信息检索与数据挖掘

第10章 文本分类

part1：文本分类及朴素贝叶斯方法

part2：基于向量空间的文本分类

part3：支持向量机及机器学习方法

回顾：概率检索模型

- **概率检索模型是通过概率的方法将查询和文档联系起来**
  - 定义3个随机变量R、Q、D：相关度$R=\{0,1\}$，查询$Q=\{q_1,q_2,\ldots\}$，文档$D=\{d_1,d_2,\ldots\}$，则可以通过计算条件概率$P(R=1|Q=q,D=d)$来度量文档和查询的相关度。
- **概率模型包括一系列模型，如Logistic Regression(回归)模型及最经典的二值独立概率模型BIM、BM25模型等等(还有贝叶斯网络模型)。基于统计语言建模的IR模型本质上也是概率模型的一种。**

文档和查询表示为词项的集合 →相关度为布尔运算结果
文档和查询表示为向量（词项对应不同的维度） →相关度为向量的余弦相似度
文档和查询表示为随机变量 →相关度为随机变量(二值或非二值)

回顾：概率排序原理

PRP（probability ranking principle）

- 利用概率模型来**估计每篇文档和需求的相关概率** P(R=1|d,q)，然后对结果进行次序。
- **怎么求 P(R=1|d,q)？**
  - 由乘法公式: P(R,d,q)=P(q)·P(R|q)·P(d|R,q)
  - P(R|q)： P(R=1|q)和P(R=0|q)可根据不相关文档百分比估计
  - P(R,d,q)的估计转化为估计P(d|R,q)

乘法公式
全概率公式
贝叶斯公式

给定d和q时候d和q相关的概率 →给定q时d为相关文档的概率

- **直接求P(d|R,q)仍然很困难**

优势率： $O(A)=\frac{P(A)}{P(\bar{A})}=\frac{P(A)}{1-P(A)}$

$O(R|\vec{x},\vec{q})=\frac{P(R=1|\vec{x},\vec{q})}{P(R=0|\vec{x},\vec{q})}$

回顾：二值独立概率模型 (Binary Independence Model，简称BIM)

- “二值”：文档和查询都表示为词项出现与否的布尔向量。文档d表示为向量 $\vec{x}=(x_1,\ldots,x_M)$，其中当词项t 出现在文档d 中时，$x_t=1$，否则$x_t=0$。
- “独立性”：词项在文档中的出现是互相独立的

Bayes公式

对于给定查询是个常数

$$O(R\mid\vec{x},\vec{q})=\frac{P(R=1\mid\vec{x},\vec{q})}{P(R=0\mid\vec{x},\vec{q})}=\frac{\dfrac{P(R=1\mid\vec{q})P(\vec{x}\mid R=1,\vec{q})}{P(\vec{x}\mid\vec{q})}}{\dfrac{P(R=0\mid\vec{q})P(\vec{x}\mid R=0,\vec{q})}{P(\vec{x}\mid\vec{q})}}=\frac{P(R=1\mid\vec{q})}{P(R=0\mid\vec{q})}\frac{P(\vec{x}\mid R=1,\vec{q})}{P(\vec{x}\mid R=0,\vec{q})}$$

$P(\vec{x}\mid R=1,\vec{q})$ 和 $P(\vec{x}\mid R=0,\vec{q})$ 分别表示当返回一篇相关或不相关文档时文档表示为 $\vec{x}$ 的概率

$P(R=1\mid\vec{q})$ 和 $P(R=0\mid\vec{q})$ 分别表示对于查询 $\vec{q}$ 返回一篇相关和不相关文档的先验概率。

回顾： BIM排序函数的推导

$$O(R|\vec{x},\vec{q}) = \frac{P(R=1|\vec{x},\vec{q})}{P(R=0|\vec{x},\vec{q})} = \frac{\frac{P(R=1|\vec{q})P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|\vec{q})}}{\frac{P(R=0|\vec{q})P(\vec{x}|R=0,\vec{q})}{P(\vec{x}|\vec{q})}} = \frac{P(R=1|\vec{q})}{P(R=0|\vec{q})} \cdot \frac{P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|R=0,\vec{q})}$$

- 利用“二值性”， x_t 取值要么为0要么为1
- 忽略常数项
- 只考虑出现在文档中的查询词项
- p_t 词项出现在一篇相关文档中的概率
- u_t 词项出现在一篇不相关文档中的概率

$$O(R|\vec{x},\vec{q}) = O(R|\vec{q}) \cdot \prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} \cdot \prod_{t:q_t=1} \frac{1-p_t}{1-u_t}$$

回顾：

RSV（retrieval status value，检索状态值）

排序函数只需计算 $\prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)}$

最终用于排序的是

$$RSV_d = \log \prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} = \sum_{t:x_t=q_t=1} \log \frac{p_t(1-u_t)}{u_t(1-p_t)}$$

c_t 查询词项的优势率比率（odds ratio）的对数值

定义 $c_t = \log \frac{p_t(1-u_t)}{u_t(1-p_t)} = \log \frac{p_t}{1-p_t} + \log \frac{1-u_t}{u_t}$

$$RSV_d = \sum_{x_t=q_t=1} c_t$$

	文档	相关（R=1）	不相关（R=0）
词项出现	x_t=1	p_t	u_t
词项不出现	x_t=0	$1-p_t$	$1-u_t$

c_t 如何计算？

回顾：
求c_t：理论上的概率估计方法

表中df_t 是包含t 的文档数目

	文档	相关	不相关	总计
词项出现	$x_t=1$	s	df_t-s	df_t
词项不出现	$x_t=0$	$S-s$	$(N-df_t)-(S-s)$	$N-df_t$
	总计	S	$N-S$	N

$$c_t = \log\frac{p_t(1-u_t)}{u_t(1-p_t)} = \log\frac{p_t}{1-p_t} + \log\frac{1-u_t}{u_t}$$

$p_t=s/S$ ， $u_t=(df_t-s)/(N-S)$

$$c_t = K(N, df_t, S, s) = \log\frac{s/(S-s)}{(df_t-s)/((N-df_t)-(S-s))}$$

回顾：
求c_t：实际中的概率估计方法

$$c_t = \log\frac{p_t(1-u_t)}{u_t(1-p_t)} = \log\frac{p_t}{1-p_t} + \log\frac{1-u_t}{u_t}$$

- **u_t的估算**
  - 假设相关文档只占所有文档的极小一部分，那么可通过整个文档集的统计数字来计算与不相关文档有关的量。

$u_t = df_t/N$ $\quad \log[(1-u_t)/u_t] = \log[(N-df_t)/df_t] \approx \log N/df_t$

逆文档频率 idf_t

- **估算p_t**
  - 如果我们知道某些相关文档，那么可以利用这些已知相关文档中的词项出现频率来对p_t进行估计
  - Croft 和Harper（1979）在组合匹配模型（combination match model）中提出了利用常数来估计pt 的方法。
  - Greiff （1998）提出 → $p_t = \frac{1}{3} + \frac{2}{3}\cdot\frac{df_t}{N}$

回顾：求c_t：利用相关反馈获取更精确的p_t估计

不断迭代估计过程来获得pt 的更精确的估计结果

- **(1) 给出p_t 和u_t 的初始估计。如，假设所有查询中的词项的p_t 是个常数，具体地可以取p_t=0.5。**
- **(2) 利用当前p_t 和u_t 的估值对相关文档集合$R = \{d : R_{d,q} = 1\}$进行最佳的猜测。用该模型返回候选相关文档集给用户。**
- **(3) 利用用户交互对上述模型进行修正，这是通过用户对某个文档子集V 的相关性判断来实现的。基于相关性判断结果，V 可以划分成两个子集：$VR = \{d \in V, R_{d,q} = 1\}$ R 和$VNR = \{d \in V, R_{d,q} = 0\}$，后者与R 没有交集。**
- **(4) 利用已知的相关文档和不相关文档对p_t 和u_t 进行重新估计。如果VR 和VNR 足够大的话，可以直接通过集合中的文档数目来进行最大似然估计：$p_t = |VR_t|/|VR|$。**

VR_t 是VR 中包含词项xt 的文档子集

实际中往往要对上述估计进行平滑

回顾： BIM→BM25

Okapi BM25: 一个非二值模型

- **BIM 模型不考虑词项频率和文档长度，简单。** $RSV(Q,D)=\sum_{t_i\in D\cup Q} W_i^{IDF}$
- **BM25考虑词项在文档中的tf权重，有：**

$$RSV(Q,D)=\sum_{t_i\in D\cup Q} W_i^{IDF}\frac{(k_1+1)tf_{t_i,D}}{k_1((1-b)+b\times(L_D/L_{ave}))+tf_{t_i,D}}$$

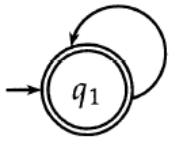
  - $tf_{ti,D}$: 词项ti在文档D中的词项频率
  - L_D (L_{ave}): 文档D的长度(整个文档集的平均长度)
  - k_1: 用于控制文档中词项频率比重的调节参数
  - b: 用于控制文档长度比重的调节参数
- **如果查询比较长，则加入查询的tf**

$$RSV(Q,D)=\sum_{t_i\in D\cup Q} W_i^{IDF}\cdot\frac{(k_1+1)tf_{ti,D}}{k_1((1-b)+b\times(L_D/L_{ave}))+tf_{ti,D}}\cdot\frac{(k_3+1)tf_{t_i,Q}}{k_3+tf_{t_i,Q}}$$

问题1：语言模型

什么是语言模型(language model, LM)

$$P(\text{frog said that toad likes frog}) = (0.01 \times 0.03 \times 0.04 \times 0.01 \times 0.02 \times 0.01) \times (0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.2) \approx 0.000\,000\,000\,001\,573$$



$P(\text{STOP}|q_1) = 0.2$

the	0.2
a	0.1
frog	0.01
toad	0.01
said	0.03
likes	0.02
that	0.04
...	...

