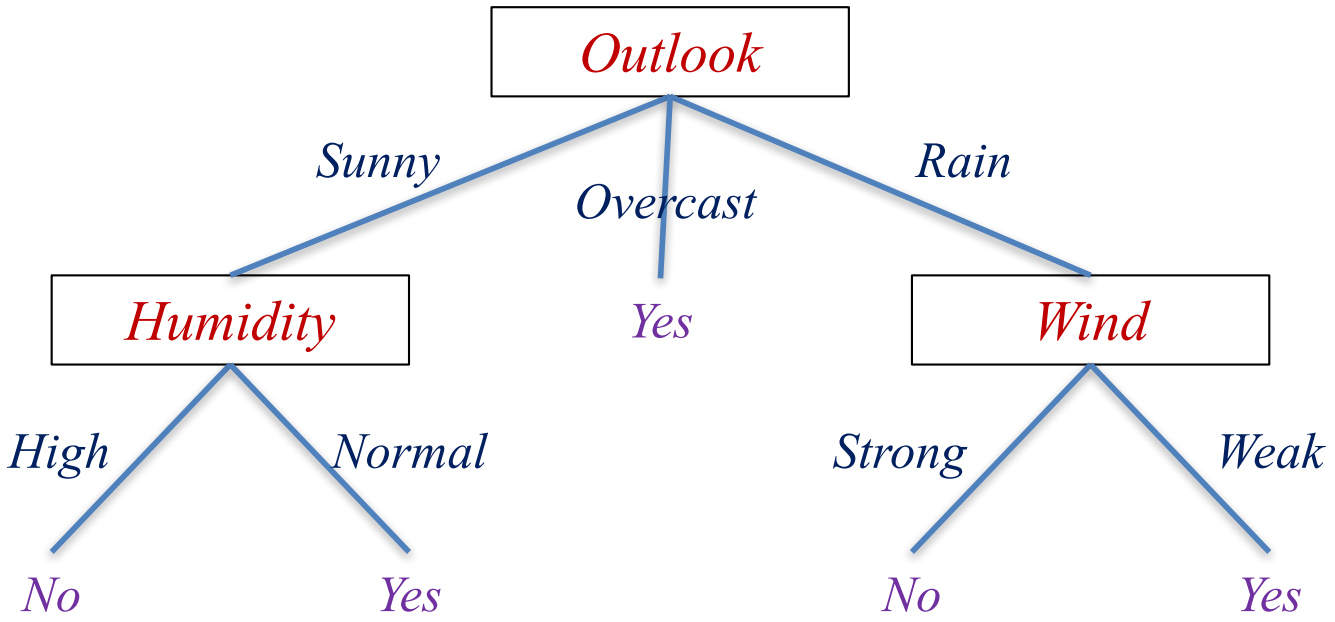
- 文本分类方法—贝叶斯分类
  - 贝叶斯分类的特点
    - 朴素的贝叶斯假定在一个位置上出现的词的概率独立于另外一个位置的单词
      - 这个假定有时并不反映真实情况
    - 虽然独立性假设很不精确，别无选择，否则计算的概率项将极为庞大
    - 幸运的是，在实践中朴素贝叶斯学习器在许多文本分类中性能非常好，即使独立性假设不成立

文本分类方法

- 文本分类方法—决策树
  - 决策树通过把实例从根节点排列到某个叶子节点来分类实例，叶子节点即为实例所属的分类
  - 树上的每一个节点说明了对实例的某个属性的测试
    - 该节点的每一个后继分支对应于该属性的一个可能值

文本分类方法

- 文本分类方法—决策树



$$(Outlook = Sunny \cap Humidity = Normal) \cup$$
$$(Outlook = Overcaset) \cup (Outlook = Rain \cap Wind = Weak)$$

文本分类方法

- 文本分类方法—决策树
  - 决策树方法的起源是概念学习系统*CLS*（Concept Learning System），然后发展到*ID3*方法
    - 常见的决策树方法有*CHAID*、*CART*、*Quest*、*C4.5*、*C5.0*……
  - 应用最广的归纳推理算法之一
  - 一种逼近离散值目标函数的方法

