信息检索导论

An Introduction to Information Retrieval

第12讲 文本分类及朴素贝叶斯分类器

Text Classification & Naïve Bayes

授课人：林政

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. 文本分类
3. 朴素贝叶斯
4. 朴素贝叶斯理论
5. 文本分类评价

提纲

1. 上一讲回顾
2. 文本分类
3. 朴素贝叶斯
4. 朴素贝叶斯理论
5. 文本分类评价

基于语言模型的IR

- **基本思想**

区别于其他大多数检索模型从查询到文档（即给定用户查询，如何找出相关的文档），语言模型**由文档到查询**，即为每个文档建立不同的语言模型，**判断由文档对应的语言模型抽样出用户查询的可能性**有多大，然后**按照概率由高到低排序**，作为搜索结果。

- **生成查询概率**

为每个文档建立一个语言模型，语言模型代表了单词（或单词序列）在文档中的分布情况。针对查询中的单词，每个单词都有一个抽取概率，将这些单词的抽取概率相乘就是文档生成查询的概率。

两个不同的语言模型

language model of d_1

w	$P(w\|.)$	w	$P(w\|.)$
STOP	.2	toad	.01
the	.2	said	.03
a	.1	likes	.02
frog	.01	that	.04
		...	...

language model of d_2

w	$P(w\|.)$	w	$P(w\|.)$
STOP	.2	toad	.02
the	.15	said	.03
a	.08	likes	.02
frog	.01	that	.05
		...	...

string = frog said that toad likes frog STOP

则 $P(\text{string}|M_{d1}) = 0.01 \cdot 0.03 \cdot 0.04 \cdot 0.01 \cdot 0.02 \cdot 0.01 \cdot 0.2 = 0.0000000000048 = 4.8 \cdot 10^{-12}$

$P(\text{string}|M_{d2}) = 0.01 \cdot 0.03 \cdot 0.05 \cdot 0.02 \cdot 0.02 \cdot 0.01 \cdot 0.2 = 0.0000000000120 = 12 \cdot 10^{-12}$

$P(\text{string}|M_{d1}) < P(\text{string}|M_{d2})$

因此, 相对于d_1，文档 d_2 与字符串 “frog said that toad likes frog STOP” 更相关

统计语言建模IR模型(SLMIR)

- 马萨诸塞大学(University of Massachusetts, UMass)大学Ponte、Croft等人于1998年提出。随后又发展了出了一系列基于SLM的模型。代表系统Lemur。
  - **查询似然模型**：把相关度看成是每篇文档对应的语言下生成该查询的可能性
  - **翻译模型**：假设查询经过某个噪声信道变形成某篇文章，则由文档还原成该查询的概率(翻译模型)可以视为相关度
  - **KL距离模型**：查询对应某种语言，每篇文档对应某种语言，查询语言和文档语言的KL距离作为相关度度量

查询似然模型QLM

- QLM计算公式

$$RSV(Q,D) = P(Q|D) = P(Q|M_D)$$ 已知文档D，抽样出查询Q的概率

$$= P(q_1 q_2 ... q_m | M_D)$$ Md是文档的语言模型

$$= P(q_1|M_D)P(q_2|M_D)...P(q_m|M_D)$$

$$= \prod_{w \in Q} P(w|M_D)^{c(w,Q)}$$

- 于是检索问题转化为估计文档D的一元语言模型M_D，也即求所有词项 w的概率$P(w|M_D)$

QLM求解步骤

- 第一步：根据文档D(样本)，估计文档模型M_D(总体)，在一元模型下，即计算所有词项 w的概率 $P(w|M_D)$
- 第二步：计算在模型M_D下生成查询Q的似然(即概率)
- 第三步：按照得分对所有文档排序

M_D的MLE估计

- 设词项词典的大小为L，则模型M_D的参数可以记为：

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

- MLE估计：

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

- 关键是如何求 $P(D | \vec{\theta}_D)$，也就是说假设这些参数已知的情况下，如何求上述概率。

M_D的参数求解

- 求解 $\vec{\theta}_D^{\,*} = \arg\max_{\vec{\theta}_D} P(D|\vec{\theta}_D) = \arg\max n! \prod_{i=1}^{L} \frac{\theta_i^{c(w_i,D)}}{c(w_i,D)!}$