- 比较两个LM，可计算似然比（likelihood ratio），我们忽略停止概率
- 若语言模型与文档是一一映射的关系，那么查询与文档的相关性可以转化为查询与语言模型的相关性

问题1：语言模型

怎样由文档生成语言模型？

- 一元语言模型（**unigram language model**）：

$$P_{uni}(t_1t_2t_3t_4) = P(t_1)P(t_2)P(t_3)P(t_4)$$

- 问题：已知样本D，求其模型M_D的参数$P(w|M_D)$。

$$P(d) = \frac{L_d!}{tf_{t_1,d}!tf_{t_2,d}!\cdots tf_{t_M,d}!} P(t_1)^{tf_{t_1,d}} P(t_2)^{tf_{t_2,d}} \cdots P(t_M)^{tf_{t_M,d}}$$

$$\vec{\theta}_D = (\theta_1, \theta_2, ..., \theta_L)$$
$$= (P(w_1|M_D), P(w_2|M_D), ..., P(w_L|M_D))$$

M_D的参数求解

$$\vec{\theta}_D^{\,*} = \arg\max_{\vec{\theta}_D} P(D|\vec{\theta}_D)$$

arg max

- **In mathematics, arg max stands for the argument of the maximum, that is to say, the set of points of the given argument for which the given function attains its maximum value:**

$$\arg\max_x f(x) := \{x \mid \forall y : f(y) \leq f(x)\}$$

$$\arg\max_x f(x) \qquad \arg\min_x f(x)$$

问题1：语言模型

语言模型的种类：n-gram

- 一元语言模型（**unigram language model**）,也称上下文无关语言模型

⇩ $P_{uni}(t_1t_2t_3t_4) = P(t_1)P(t_2)P(t_3)P(t_4)$

- 二元语言模型（**bigram language model**），即计算条件概率时只考虑前一个词项的出现情况：

⇩ $P_{bi}(t_1t_2t_3t_4) = P(t_1)P(t_2|t_1)P(t_3|t_2)P(t_4|t_3)$ 一阶马尔科夫链

- 三元语言模型（ **trigram language model** ）
- …
- **Unigram→ bigram→…→n-gram**

还有其他更复杂的种类，本课程不介绍

问题2：语言模型如何应用到IR中？

$P(d|q) \rightarrow P(q|M_d)$

- **IR中使用LM的问题**
  - N个文档，各自有一个语言模型，给定一个查询，求查询与哪个文档相关度最高？

    一种可能的思路：把相关度看成是每篇文档对应的语言模型下生成该查询的可能性

- **总体分布&抽样**
  - 文档的模型(风格)实际上是某种总体分布
  - 文档和查询都是该总体分布下的一个抽样样本实例
  - 根据文档，估计文档的模型，即求出该总体分布(一般假设某种总体分布，然后求出其参数)
  - 然后计算**该总体分布下抽样出查询的概率**

问题2：语言模型如何应用到IR中？

查询似然模型

- **查询似然模型（query likelihood model, QLM）**
  - 目标：将文档按照其与查询相关的似然P(d|q) 排序
  - 实现目标的途径：按照P(q|d)进行排序
- **具体的方法是：**
  - (1) 对每篇文档推导出其LM
  - (2) 估计查询在每个文档d_i 的LM 下的生成概率$P(q|M_{di})$

$$P(q \mid M_d) = K_q \prod_{t \in V} P(t \mid M_d)^{tf_{t,d}}$$

$$\hat{P}(q \mid M_d) = \prod_{t \in q} \hat{P}_{\text{mle}}(t \mid M_d) = \prod_{t \in q} \frac{tf_{t,d}}{L_d}$$

  - (3) 按照上述概率对文档进行排序

问题2：语言模型如何应用到IR中？

线性插值LM（linear interpolation LM）

- 如果$tf_{t,d}=0$，那么有 $\hat{P}(t|M_d) \le cf_t / T$
- 其中，cf_t 是t 在整个文档集的出现次数，T 是所有文档集中词条的个数。一个实际效果较好的简单方法是，将基于文档的多项式分布和基于全部文档集估计出的多项式分布相混合，即

$$\hat{P}(t|d) = \lambda \hat{P}_{\text{mle}}(t|M_d) + (1-\lambda)\hat{P}_{\text{mle}}(t|M_c)$$

- 其中，$0 < \lambda < 1$，Mc是基于全部文档集构造的LM。上述公式混合了来自单个文档的概率词在整个文档集的出现频率。该模型中如何设置正确的λ是获得良好性能的关键。

关于LM的讨论：LM vs. BIM

- **LM与BIM质的区别在哪里？**
  - 语言建模的方法给出了一个全新的看待文本检索的视角。正如Ponte 和 Croft (1998)强调的那样，IR 中的LM 方法提供了另外一种计算查询与文档匹配得分的方式，其带来的希望在于，概率语言建模能够优化已有的**权重**计算方法从而升高检索的性能。
  - 语言建模IR 中最主要的问题是**文档模型的估计**，包括如何选择有效的**平滑方法**。基于语言建模的IR 模型已经获得了非常好的检索结果，与其他概率模型的方法相比（如BIM 模型），最主要的区别似乎是语言模型的方法**不再对相关性进行显式建模**（而在BIM 模型中，相关性作为一个最主要的变量来建模）。

关于LM的讨论：LM与相关反馈

- **为什么LM与相关反馈很难结合？**
  - 对基于语言建模的IR 模型人们也有很多不同意见。文档和信息需求表示之间的对象同类假设显然不符合实际情况。当前的LM 方法采用了非常简单的语言模型，比如常常采用一元模型。
  - 由于**没有定义一个显式的相关性概念**，相关反馈技术和用户偏好信息很难集成到模型中。另外，将一元模型扩展到更复杂的、能够容纳短语或段落匹配或布尔操作符的模型看上去也很有必要。一些后续的有关LM 的工作已经对上述关注点进行了研究，包括将相关反馈放回到模型中，并且允许查询和文档的语言不匹配。

关于LM的讨论：LM vs. tf-idf

- **基于LM的IR与基于tf-idf的IR的关联性？**
  - 在tf-idf 模型中，词项频率会直接进行表示。
  - 将文档生成概率和文档集生成概率二者相混合的效果有点像idf，那些在整个文档集中出现较少而在某些文档中出现较多的词项将对文档的最后排序起到较大作用。
  - 统计建模IR 模型和tf-idf 模型相比，在直觉上一个来自概率论而另一个来自几何学，在数学模型上一个更具理论性而另一个更基于启发式知识，在诸如词项频率和文档长度等统计因素的使用细节上也有区别。

$$\hat{P}(q \mid M_d) = \prod_{t \in q} \hat{P}_{\text{mle}}(t \mid M_d) = \prod_{t \in q} \frac{tf_{t,d}}{L_d}$$

$$\hat{P}(t \mid d) = \lambda \hat{P}_{\text{mle}}(t \mid M_d) + (1-\lambda)\hat{P}_{\text{mle}}(t \mid M_c)$$

关于相关性的讨论：表示相关性

- **文档和查询表示决定了相关性的表示**
  - 表示为词项的集合→相关度为布尔运算结果
  - 表示为向量→相关度为向量的余弦相似度
  - 表示为随机变量→相关度为随机变量(二值或非二值)

- **灵活多样的概率表示**
  - 相关的概率：$P(R=1|Q=q,D=d)$ →$P(D=d|R=1,Q=q)$
  - 查询生成的概率：$P(Q=q|D=d)$ → $P(Q=q|M=M_d)$
  - 文档生成的概率：$P(D=d|Q=q)$ → $P(D=d|M=M_q)$

$O(R|Q=q,D=d)$

$$R(d;q) = LK(M_d \| M_q) = \sum_{t \in V} P(t|M_q) \log \frac{P(t|M_q)}{P(t|M_d)}$$