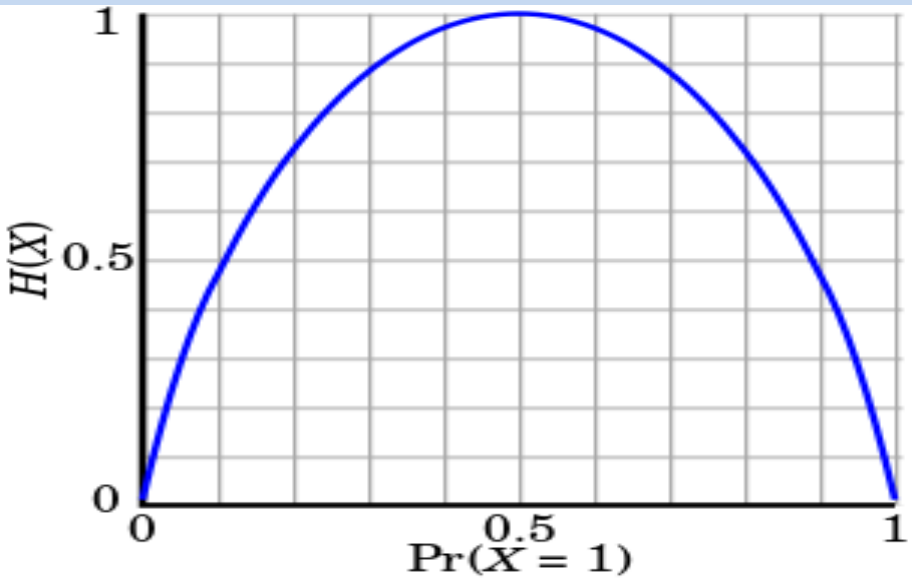
文本分类方法

- 文本分类方法—决策树
  - 实例是由属性-值对表示的
    - Humidity-*High*， Humidity-*Normal*
  - 目标函数具有离散的输出值： Yes，No
  - 可能需要析取的描述
    - $(Outlook = Overcaset) \cup (Outlook = Rain \cap Wind = Weak)$
  - 训练数据可以包含错误
    - 决策树有很好的鲁棒性，允许训练样例有少量错误
  - 训练数据可以包含缺少属性值的实例
    - 决策树学习可以在未知属性值的训练样本中使用（仅有部分训练样例知道当天的湿度）

文本分类方法

- 文本分类方法—决策树
  - 属性的选择（特征选择）
    - 构造好的决策树的关键在于如何选择好的逻辑判断或属性
    - 对于同样一组例子，可以有很多决策树能符合这组例子
      - 树越小则树的预测能力越强
    - 要构造尽可能小的决策树，关键在于选择恰当的逻辑判断或属性

文本分类方法



- 文本分类方法—决策树
  - 属性的选择
    - 熵的定义（回顾）
      - $Entropy(S) = -\sum_{i=1}^{n} P_i \times log_2 P_i$
    - 用熵来度量样例的均一性（纯度）
      - 样例纯度越高，某一个事件发生的概率越大，熵值越低
      - $Entropy(S) = -p_{\oplus} log_2 p_{\oplus} - p_{\ominus} log_2 p_{\ominus}$

文本分类方法

- 文本分类方法—决策
  - 属性的选择
    - 熵的定义（回顾）
      - $Entropy(S) = -\sum_{i=}^{n}$
    - 用熵来度量样例的均一性（纯度）
      - 样例纯度越高，某一个事件发生的概率越大，熵值越低
      - $Entropy(S) = -p_{\oplus} log_2 p_{\oplus} - p_{\ominus} log_2 p_{\ominus}$
    - 举例
      - $Entropy(9_{\oplus}, 5_{\ominus}) = -\left(\frac{9}{14}\right) log_2 \left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) log_2 \left(\frac{5}{14}\right) = 0.940$

Outlook	Temperature	Humidity	Wind	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

文本分类方法

- 文本分类方法—决策树
  - 属性的选择
    - 信息增益
      - 一个属性的信息增益就是由于使用这个属性分割样例而导致的期望熵的降低
      - $Gain(S,A) = Entropy(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$
        - S，样例（$[9_{\oplus}, 5_{\ominus}]$）
        - A，选择的属性
        - v，属性值