  $\sum_{i=1}^{L} \theta_i = 1$

- 条件极值问题，采用拉格朗日法求解，得到拉格朗日函数：

$$L(\lambda, \vec{\theta}_D) = n! \prod_{i=1}^{L} \frac{\theta_i^{c(w_i,D)}}{c(w_i,D)!} + \lambda(1 - \sum_{i=1}^{L} \theta_i)$$

- 对每个θ_i求偏导，令其为0，解得：

$$\theta_i^* = P_{ML}(w_i|M_D) = \frac{c(w_i,D)}{\sum_{j=1}^{L} c(w_j,D)} = \frac{c(w_i,D)}{|D|}$$

几种QLM中常用的平滑方法

- Jelinek-Mercer(JM)，$0 \le \lambda \le 1$ 文档估计和文档集估计的混合

$$p(w|D) = \lambda p_{ML}(w|D) + (1-\lambda) p(w|C)$$

  - 课堂提问，对于$w \in D$，折扣后的$P_{DML}(w|D)$是不是一定小于$P_{ML}(w|D)$？

- Dirichlet Priors(Dir)，$\mu \ge 0$

$$p(w|D) = \frac{c(w,D) + \mu p(w|C)}{|D| + \mu}$$

- Absolute Discounting(Abs)，$0 \le \delta \le 1$，$|D|_u$表示D中不相同的词个数(u=unique)

$$p(w|D) = \frac{\max(c(w,D) - \delta, 0)}{|D|} + \frac{\delta |D|_u}{|D|} p(w|C)$$

文档排名函数的转换

- $P(Q|D)=\prod_{w\in Q} p(w|D)^{c(w,Q)}$ $\quad p(w|D)=\begin{cases} p_s(w|D) & w\in D \\ p_u(w|D) & otherwise \end{cases}$

$$\log P(Q|D)=\sum_{w\in Q} c(w,Q)\log p(w|D)=\sum_{w} c(w,Q)\log p(w|D)$$

w不属于Q时，c(w,Q)=0

$$=\sum_{w\in D} c(w,Q)\log p_s(w|D)+\sum_{w\notin D} c(w,Q)\log p_u(w|D)$$

$$=\sum_{w\in D} c(w,Q)\log p_s(w|D)+\sum_{w} c(w,Q)\log p_u(w|D)-\sum_{w\in D} c(w,Q)\log p_u(w|D)$$

$$=\sum_{w\in D\cap Q} c(w,Q)\log \frac{p_s(w|D)}{p_u(w|D)}+\sum_{w\in Q} c(w,Q)\log p_u(w|D)$$

w不属于Q时，c(w,Q)=0

- 将 $p_s(w|D)=p_{DML}(w|D), p_u(w|D)=\alpha_D p(w|C)$ 代入

$$\log P(Q|D)=\sum_{w\in Q\cap D} c(w,Q)\log \frac{p_s(w|D)}{\alpha_D p(w|C)}+|Q|\log\alpha_D+\sum_{w\in Q} c(w,Q)\log p(w|C)$$

查询中w的总次数

不影响排名

- 最终排名函数：$RSV(Q,D)=\sum_{w\in Q\cap D} c(w,Q)\log \frac{p_{DML}(w|D)}{\alpha_D p(w|C)}+|Q|\log\alpha_D$