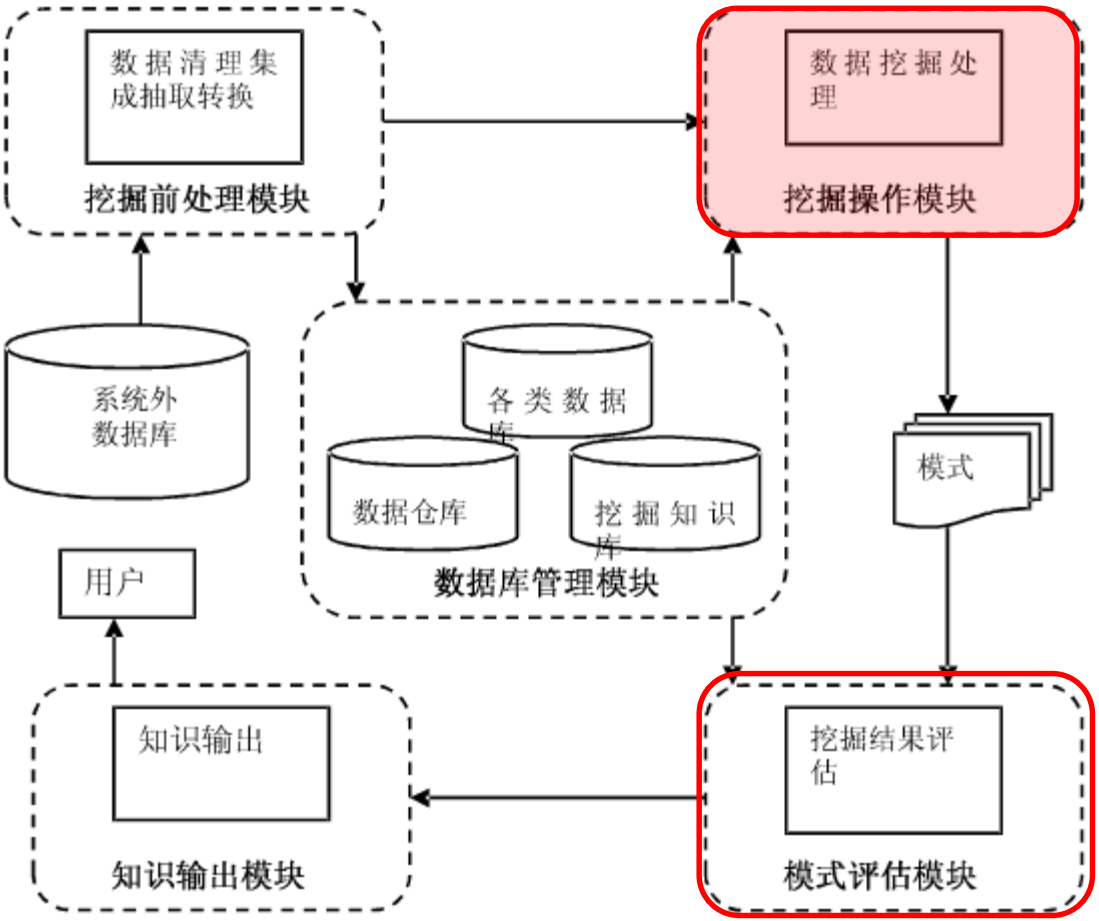
课程内容

- 第1章 绪论
- 第2章 布尔检索及倒排索引
- 第3章 词项词典和倒排记录表
- 第4章 索引构建和索引压缩
- 第5章 向量模型及检索系统
- 第6章 检索的评价
- 第7章 相关反馈和查询扩展
- 第8章 概率模型
- 第9章 基于语言建模的检索模型
- 第10章 文本分类
  - 文本分类及朴素贝叶斯方法
  - 基于向量空间的文本分类
  - 支持向量机及机器学习方法
- 第11章 文本聚类
- 第12章 Web搜索
- 第13章 多媒体信息检索
- 第14章 其他应用简介

Information Retrieval(IR)：从大规模非结构化数据（通常是文本）的集合（通常保存在计算机上）中找出满足用户信息需求的资料（通常是文档）的过程

数据挖掘（Data Mining）从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中，提取隐含在其中的、人们事先不知道的、但又是潜在有用的信息和知识的过程

引言：数据挖掘系统



数据挖掘系统的体系结构图

本讲内容：文本分类及朴素贝叶斯方法

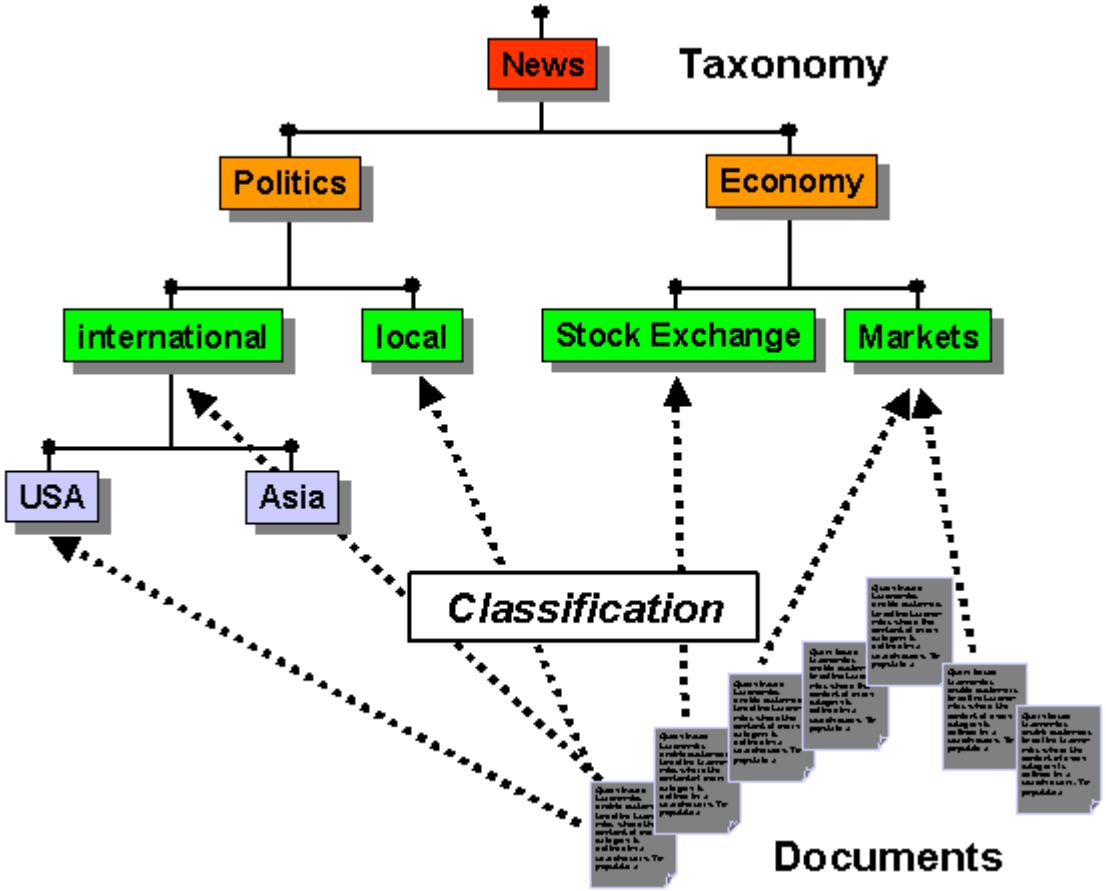
- 什么是文本分类？

- 什么是朴素贝叶斯分类器？
- 朴素贝叶斯分类器的生成模型
- 朴素贝叶斯分类器的的性质

- 提高分类器效率的方法：特征选择
- 文本分类的评价

Taxonomies and Classification

Note that documents can be assigned to more than one category



A taxonomy depicts the hierarchical ordering of categories. Taxonomies allow you to structure a large number of documents that belong to a document set clearly. The classification procedure assigns documents to the categories according to topic.

文本分类的定义

- **Text classification或者 Text Categorization**
  - 给定分类体系（ taxonomy），将一篇文本分到其中一个或者多个类别中的过程。

- **文本分类中，给定文档$d \in X$和一个固定的类别集合$C= \{c_1, c_2, \ldots, c_J\}$，其中X表示文档空间 （document space），类别（class）也通常称为类（category） 或类标签（ label）。**
  - 按类别数目： binary vs. multi-class
  - 按每篇文档赋予的标签数目： sing label vs. multi label

分类方法: 1. 手工方法

- **Web发展的初期，Yahoo使用人工分类方法来组织Yahoo目录，类似工作还有： ODP, PubMed**
- **如果是专家来分类精度会非常高**
- **如果问题规模和分类团队规模都很小的时候，能否保持分类结果的一致性**
- **但是对人工分类进行规模扩展将十分困难，代价昂贵**

- **→ 因此，需要自动分类方法**

分类方法: 2. 规则方法

只要进入Google 快讯主页，输入您的搜索字词、您要的搜索结果类型（新闻，网页或新闻与网页及论坛）、希望我们检查搜索结果的频率，以及您的电子邮件地址。然后，单击“创建快讯”按钮。我们将向您发送确认电子邮件。在您单击确认电子邮件中的链接后，快讯即可启动您还可以通过访问我们的“管理快讯”页面一次完成快讯的创建和确认。

分类方法: 2. 规则方法

- 规则：如含有“多媒体”的书籍归入“TP37”

http://ztflh.jourserv.com/

中图分类号查询 > 工业技术 > 自动化技术、计算机技术 > 计算技术、计算机技术 > 多媒体技术与多媒体计算机

检索词： 检索

TP37 多媒体技术与多媒体计算机

- 对于p234提到的multicore computer chips 的例子，一个可能的规则是(multicore OR multi-core) AND (chip OR processor OR microprocessor)。
- 有时规则即等价于布尔表达式。
- 如果规则经过专家长时间的精心调优，精度会非常高
- 建立和维护基于规则的分类系统非常繁琐，开销也大

分类方法: 3.机器学习的方法

- **机器学习**
  - 除了手工分类和人工编写规则之外，还存在第3 种文本分类的方法，即基于机器学习的方法。我们主要关注这种方法。在机器学习中，**规则集（更通用的说法是分类决策准则）是从训练数据中自动学习得到的。**