文本分类方法

Outlook	Temperature	Humidity	Wind	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

- 文本分类方法—决策
  - 属性的选择
    - 信息增益举例
      - 属性
        - *Wind*
      - 属性值
        - *Weak (false)*， *Strong (true)*
      - S= [9 $_{\oplus}$, 5 $_{\ominus}$] ， S_{false}=[6 $_{\oplus}$, 2 $_{\ominus}$]， S_{true}=[3 $_{\oplus}$, 3 $_{\ominus}$]
      - $Gain(S, Wind) = Entropy(S) - \sum_{v \in \{false,true\}} \frac{|S_v|}{|S|} Entropy(S_v)$

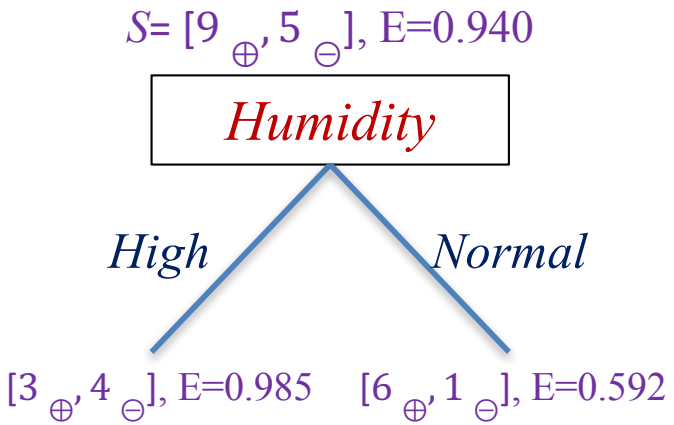
$$= Entropy(S) - \left(\frac{8}{14}\right) Entropy(S_{false}) - \left(\frac{6}{14}\right) Entropy(S_{true})$$

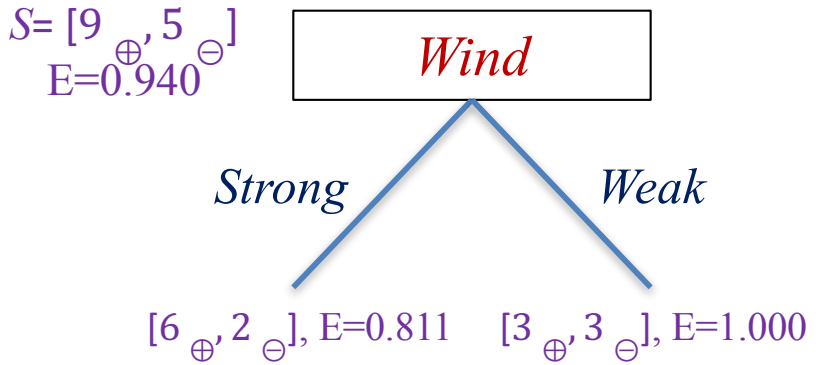
$$= 0.940 - \left(\frac{8}{14}\right) * 0.811 - \left(\frac{6}{14}\right) * 1.00 = 0.048$$

文本分类方法

- 文本分类方法—决策
  - 属性的选择
    - 确定最佳分类的属性

Outlook	Temperature	Humidity	Wind	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N





$$Gain(S, Humidity) = 0.940 - \left(\frac{7}{14}\right) * 0.985 - \left(\frac{7}{14}\right) * 0.592$$

$$Gain(S, Wind) = 0.940 - \left(\frac{8}{14}\right) * 0.811 - \left(\frac{6}{14}\right) * 1.000$$

文本分类方法

- 文本分类方法—决策树
  - 属性的选择
    - 不同属性得到的信息增益值
      - Gain(*S,Outlook*)=0.246
      - Gain(*S,Humidity*)=0.151
      - Gain(*S,Wind*)=0.048
      - Gain(*S,Temperature*)=0.029
    - Outlook的信息增益值最大