后面将介绍一系列分类方法: 朴素贝叶斯, Rocchio, kNN, SVM

- **统计文本分类**
  - 当学习方法基于统计时，这种方法也称为统计文本分类（statistical text classification）。在统计文本分类中，对于每个类别我们需要一些好的文档样例（或者称为训练文档）。由于需要人来标注训练文档，所以对人工分类的需求仍然存在。这里的标注（labeling）指的是对每篇文档赋予类别标签的过程。

基于学习的文本分类

- **文档空间X**
  - 文档都在该空间下表示（通常都是某种高维空间）
- **固定的类别集合C= {$c_1, c_2, \ldots, c_J$ }**
  - 类别往往根据应用的需求来认为定义
- **训练集 D,文档d的类别用c标记，$\langle d, c\rangle \in X \times C$**
  - 利用学习算法，根据给定的$\langle d, c\rangle$可以学习一个分类器 Υ ，它可以将文档映射成类别： $\Upsilon : X \rightarrow C$
- **文档分类的实现**
  - 对于文档空间中文档， $d \in X$ ， 可确定$\Upsilon(d) \in C$, 即确定d最可能属于的类别$c_i = \Upsilon(d)$， $c_i \in C$

文本分类

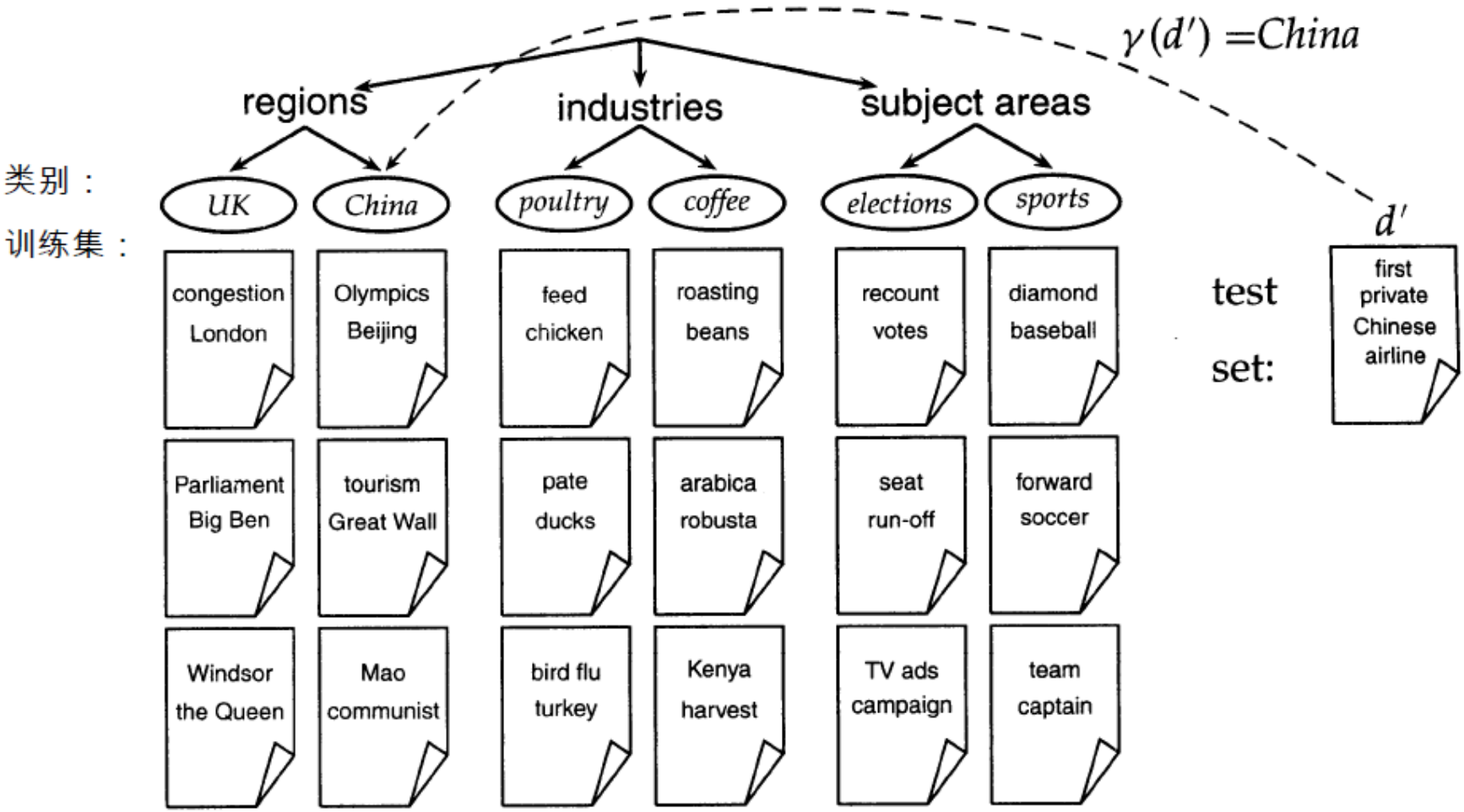
- **给定训练集**
  - <d, c> = <Beijing joins the World Trade Organization, China>
  - 表示的是单句文档Beijing joins the World Trade Organization 被标记为China 类。

- 利用某种学习方法（learning method）或学习算法（learning algorithm），我们希望学到某个分类函数（classification function）γ，它可以将文档映射到类别
  - $\Upsilon : X \rightarrow C$

- 判断文档d'最可能属于的类别$c_i = \Upsilon(d')$， $c_i \in C$

文本分类中的类别、训练集及测试集

Classes, **training** set, and **test** set in text classification

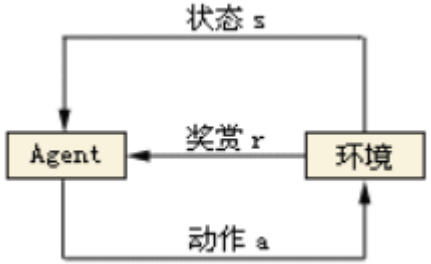


无监督/有监督的学习

- **supervised learning 监督学习**
  - 利用一组已知类别的样本调整分类器的参数，使其达到所要求性能的过程，也称为监督训练或有教师学习。正如人们通过已知病例学习诊断技术那样，计算机要通过学习才能具有识别各种事物和现象的能力。用来进行学习的材料就是与被识别对象属于同类的有限数量样本。监督学习中在给予计算机学习样本的同时，还告诉计算各个样本所属的类别。
- **无监督学习**
  - 若所给的学习样本不带有类别信息,就是无监督学习。

增强学习/强化学习

- 强化学习(reinforcement learning，又称再励学习，评价学习)是一种重要的机器学习方法。但在传统的机器学习分类中没有提到过强化学习，而在连接主义学习中，把学习算法分为三种类型，即非监督学习(unsupervised learning)、监督学习(supervised leaning)和强化学习。



- 强化学习是从动物学习、参数扰动自适应控制等理论发展而来，其基本原理是：如果Agent的某个行为策略导致环境正的奖赏(强化信号)，那么Agent以后产生这个行为策略的趋势便会加强。Agent的目标是在每个离散状态发现最优策略以使期望的折扣奖赏和最大。

IR中的文本分类应用

- 语言识别 (类别: **English vs. French**等)
- 垃圾网页的识别 (垃圾网页 **vs.** 正常网页)
- 是否包含淫秽内容 (色情 **vs.** 非色情)
- 领域搜索或垂直搜索 – 搜索对象限制在某个垂直领域（如健康医疗） (属于该领域 **vs.** 不属于该领域)
- 静态查询 (如，**Google Alerts**)
- 情感识别: 影评或产品评论是贬还是褒 (褒评 **vs.** 贬评)

小结：什么是文本分类

- **Taxonomies and Classification**
- **文本分类中，给定文档d ∈ X和一个固定的类别集合C= {c_1, c_2, . . . , c_J }，其中X表示文档空间 （document space），类别（class）也通常称为类（category） 或类标签（ label）。**
- **分类方法**
  - 手工方法→规则方法→基于学习的文本分类
- **文本分类中的类别、训练集及测试集**
- **无监督/有监督的学习**
- **IR中的文本分类应用**

本讲内容：文本分类及朴素贝叶斯方法

- 什么是文本分类？
- 什么是朴素贝叶斯分类器？
- 朴素贝叶斯分类器的生成模型
- 朴素贝叶斯分类器的的性质
- 提高分类器效率的方法：特征选择
- 文本分类的评价

朴素贝叶斯分类器
Naive Bayes text classification

独立性的假设

- 朴素贝叶斯是一个概率分类器
- 文档 *d* 属于类别 *c* 的概率计算如下：

$$P(c|d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k|c)$$

Bayes公式：P(c|d) → P(d|c)

- $<t_1, t_2, \ldots, t_{nd}>$是d 中的词条，它们是分类所用词汇表的一部分，$n_d$ 是文档的长度(词条的个数)
- $P(t_k|c)$ 是词项t_k 出现在类别c中文档的概率
- $P(c)$ 是类别c的先验概率
- 如果文档的词项无法提供属于哪个类别的信息，那么我们直接选择$P(c)$最高的那个类别

39

具有最大后验概率的类别

- 在文本分类中，我们的目标是找出文档最可能属于的类别。对于NB 分类来说，最可能的类是具有MAP（maximum a posteriori，最大后验概率）估计值的结果c_{map}：

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k|c)$$

- 由于我们不知道参数的真实值，所以上述公式中采用了从训练集中得到的估计值^P 来代替P。为避免浮点数下界溢出，可引入对数：

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} [\log \hat{P}(c) + \sum_{1 \le k \le n_d} \log \hat{P}(t_k|c)].$$

如何估计参数P^(c)及P^(t_k|c)？

- **MLE估计**

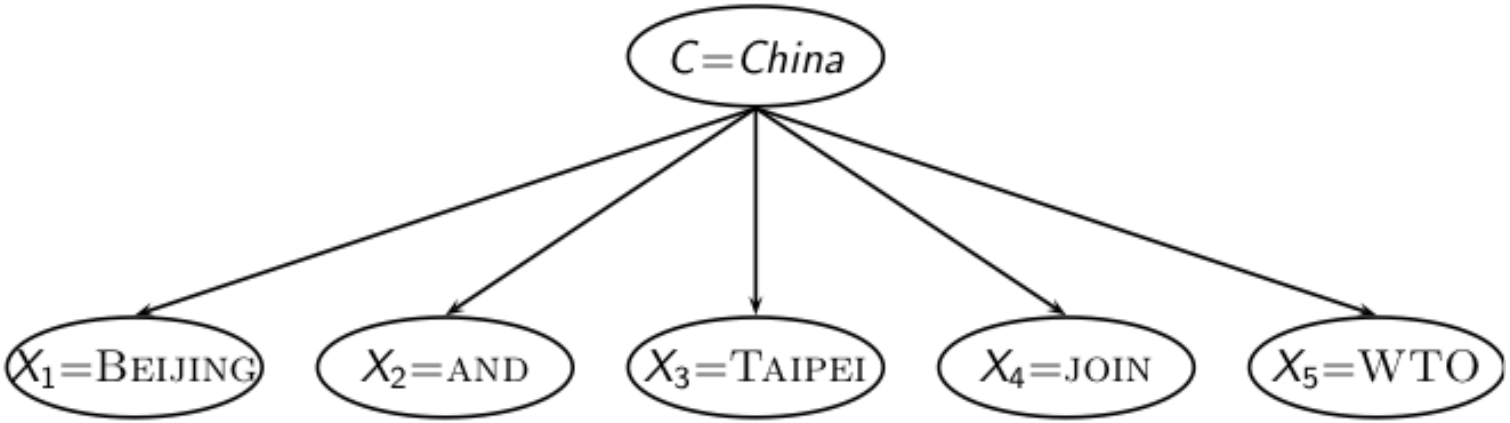
$$c_{map} = \arg\max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k|c)$$

$$\hat{P}(c) = \frac{N_c}{N} \qquad \hat{P}(t \mid c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

  - N_c 是训练集合中c 类包含的文档数目，N 是训练集合中的文档总数
  - T_{ct} 是t 在训练集合c 类文档中出现的次数，在对每篇文档计算时用的是其在文档中多次出现的词频。

- **位置独立性假设**
  - 引入了位置独立性假设（positional independence assumption），在该假设下，T_{ct} 是t 在训练集某类文档中所有位置k 上的出现次数之和。这样，对于不同位置上的概率值都采用相同的估计办法，比如，如果某词在一篇文档中出现过两次，分别在k1 和k2 的位置上，那么我们假定^P(t_{k1}|c)=^P(t_{k2}|c)

MLE估计中的问题：零概率问题



$P(China|d) \propto P(China) \cdot P(\texttt{BEIJING}|China) \cdot P(\texttt{AND}|China) \cdot P(\texttt{TAIPEI}|China) \cdot P(\texttt{JOIN}|China) \cdot P(\text{WTO}|China)$

- 如果 WTO 在训练集中没有出现在类别 China中:

$$\hat{P}(\text{WTO}|China) = \frac{T_{China,\text{WTO}}}{\sum_{t' \in V} T_{China,t'}} = \frac{0}{\sum_{t' \in V} T_{China,t'}} = 0$$

一旦发生零概率，将无法判断类别

44

避免零概率: 加一平滑

- 平滑前：

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

- 平滑后: 对每个量都加上1

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V}(T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B}$$

- 其中，B = |V| 是词汇表中所有词项的数目。加一平滑可以认为是采用均匀分布作为先验分布（每个词项在每个类中出现一次）然后根据训练数据进行更新得到的结果。

45

朴素贝叶斯：训练过程

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k|c)$$

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(t \mid c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

TRAINMULTINOMIALNB($\mathbb{C}, \mathbb{D}$)

1 $V \leftarrow$ EXTRACTVOCABULARY($\mathbb{D}$)
2 $N \leftarrow$ COUNTDOCS($\mathbb{D}$)
3 **for each** $c \in \mathbb{C}$
4 **do** $N_c \leftarrow$ COUNTDOCSINCLASS($\mathbb{D}, c$)
5 $prior[c] \leftarrow N_c/N$
6 $text_c \leftarrow$ CONCATENATETEXTOFALLDOCSINCLASS($\mathbb{D}, c$)
7 **for each** $t \in V$
8 **do** $T_{ct} \leftarrow$ COUNTTOKENSOFTERM($text_c, t$)
9 **for each** $t \in V$
10 **do** $condprob[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$
11 **return** $V, prior, condprob$

运算量：

计算|c| 个 ^P(c)

计算|c| • |V|个 $\hat{P}(t_k|c)$

朴素贝叶斯：测试过程

- 训练过程已得到了估计参数P^(c)及P^(t_k |c)

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} [\log \hat{P}(c) + \sum_{1 \le k \le n_d} \log \hat{P}(t_k|c)].$$

- 测试过程根据P^(c)及P^(t_k |c)计算文档d的 c_{map}

ApplyMultinomialNB($\mathbb{C}$, V, *prior*, *condprob*, d)
1 $W \leftarrow$ ExtractTokensFromDoc(V, d)
2 **for each** $c \in \mathbb{C}$
3 **do** *score*[c] $\leftarrow$ log *prior*[c]
4 **for each** $t \in W$
5 **do** *score*[c] += log *condprob*[t][c]
6 **return** arg max$_{c \in \mathbb{C}}$ *score*[c]

运算量：
计算|C| 个 ^P(c)
计算|C| • |V|个 P^(t_k |c)

朴素贝叶斯分

$$\hat{P}(t|c) = \frac{T_{ct}+1}{\sum_{t' \in V}(T_{ct'}+1)} = \frac{T_{ct}+1}{(\sum_{t' \in V} T_{ct'}) + B}$$

表13-1 用于参数估计的数据

	文档ID	文档中的词	属于c=China类?
训练集	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
测试集	5	Chinese Chinese Chinese Tokyo Japan	?

p241

- 估计朴素贝叶斯分类器的参数，并对测试文档进行分类

Priors: $\hat{P}(c) = 3/4$ and $\hat{P}(\overline{c}) = 1/4$ Conditional probabilities:

$$\begin{aligned}
\hat{P}(\text{CHINESE}|c) &= (5+1)/(8+6) = 6/14 = 3/7 \\
\hat{P}(\text{TOKYO}|c) = \hat{P}(\text{JAPAN}|c) &= (0+1)/(8+6) = 1/14 \\
\hat{P}(\text{CHINESE}|\overline{c}) &= (1+1)/(3+6) = 2/9 \\
\hat{P}(\text{TOKYO}|\overline{c}) = \hat{P}(\text{JAPAN}|\overline{c}) &= (1+1)/(3+6) = 2/9
\end{aligned}$$

Why?

$$\begin{aligned}
\hat{P}(c|d_5) &\propto 3/4 \cdot (3/7)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003 \\
\hat{P}(\overline{c}|d_5) &\propto 1/4 \cdot (2/9)^3 \cdot 2/9 \cdot 2/9 \approx 0.0001
\end{aligned}$$

朴素贝叶斯的时间复杂度分析

mode	time complexity
training	$\Theta(\|\mathbb{D}\|L_{ave} + \|\mathbb{C}\|\|V\|)$
testing	$\Theta(L_a + \|\mathbb{C}\|M_a) = \Theta(\|\mathbb{C}\|M_a)$

- L_{ave}: 训练文档的平均长度, L_a: 测试文档的平均长度, M_a: 测试文档中不同的词项个数, $\mathbb{D}$: 训练文档个数， V : 词汇表， $\mathbb{C}$: 类别集合 $\Theta(|\mathbb{D}|L_{ave})$
- $\Theta(|\mathbb{C}||V|)$ 是计算所有数字的时间
- $|\mathbb{C}||V| < |\mathbb{D}|L_{ave}$ 是从上述数字计算参数的时间
- 通常来说:

> What is the time complexity of NB? The complexity of computing the parameters is $\Theta(|\mathbb{C}||V|)$ because the set of parameters consists of $|\mathbb{C}||V|$ conditional probabilities and $|\mathbb{C}|$ priors.