文本分类方法

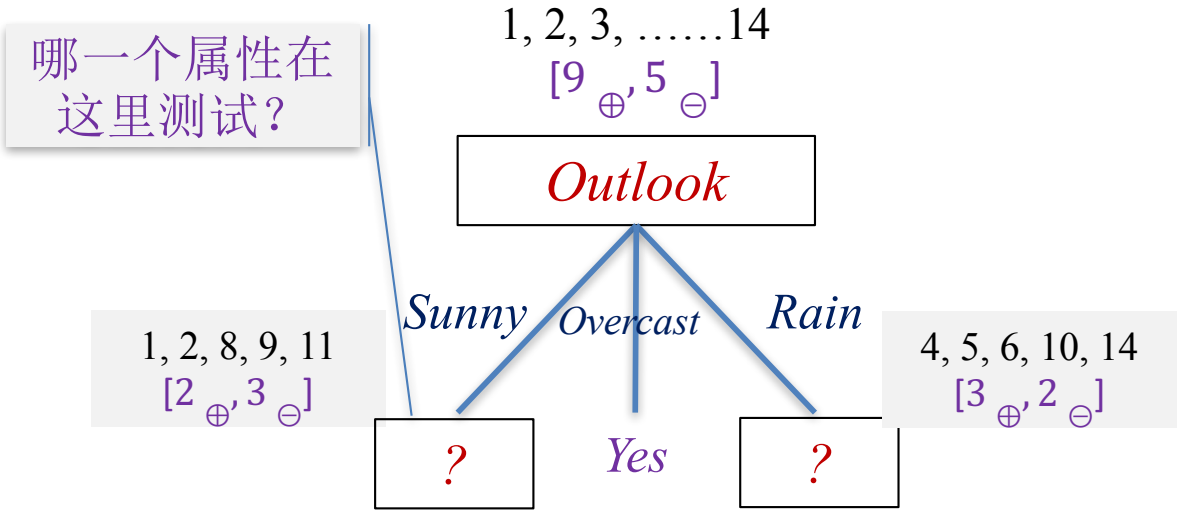
- 文本分类方法—决策
  - $Entropy\ (S_{sunny}) = 0.970$

Outlook	Temperature	Humidity	Wind	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

$Gain(S_{sunny}, Humidity)$
0.970-(3/5)*0.0-(2/5)*0.0=0.970

$Gain(S_{sunny}, Wind)$
0.970-(2/5)*1.0-(3/5)*0.918=0.019

$Gain(S_{sunny}, Temperature)$
0.970-(2/5)*0.0-(2/5)*1.0-(1/5)*0.0=0.570



文本分类方法

● 文本分类方法—决策树

输入：训练数据集D，属性集A，阈值ε;
输出：决策树T.

① 若D中所有实例属于同一类 ，则T为单结点树，并将类C_k作为该节点的类标记，返回T；

② 若$A=\emptyset$，则T为单结点树，并将D中实例数最大的类C_k作为该节点的类标记，返回T；

③ 否则，计算A中各属性对D的信息增益，选择信息增益最大的属性A_k；

④ 如果A_g的信息增益小于阈值ε，则T为单节点树，并将D中实例数最大的类C_k作为该节点的类标记，返回T；

⑤ 否则，对A_g的每一种可能值a_i，依$A_g = a_i$将D分割为若干非空子集D_i，将D_i中实例数最大的类作为标记，构建子结点，由结点及其子树构成树T，返回T；

⑥ 对第i个子节点，以D_i为训练集，以$A - A_g$为属性集合，递归调用1~5，得到子树T_i，返回。

文本分类方法

- 文本分类方法—决策树
  - 剪枝 (Pruning)
    - 决策树学习算法对付“过拟合”的主要手段
      - 决策树学习中，为了尽可能正确分类训练样本，节点划分过程不断重复，有时会造成决策树分支过多
      - 这就可能因训练样本学得“太好”了，以至于把训练样本自身的一些特点当作所有数据都具有的一般性质而导致过拟合

文本分类方法

- 文本分类方法—决策树
  - 剪枝 (Pruning)
    - 预剪枝
      - 在决策树生成的过程中，对每个节点在划分前先进行估计
      - 若当前节点的划分不能带来决策树泛化性能的提升，则停止划分并将当前节点标记为叶节点
    - 后剪枝
      - 先从训练集生成一棵完整的决策树，然后自底向上地对非叶节点进行考察
      - 若将该节点对应的子树替换为叶节点能带来决策树泛化性能提升，则将该子树替换为叶节点