- 测试时间也是线性的 (相对于测试文档的长度而言).
- 因此： 朴素贝叶斯 对于训练集的大小和测试文档的大小而言是线性的。这是最优的

49

NB与多项式LM的关系

- 上述NB 模型形式上等价于多项式一元LM

$$P(c|d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k|c)$$

$$P(d \mid q) \propto P(d) \prod_{t \in q} P(t \mid M_d)$$

- 这种NB分类器使用的是基于多项式的方法
- 稍后我们还介绍另外一种建立NB分类器的方法

小结：Naive Bayes text classification

- 在文本分类中，我们的目标是找出文档最可能属于的类别。对于NB 分类来说，最可能的类是具有MAP估计值的结果c_{map}：

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k|c)$$

- 如何估计参数P^(c)及P^(t_k |c)？

$$\hat{P}(c) = \frac{N_c}{N} \qquad \hat{P}(t \mid c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

- 零概率问题→平滑

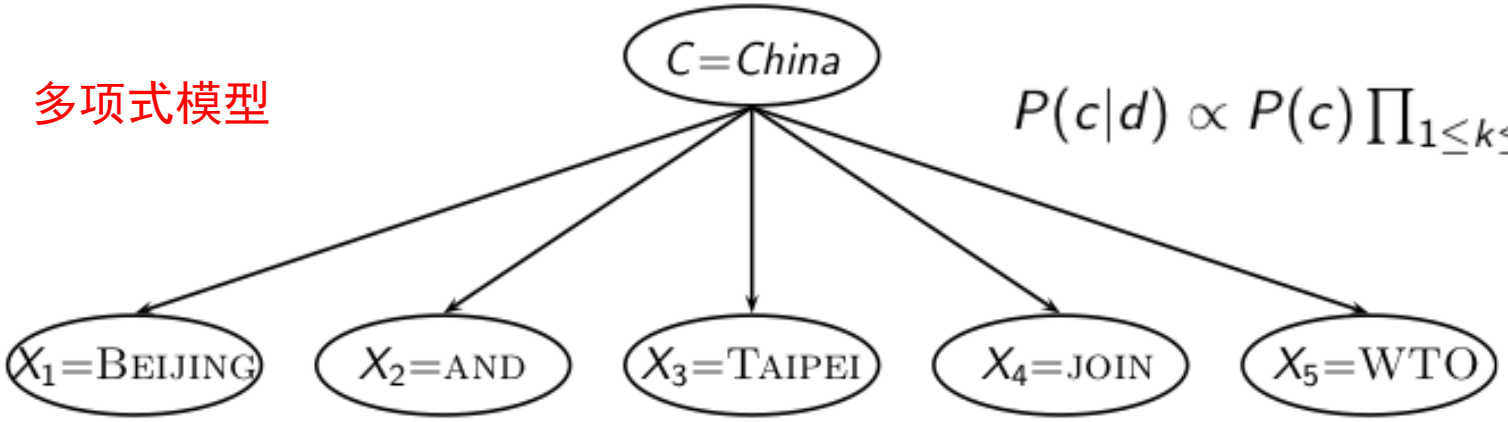
$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B}$$

本讲内容：文本分类及朴素贝叶斯方法

- 什么是文本分类？
- 什么是朴素贝叶斯分类器？
- 朴素贝叶斯分类器的生成模型
- 朴素贝叶斯分类器的的性质
- 提高分类器效率的方法：特征选择
- 文本分类的评价

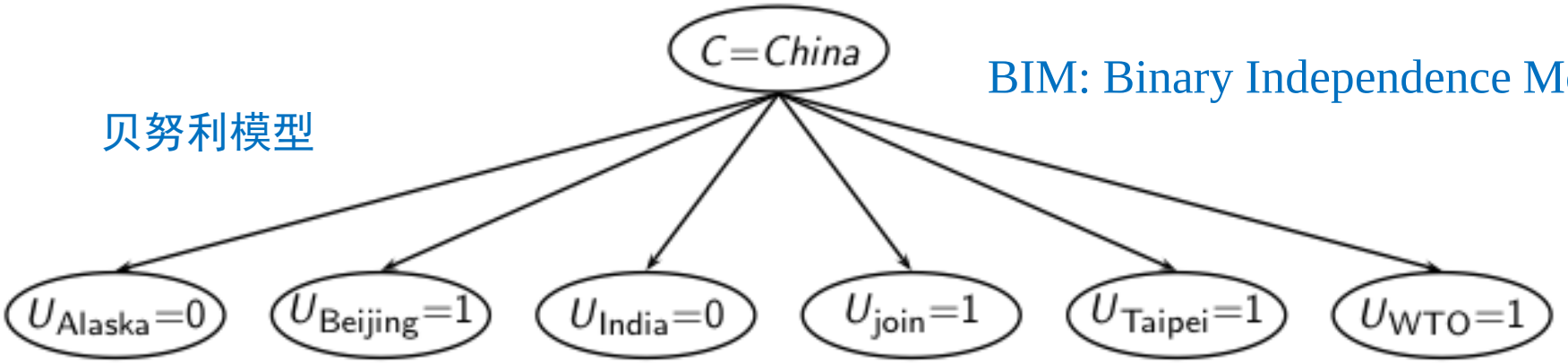
NB分类器的生成(Generative)模型

多项式模型



$$P(c|d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k|c)$$

LM: Language Model



贝努利模型

BIM: Binary Independence Model

Naive Bayes algorithm

^ P (*t* | *c*)的估计策略不同

未出现词项在分类中的使用不同

TRAINMULTINOMIALNB(C, D)
1 $V \leftarrow$ EXTRACTVOCABULARY(D)
2 $N \leftarrow$ COUNTDOCS(D)
3 **for each** $c \in C$
4 **do** $N_c \leftarrow$ COUNTDOCSINCLASS(D, c)
5 $prior[c] \leftarrow N_c/N$
6 $text_c \leftarrow$ CONCATENATETEXTOFALLDOCSINCLASS(D, c)
7 **for each** $t \in V$
8 **do** $T_{ct} \leftarrow$ COUNTTOKENSOFTERM($text_c, t$)
9 **for each** $t \in V$
10 **do** $condprob[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$
11 **return** $V, prior, condprob$

APPLYMULTINOMIALNB(C, V, $prior$, $condprob$, d)
1 $W \leftarrow$ EXTRACTTOKENSFROMDOC(V, d)
2 **for each** $c \in C$
3 **do** $score[c] \leftarrow \log prior[c]$
4 **for each** $t \in W$
5 **do** $score[c] \mathrel{+}= \log condprob[t][c]$
6 **return** $\arg\max_{c \in C} score[c]$

multinomial model

TRAINBERNOULLINB(C, D)
1 $V \leftarrow$ EXTRACTVOCABULARY(D)
2 $N \leftarrow$ COUNTDOCS(D)
3 **for each** $c \in C$
4 **do** $N_c \leftarrow$ COUNTDOCSINCLASS(D, c)
5 $prior[c] \leftarrow N_c/N$
6 **for each** $t \in V$
7 **do** $N_{ct} \leftarrow$ COUNTDOCSINCLASSCONTAININGTERM(D, c, t)
8 $condprob[t][c] \leftarrow (N_{ct}+1)/(N_c+2)$
9 **return** V, $prior$, $condprob$

APPLYBERNOULLINB(C, V, $prior$, $condprob$, d)
1 $V_d \leftarrow$ EXTRACTTERMSFROMDOC(V, d)
2 **for each** $c \in C$
3 **do** $score[c] \leftarrow \log prior[c]$
4 **for each** $t \in V$
5 **do if** $t \in V_d$
6 **then** $score[c] \mathrel{+}= \log condprob[t][c]$
7 **else** $score[c] \mathrel{+}= \log(1 - condprob[t][c])$
8 **return** $\arg\max_{c \in C} score[c]$

Bernoulli model

基于贝努利模型的NB示例：参数的计算（$\hat{P}(c)$和$\hat{P}(t|c)$的估计）

表13-1 用于参数估计的数据

	文档ID	文档中的词	属于c=China类?
训练集	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
测试集	5	Chinese Chinese Chinese Tokyo Japan	?

Priors: $\hat{P}(c) = 3/4$ and $\hat{P}(\overline{c}) = 1/4$ Conditional probabilities

$$\hat{P}(\text{Chinese}|c) = (3+1)/(3+2) = 4/5$$

$$\hat{P}(\text{Japan}|c) = \hat{P}(\text{Tokyo}|c) = (0+1)/(3+2) = 1/5$$

$$\hat{P}(\text{Beijing}|c) = \hat{P}(\text{Macao}|c) = \hat{P}(\text{Shanghai}|c) = (1+1)(3+2) = 2/5$$

$$\hat{P}(\text{Chinese}|\overline{c}) = (1+1)/(1+2) = 2/3$$

$$\hat{P}(\text{Japan}|\overline{c}) = \hat{P}(\text{Tokyo}|\overline{c}) = (1+1)/(1+2) = 2/3$$

$$\hat{P}(\text{Beijing}|\overline{c}) = \hat{P}(\text{Macao}|\overline{c}) = \hat{P}(\text{Shanghai}|\overline{c}) = (0+1)/(1+2) = 1/3$$

基于贝努利模型的NB示例：测试文档的分类结果

因此，测试文档分别属于两个类别的得分为

$$\hat{P}(c \mid d_5) \propto \hat{P}(c) \cdot \hat{P}(\text{Chinese} \mid c) \cdot \hat{P}(\text{Japan} \mid c) \cdot \hat{P}(\text{Tokyo} \mid c)$$
$$\cdot \left(1-\hat{P}(\text{Beijing} \mid c)\right) \cdot \left(1-\hat{P}(\text{Shanghai} \mid c)\right) \cdot \left(1-\hat{P}(\text{Macao} \mid c)\right)$$
$$= 3/4 \cdot 4/5 \cdot 1/5 \cdot 1/5 \cdot (1-2/5) \cdot (1-2/5) \cdot (1-2/5)$$
$$\approx 0.005$$

类似地，有

$$\hat{P}(\overline{c} \mid d_5) \propto 1/4 \cdot 2/3 \cdot 2/3 \cdot 2/3 \cdot (1-1/3) \cdot (1-1/3) \cdot (1-1/3)$$
$$\approx 0.022$$