文本分类方法

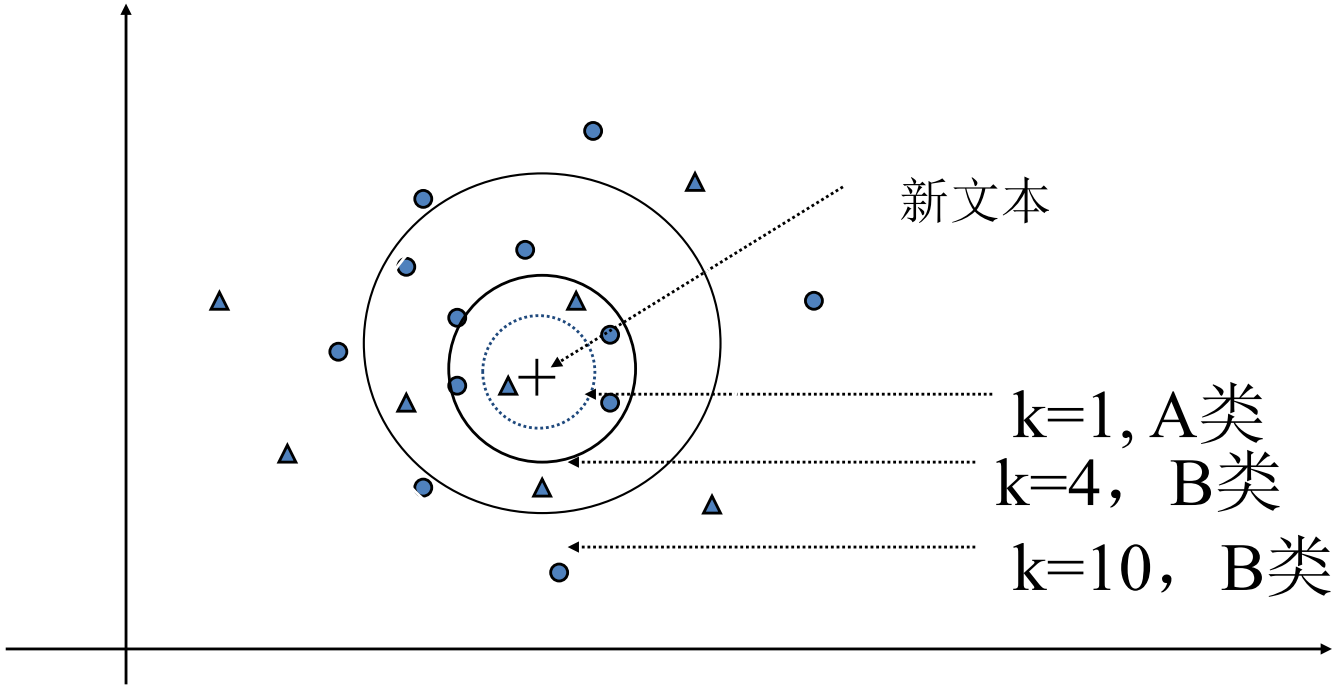
- 文本分类方法—决策树
  - 决策树方法的优点
    - 可读性强
      - 如果给定一个模型，那么根据所产生的决策树很容易推理出相应的逻辑表达
    - 分类速度快
      - 能在相对短的时间内能够对大型数据源做出可行且效果良好的结果
  - 决策树方法的缺点：
    - 对未知的测试数据未必有好的分类、泛化能力
      - 即可能发生过拟合现象，此时可采用剪枝

文本分类方法

- 文本分类方法—*KNN*（K-Nearest Neighbor）
  - *KNN*分类规则
    - 对于测试样本点y，在集合中距离它最近的点（文档）x。***KNN***分类就是把y分为x所属的类别
    - 在训练集X中，寻找和y最相似的训练样本x
      - $sim_{MAX}(y) = MAX_{x \in X} sim(x, y)$
    - 得到k个最相似的文档集合A，A为X的一个子集
      - $A = \{x \in X | sim(x, y) = sim_{MAX}(y)\}$
    - 设n_1，n_2分别为集合中属于类别c_1，c_2的文档个数
      - $p(c_1|y) = n_1 / (n_1+n_2)$，$p(c_2|y) = n_2 / (n_1+n_2)$
      - 如果$p(c_1|y) > p(c_2|y)$，则判定为c_1，否则为c_2

文本分类方法

- 文本分类方法—*KNN*（K-Nearest Neighbor）



带权重计算，计算权重和最大的类。k常取3或者5

文本分类方法

- 文本分类方法—*KNN*（K-Nearest Neighbor）
  - *KNN*方法依赖于相似度矩阵（或距离）
    - 对连续m维空间最简单的方法采用欧氏距离
      - 两个n维向量a $(x_{11}, x_{12}, \ldots\ldots, x_{1n})$ 与b $(x_{21}, x_{22}, \ldots\ldots, x_{2n})$
      - $d = \sqrt{\sum_{k=1}^{n}(x_{1k} - x_{2k})^2}$
    - 对m维二值实例空间最简单的方法是海明距离
      - 两个向量不同的分量所占的百分比
    - 对于基于文本*tf/idf*权重向量的余弦相似度、*Jaccard*系数是经常被采用的

文本分类方法

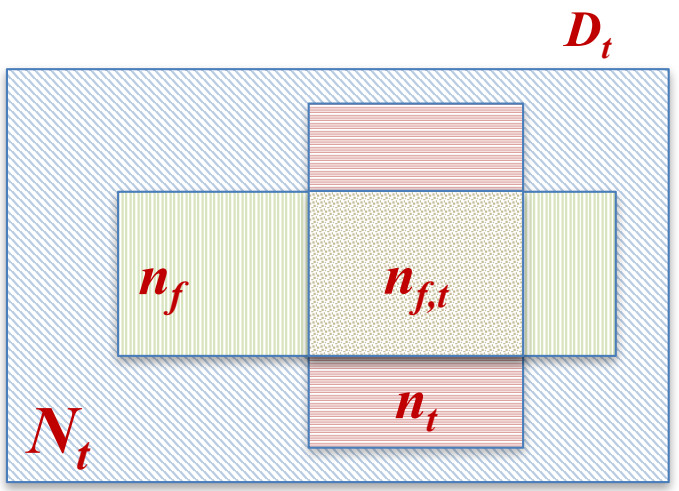
- 文本分类方法—*KNN*（K-Nearest Neighbor）
  - *KNN*算法的优点
    - 训练时间复杂度低
    - 与朴素贝叶斯之类的算法相比，对数据没有假设
    - 实现起来简单
  - *KNN*算法的缺点
    - 计算量大，测试的时候速度比较慢
    - 样本不均衡时，对稀有类别的预测准确率低

主要内容

- 聚类方法
- 特征选择方法
- 文本分类方法
- 文本分类的评价

文本分类的评价

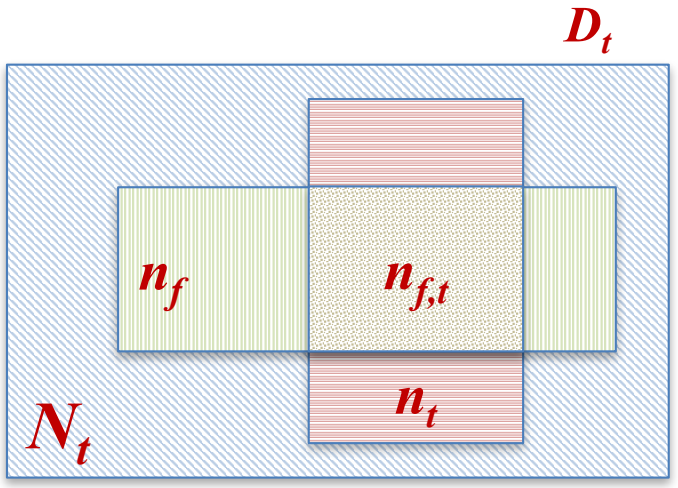
- 联列表（Contingency Table）



  - D：文档集合
  - D_t：已标注的文档集合
  - N_t：D_t中文本的数量
  - $C = \{c_1, c_2, \ldots, c_L\}$：类别集合
  - $T : D_t \times C \rightarrow [0, 1]$：$D_t$的分类函数（实际值）
  - n_t：D_t中属于类别c_p的文档数量
  - $F : D \times C \rightarrow [0, 1]$：文本分类函数
  - n_f：D_t中通过文本分类函数划分到类别c_p的文档数量