因此，根据上述结果，分类器最终会将测试文档归为非 c 类。当只关注词项出现与否而不考虑词项频率时，Japan 和 Tokyo 对于 $\overline{c}$ 来说是正向标志特征（2/3 > 1/5），而 Chinese 属于 c 类和非 c 类的条件概率的差异还不足以影响分类的结果。

小结：朴素贝叶斯分类器的生成模型

- **文本分类的步骤**
  - 训练
  - 测试
- **建立 NB 分类器有两种不同的方法**
  - Multinomial NB model
  - Bernoulli model
- **Naive Bayes algorithm**
  - ˆ P (t | c)的估计策略不同
  - 未出现词项在分类中的使用不同

本讲内容：文本分类及朴素贝叶斯方法

- 什么是文本分类？
- 什么是朴素贝叶斯分类器？
- 朴素贝叶斯分类器的生成模型
- 朴素贝叶斯分类器的的性质
- 提高分类器效率的方法：特征选择
- 文本分类的评价

朴素贝叶斯规则

- 给定文档的条件下，我们希望得到最可能的类别

$$c_{map} = \arg\max_{c \in \mathbb{C}} P(c|d)$$

- 应用贝叶斯定律

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}:$$

$$c_{map} = \arg\max_{c \in \mathbb{C}} \frac{P(d|c)P(c)}{P(d)}$$

- 由于分母**P(*d*)**对所有类别都一样，因此可以去掉：

$$c_{map} = \arg\max_{c \in \mathbb{C}} P(d|c)P(c)$$

两种模型的文本生成过程

- 给定类别的时文档生成的条件概率计算有所不同：
- 多项式模型$P(d|c) = P(<t_1, \ldots, t_k, \ldots, t_{nd}>|c)$
- 贝努利模型$P(d|c) = P(<e_1, \ldots, e_i, \ldots, e_M>|c)$
- 其中，$< t_1, \ldots, t_{nd} >$是在d 中出现的词项序列（当然要去掉那些从词汇表中去掉的词，如停用词），$<e_1, \ldots, e_i, \ldots, e_M>$是一个M维的布尔向量，表示每个词项在文档d 中存在与否。
- $< t_1, \ldots, t_{nd} >$和$<e_1, \ldots, e_i, \ldots, e_M>$正好是两种不同的文档表示方法。第一种表示方法中，文档空间X是所有词项序列的集合；在第二种表示方法中，文档空间X是$\{0,1\}^M$。

两种生成模型需要估计的参数

- **多项式模型$P(d|c) = P(<t_1, \ldots, t_k, \ldots, t_{nd}>|c)$**
  - n_d 是文档的长度(词条的个数)
  - ˆ $P(c)$：$|C|$个
  - ˆ $P(t|c)$：$M^{nd} \bullet |C|$个
- **贝努利模型$P(d|c) = P(<e_1, \ldots, e_i, \ldots, e_M>|c)$**
  - M是词项的个数
  - ˆ $P(c)$：$|C|$个
  - ˆ $P(e|c)$：$2^M \bullet |C|$个不同的参数，每个参数都是M个e_i取值和一个类别取值的组合

多项式模型和贝努利模型具有相同数量级的参数个数。由于参数空间巨大，对这些参数进行可靠估计是不可行的。

朴素贝叶斯的条件独立性假设

- 为了减少参数的数目，我们引入了朴素贝叶斯的条件独立性假设（**conditional independence assumption**），即给定类别时，假设属性值之间是相互独立的：

$$\text{Multinomial} \quad P(d|c) = P(\langle t_1, \ldots, t_{n_d} \rangle | c) = \prod_{1 \le k \le n_d} P(X_k = t_k | c)$$

$$\text{Bernoulli} \quad P(d|c) = P(\langle e_1, \ldots, e_M \rangle | c) = \prod_{1 \le i \le M} P(U_i = e_i | c).$$

- 上面公式中引入了两类随机变量X_k 和U_i，这样的话两个不同的文本生成模型就更清晰。X_k 是文档在位置k 上的随机变量，$P(X_k = t \mid c)$表示的是一篇c 类文档中词项t 出现在位置k 上的概率。随机变量U_i 对应词项i，当词项在文档中不出现时取0，出现时取1。$P(U_i = 1|c)$ 表示的是t_i出现在c 类文档中的概率，这时可以是在任意位置上出现任意多次。

朴素贝叶斯的位置独立性假设

- 如果假设在不同位置**k** 上的词项分布不一样的话，那么就要估计针对每个**k** 的一系列参数。比如，**bean** 出现在**coffee** 类文档的第一个位置和出现在其第二个位置的概率是不同的，其他位置可以依次类推。这会再次导致数据估计中的稀疏性问题。故我们在多项式模型中引入第二个独立性假设—位置独立性假设（**positional independence**），即词项在文档中每个位置的出现概率是一样的，也就是对于任意位置k_1、k_2、词项t 和类别c，有

$$P(X_{k_1} = t \mid c) = P(X_{k_2} = t \mid c)$$

$M^{nd} \bullet |C|$

$2^M \bullet |C|$

基于条件独立性和位置独立性假设，我们只需要估计$\Theta(M \bullet |C|)$个多项式模型下的参数$P(t_k|c)$或贝努利模型下的参数$P(e_i|c)$，其中每个参数对应一个词项和类别的组合。

两个模型的比较

表13-3 多项式模型和贝努利模型的比较

	多项式模型	贝努利模型
事件模型	词条生成模型	文档生成模型
随机变量	$X=t$，当且仅当 t 出现在给定位置	$U_t=1$，当且仅当 t 出现在文档中
文档表示	$d=<t_1,\ldots,t_k,\ldots,t_{nd}>, t_k \in V$	$d=<e_1,\ldots,e_i,\ldots,e_M>, e_i \in \{0,1\}$
参数估计	$\hat{P}(X=t\mid c)$	$\hat{P}(U_i=e\mid c)$
决策规则：最大化	$\hat{P}(c)\prod_{1\le k\le n_d}\hat{P}(X=t_k\mid c)$	$\hat{P}(c)\prod_{t_i\in V}\hat{P}(U_i=e_i\mid c)$
词项多次出现	考虑	不考虑
文档长度	能处理更长文档	最好处理短文档
特征数目	能够处理更多特征	特征数目较少效果更好
词项the的估计	$\hat{P}(X=\text{the}\mid c)\approx 0.05$	$\hat{P}(U_{\text{the}}\mid c)\approx 1.0$

“朴素”

- 条件独立性假设声称在给定类别的情况下特征之间相互独立，这对于实际文档中的词项来说几乎不可能成立。多项式模型中还给出了位置独立性假设。而由于贝努利模型中只考虑词项出现或不出现，所以它忽略了所有的位置信息。
- 这种词袋模型忽略了自然语言句子中词序相关的信息，所以NB 对自然语言的建模做了非常大的简化，从这个意义上讲，如何能保证NB 方法的分类效果？

朴素贝叶斯方法起作用的原因

- 即使在条件独立性假设严重不成立的情况下，朴素贝叶斯方法能够高效地工作。例如

表13-4 正确的参数估计意味着精确的预测，但是精确的预测不一定意味着正确的参数估计

	c_1	c_2	选择的类别
真实概率$P(c \mid d)$	0.6	0.4	c_1
$\hat{P}(c)\prod_{1\le k\le n_d}\hat{P}(t_k \mid c)$（公式（13-13））	0.000 99	0.00 001	
NB估计 $\hat{P}(c \mid d)$	0.99	0.01	c_1

- 概率$P(c_2|d)$被过低估计(0.01)，而$P(c_1|d)$被过高估计 (0.99)。然而，分类决策取决于哪个类别得分最高，它并不关注得分本身的精确性。尽管概率估计效果很差，但是NB 会给c1 一个很高的分数，因此最后会将d 归到正确的类别中

分类的目标是预测正确的类别，并不是准确地估计概率

准确估计 ⇒ 精确预测， 反之并不成立！

小结：朴素贝叶斯分类器的的性质

- 多项式模型$P(d|c) = P(<t_1, \ldots, t_k, \ldots, t_{nd}>|c)$
- 贝努利模型$P(d|c) = P(<e_1, \ldots, e_i, \ldots, e_M>|c)$

- 朴素贝叶斯的条件独立性假设
- 朴素贝叶斯的位置独立性假设

- 准确估计概率 ⇒ 精确预测， 反之并不成立!

本讲内容：文本分类及朴素贝叶斯方法

- 什么是文本分类？

- 什么是朴素贝叶斯分类器？
- 朴素贝叶斯分类器的生成模型
- 朴素贝叶斯分类器的的性质

- 提高分类器效率的方法：特征选择
- 文本分类的评价

特征选择（feature selection）

- 特征选择是从训练集合出现的词项中选出一部分子集的过程。在文本分类过程也仅仅使用这个子集作为特征。特征选择有两个主要目的：第一，通过减少有效的词汇空间来提高分类器训练和应用的效率。这对于除NB 之外其他的训练开销较大的分类器来说尤为重要。第二，特征选择能够去除噪音特征，从而提高分类的精度。噪音特征（noise feature）指的是那些加入文本表示之后反而会增加新数据上的分类错误率的特征。假定某个罕见词项（如arachnocentric ）对某个类别（如China）不提供任何信息，但训练集中所有的arachnocentric恰好都出现在China 类，那么学习后产生的分类器会将包含arachnocentric的测试文档误分到China 类中去。这种由于训练集的偶然性导出的不正确的泛化结果称为过学习（overfitting）。

特征选择算法

- 给定类别c，对词汇表中的每个词项t，我们计算效用指标A(t,c)，然后从中选择k 个具有最高值的词项作为最后的特征，其他的词项则在分类中都被忽略。

```
SELECTFEATURES(𝔻, c, k)
1  V ← EXTRACTVOCABULARY(𝔻)
2  L ← []
3  for each t ∈ V
4  do A(t, c) ← COMPUTEFEATUREUTILITY(𝔻, t, c)
5     APPEND(L, ⟨A(t, c), t⟩)
6  return FEATURESWITHLARGESTVALUES(L, k)
```

图 13-6　选择 k 个最佳特征的基本特征选择算法

不同的效用指标

- **互信息 $A(t, c) = I(U_t; C_c)$**

$$I(U;C) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} P(U = e_t, C = e_c) \log_2 \frac{P(U = e_t, C = e_c)}{P(U = e_t) P(C = e_c)}$$

  - 其中，U 是一个二值随机变量，当文档包含词项t 时，它取值为e_t=1，否则取值为e_t=0。而C也是个二值随机变量，当文档属于类别c 时，它取值为e_c=1，否则取值为e_c=0。

- **$\chi 2$ 统计量 $A(t, c) = X2(t,c)$**

$$X^2(\mathrm{D}, t, c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}}$$

  - 字母N 表示的是D中观察到的频率，而E 则是期望频率

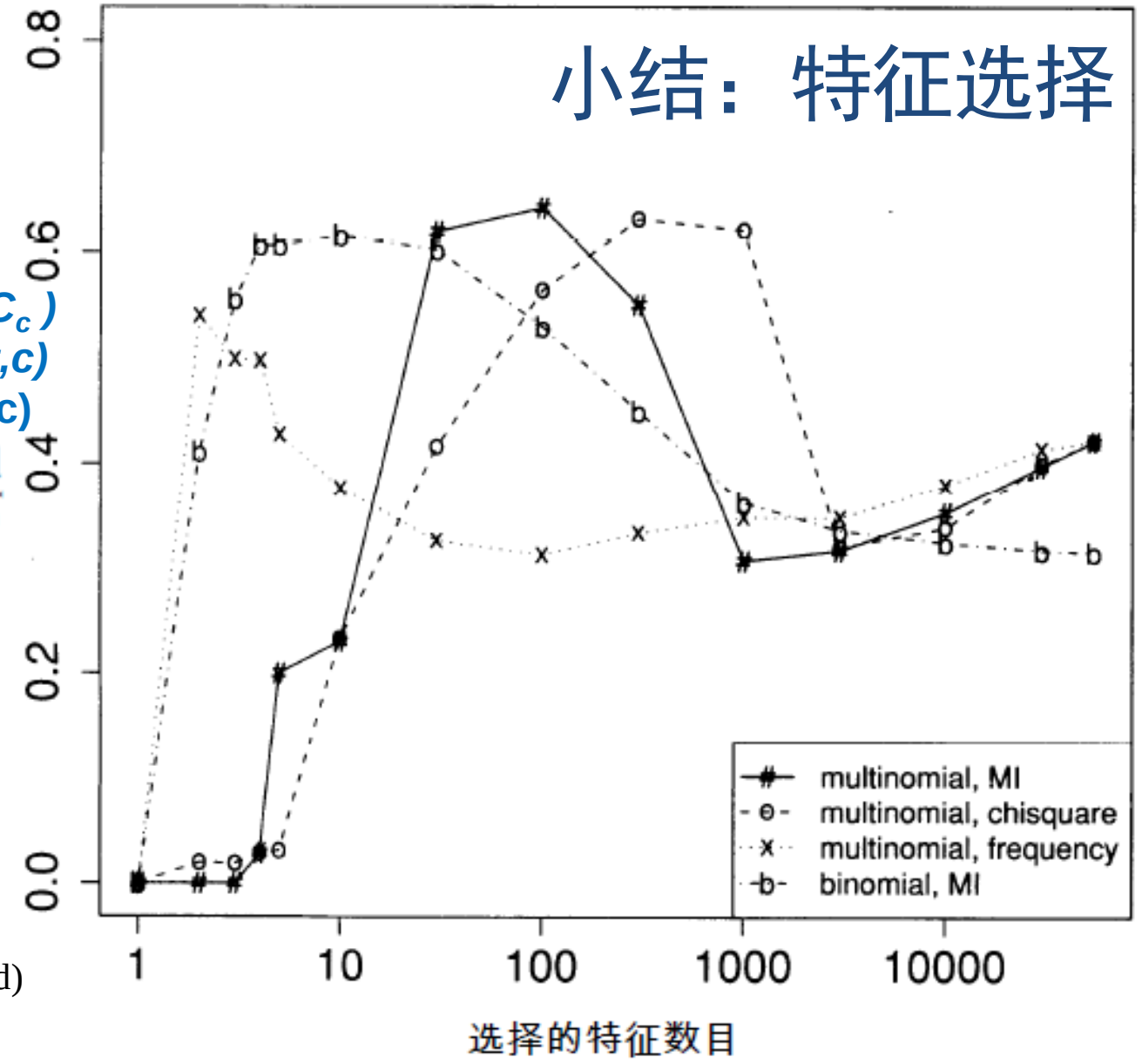
- **词项频率A(t, c) = N(t, c)**
  - 即选择那些在类别中频率较高的词项作为特征。频率可以定义为文档频率或文档集频率。

小结：特征选择

互信息 $A(t, c) = I(U_t; C_c)$

χ2 统计量 $A(t, c) = X^2 t,c)$

词项频率A(t, c) = N(t, c)



$$F_{\beta=1} = \frac{2PR}{P+R}$$

Precision=P(relevant|retrieved)

Recall =P(retrieved|relevant)

图 13-8 不同特征数目下多项式模型和贝努利模型的分类效果

本讲内容：文本分类及朴素贝叶斯方法

- 什么是文本分类？

- 什么是朴素贝叶斯分类器？
- 朴素贝叶斯分类器的生成模型
- 朴素贝叶斯分类器的的性质

- 提高分类器效率的方法：特征选择
- 文本分类的评价

文本分类的评价

- 分类的评价必须基于测试数据进行，而且该测试数据是与训练数据完全独立的 (通常两者样本之间无交集)。常用的指标: 正确率、召回率、 F1值、分类精确率等。
- 当对具有多个分类器的文档集进行处理时，往往需要计算出一个融合了每个分类器指标的综合指标。为实现这个目的，通常有宏平均和微平均两种做法： 其中宏平均（macroaveraging ） 是在类别之间求平均值， 而微平均（microaveraging）则是将每篇文档在每个类别上的判定放入一个缓冲池，然后基于这个缓冲池计算效果指标。

微平均 vs. 宏平均

- **宏平均(Macroaveraging)**
  - 对类别集合C 中的每个类都计算一个F_1 值
  - 对C个结果求平均Average these C numbers
- **微平均(Microaveraging)**
  - 对类别集合C 中的每个类都计算TP、FP和FN
  - 将C中的这些数字累加
  - 基于累加的TP, FP, FN计算P、R和F_1

表13-8 宏平均和微平均的计算

	类别1			类别2			缓冲表	
	实际 yes	实际 no		实际 yes	实际 no		实际 yes	实际 no
判定yes	10	10	判定 yes	90	10	判定yes	100	20
判定 no	10	970	判定 no	10	890	判定 no	20	1860

注：“实际”表示实际上属于该类；“判定”表示的是分类器的判定情况。下例中，宏平均正确率为[10/(10 + 10) + 90/(10 + 90)]/2 = (0.5 + 0.9)/2 = 0.7，而微平均正确率为 100/(100 + 20) ≈ 0.83。

宏平均和微平均的适用范围

- 宏平均和微平均的计算结果可能会相差很大。宏平均对每个类别同等对待，而微平均则对每篇文档的判定结果同等对待。
- 由于F1 值忽略判断正确的负例，所以它的大小主要由判断正确的正例数目所决定，所以在**微平均计算中大类起支配作用**。上例中，系统的微平均正确率(0.83)更接近c_2 类的正确率(0.9)，而与c_1 类的正确率(0.5)相差较大，这是因为c_2 的大小是c_1 的5 倍。因此，微平均实际上是文档集中大类上的一个效果度量指标。如果要**度量小类上的效果，往往需要计算宏平均指标**。

小结：文本分类的评价

- **文本分类的目标**
  - 使得测试数据上的分类错误率最小
- **常用的指标**
  - 正确率、召回率、 F1值、分类精确率等
- **多个分类器的文档集**
  - 当对具有多个分类器的文档集进行处理时，往往需要计算出一个融合了每个分类器指标的综合指标
- **宏平均和微平均**
  - 微平均计算中大类起支配作用
  - 度量小类上的效果，往往需要计算宏平均指标

本讲要点

- 什么是文本分类？**Taxonomies and Classification**
- 什么是朴素贝叶斯分类器？

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k|c)$$

- 朴素贝叶斯分类器的生成模型
  - Multinomial NB model & Bernoulli model
- 朴素贝叶斯分类器的的性质
  - 条件独立性假设&位置独立性假设
- 特征选择：互信息、 $\chi 2$ 统计量 、词项频率
- 文本分类的评价：宏平均和微平均

谢谢大家！