文本分类的评价

- 联列表（Contingency Table）



	$T(d_j,c_p)=1$（实际Y）	$T(d_j,c_p)=0$（实际N）	总计
$F(d_j,c_p)=1$（判定Y）	$n_{f,t}$	$n_f - n_{f,t}$	n_f
$F(d_j,c_p)=0$（判定N）	$n_t - n_{f,t}$	$N_t - n_f - n_t + n_{f,t}$	$N_t - n_f$
所有文档	n_t	$N_t - n_t$	N_t

文本分类的评价

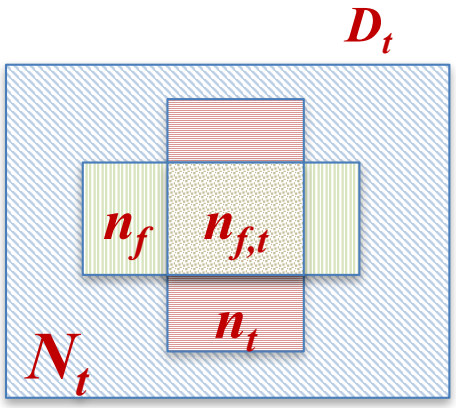
- 精确率、误差

精确率：$Acc(c_p) = \frac{n_{f,t} + (N_t - n_f - n_t + n_{f,t})}{N_t}$

误　差：$Err(c_p) = \frac{(n_f - n_{f,t}) + (n_t - n_{f,t})}{N_t}$

$$Acc(c_p) + Err(c_p) = 1$$

	$T(d_j,c_p)=1$（实际Y）	$T(d_j,c_p)=0$（实际N）	总计
$F(d_j,c_p)=1$（判定Y）	$n_{f,t}$	$n_f - n_{f,t}$	n_f
$F(d_j,c_p)=0$（判定N）	$n_t - n_{f,t}$	$N_t - n_f - n_t + n_{f,t}$	$N_t - n_f$
所有文档	n_t	$N_t - n_t$	N_t



文本分类的评价

*CASE2*比*CASE1*好
*CASE2*比*CASE1*好*1%*，两个分类器性能看起来差不多，但事实不是这样

- 精确率、误差

CASE1	$T(d_j,c_p)=1$（实际Y）	$T(d_j,c_p)=0$（实际N）	总计
$F(d_j,c_p)=1$（判定Y）	*0*	*0*	*0*
$F(d_j,c_p)=0$（判定N）	*20*	*980*	*1000*
所有文档	*20*	*980*	*1000*

$$Acc(c_p)=\frac{0+980}{1000}=98\%$$

$$Err(c_p)=\frac{0+20}{1000}=2\%$$

CASE2	$T(d_j,c_p)=1$（实际Y）	$T(d_j,c_p)=0$（实际N）	总计
$F(d_j,c_p)=1$（判定Y）	*10*	*0*	*10*
$F(d_j,c_p)=0$（判定N）	*10*	*980*	*990*
所有文档	*20*	*980*	*1000*

$$Acc(c_p)=\frac{10+980}{1000}=99\%$$

$$Err(c_p)=\frac{0+10}{1000}=1\%$$

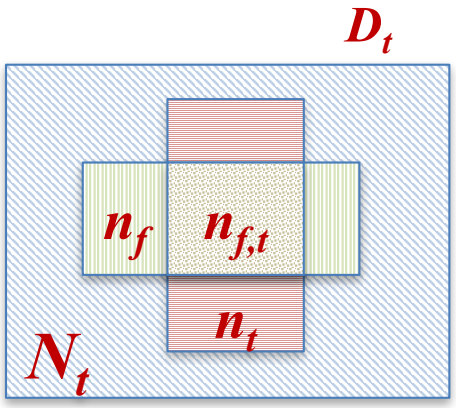
文本分类的评价

准确率：$P(c_p) = \frac{n_{f,t}}{n_f}$

召回率：$R(c_p) = \frac{n_{f,t}}{n_t}$

- 准确率和召回率
  - 准确率
    - 系统判定的属于c_p类中的文档中判断正确的文档数目所占的比例
  - 召回率
    - 系统判定的属于c_p类中的文档数目占正确文档总数的比例

	$T(d_j,c_p)=1$（实际Y）	$T(d_j,c_p)=0$（实际N）	总计
$F(d_j,c_p)=1$（判定Y）	$n_{f,t}$	$n_f - n_{f,t}$	n_f
$F(d_j,c_p)=0$（判定N）	$n_t - n_{f,t}$	$N_t - n_f - n_t + n_{f,t}$	$N_t - n_f$
所有文档	n_t	$N_t - n_t$	N_t



文本分类的评价

- 准确率和召回率

CASE	$T(d_j,c_p)=1$（实际Y）	$T(d_j,c_p)=0$（实际N）	总计
$F(d_j,c_p)=1$（判定Y）	*10*	*0*	*10*
$F(d_j,c_p)=0$（判定N）	*10*	*980*	*990*
所有文档	*20*	*980*	*1000*

$$P(c_p) = \frac{10}{10} = 100\%$$

$$R(c_p) = \frac{10}{20} = 50\%$$

文本分类的评价

- *F*测度
  - 准确率和召回率
    - 需要对文档集合中的每个类别都要计算
    - 如果类别数较多的时候，需要进行大量的计算
  - *F*测度：将准确率和召回率进行综合考虑

$$F\text{测度：}\ F_{\alpha}(c_p)=\frac{(\alpha^2+1)P(c_p)R(cp)}{\alpha^2P(c_p)+R(cp)}$$

  - α：调节准确率和召回率的重要度
    - $\alpha=0$，只考虑准确率
    - $\alpha=\infty$，只考虑召回率
    - $\alpha=1$，为*F1*值

$$F1\text{值：}\ F_{1}(c_p)=\frac{2P(c_p)R(cp)}{P(c_p)+R(cp)}$$

文本分类的评价

- *F*测度示例

CASE	$T(d_j,c_p)=1$（实际Y）	$T(d_j,c_p)=0$（实际N）	总计
$F(d_j,c_p)=1$（判定Y）	10	0	10
$F(d_j,c_p)=0$（判定N）	10	980	990
所有文档	20	980	1000

$$P(c_p)=\frac{10}{10}=100\%$$

$$R(c_p)=\frac{10}{20}=50\%$$

$$F_1(c_p)=\frac{2*1*0.5}{1+0.5}=67\%$$

文本分类的评价

- *Macro F1* 和 *Micro F1*
  - 前面的到的*F*测度是针对单个类别的
  - 为了评价算法在整个数据集上的性能
    - 宏平均：*Macro F1*
      - 对每一个类的性能指标的算术平均值

$$MacroF_1 = \frac{\sum_{p=1}^{|C|} F_1(c_p)}{|C|}$$

    - 微平均：*Micro F1*
      - 对每一个实例（文档）的性能指标的算术平均值

$$MicroF_1 = \frac{2PR}{P+R}$$

$$P = \frac{\sum_{c_p \in C} n_{f,t}}{\sum_{c_p \in C} n_f}$$

$$R = \frac{\sum_{c_p \in C} n_{f,t}}{\sum_{c_p \in C} n_t}$$

文本分类的评价

- 交叉验证（Cross-Validation）
  - 用来验证分类器性能的一种统计分析方法
    - 构建k个分类器：Ψ_1，Ψ_2，..... Ψ_k
      - 将训练集D_t分为k个大小不相交的集合（folds）
        - N_{t1}，N_{t2}，…，N_{tk}
      - 对于分类器Ψ_i
        - 在D_t中除了N_{ti}的集合上对算法进行训练、调优
        - 在集合N_{ti}上完成完成算法的测试
    - 每个分类器独立使用准确率、召回率或$F1$值评价
    - 将每个分类器上的结果进行平均作为k-*fold*交叉验证的结果（k通常取10）

本章小结

- 掌握分类、聚类技术在信息检索中的应用
- 掌握基本的特征选择方法的原理、思想、优缺点
- 掌握基本的分类、聚类方法
- 掌握分类技术的评价方法