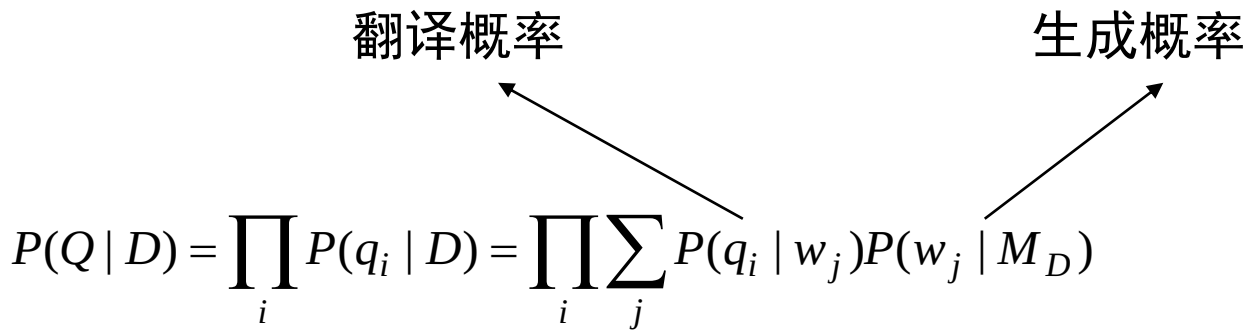
TF

DF

D长度有关

基于翻译模型的IR模型

- 基本的QLM模型不能解决词语失配(word mismatch)问题，即查询中的用词和文档中的用词不一致，如：电脑 vs. 计算机
- 假设Q通过一个有噪声的香农信道变成D，从D估计原始的Q



- 翻译概率$P(q_i|w_j)$在计算时可以将词项之间的关系融入。
  - 基于词典来计算(人工或者自动构造的同义词/近义词/翻译词典)
  - 基于语料库来计算(标题、摘要 vs. 文本；文档锚文本 vs. 文档)

KL距离(相对熵)模型

$$Score(Q,D)=\log\frac{P(Q|M_D)}{P(Q|M_C)}$$

$$Score(Q,D)=\sum_{q_i\in Q}tf(q_i,Q)*\log\frac{P(q_i|M_D)}{P(q_i|M_C)}$$

$$\propto\sum_{q_i\in Q}P(q_i|M_Q)*\log\frac{P(q_i|M_D)}{P(q_i|M_C)}$$

分子分母同乘$P(q_i|M_Q)$

$$=\sum_{q_i\in Q}P(q_i|M_Q)*\log\frac{P(q_i|M_D)}{P(q_i|M_Q)}-\sum_{q_i\in Q}P(q_i|M_Q)*\log\frac{P(q_i|M_C)}{P(q_i|M_Q)}$$

$$=-KL(M_Q,M_D)+KL(M_Q,M_C)$$

对同一Q，为常数

$$\propto -KL(M_Q,M_D)=\sum_{q_i\in Q}P(q_i|M_Q)*\log P(q_i|M_D)-\sum_{q_i\in Q}P(q_i|M_Q)*\log P(q_i|M_Q)$$

负的交叉熵

$$\propto\sum_{q_i\in Q}P(q_i|M_Q)*\log P(q_i|M_D)$$

$$P(Q|M_D)=\frac{|Q|!}{\prod_{q_i\in Q}tf(q_i,Q)!}\prod_{q_i\in Q}P(q_i|D)^{tf(q_i,Q)}$$

$$P(Q|M_C)=\frac{|Q|!}{\prod_{q_i\in Q}tf(q_i,Q)!}\prod_{q_i\in Q}P(q_i|C)^{tf(q_i,Q)}$$

多项分布

查询对应某种语言，每篇文档对应某种语言，查询语言和文档语言的KL距离作为相关度度量

QLM模型小结

检索问题

①把相关度看成是每篇文档对应的语言下生成该查询的可能性

②求文档背后的语言模型。在一元模型下，即求所有词w的概率$P(w|M_D)$

一元模型估计

多项式分布假设

③D是抛1个L面的骰子抛$|D|$次生成的，将每次朝上的那面对应的词项集合起来便生成文本D。

MLE估计

平滑技术

本讲内容

- 文本分类的概念及其与IR的关系
- 朴素贝叶斯分类器(朴素贝叶斯)
- 文本分类的评价

提纲

1. 上一讲回顾
2. 文本分类
3. 朴素贝叶斯
4. 朴素贝叶斯理论
5. 文本分类评价

什么是分类？

- 简单地说，分类(Categorization or Classification)就是按照某种标准给对象贴标签(label)

为什么要分类？

- 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！

分类非常普遍

- 性别、籍贯、民族、学历、年龄等等，我们每个人身上贴满了“标签”
- 我们从孩提开始就具有分类能力：爸爸、妈妈；好阿姨、坏阿姨；电影中的好人、坏人等等。
- 分类无处不在，从现在开始，我们可以以分类的眼光看世界☺

课堂思考题

- 从如下叙述中找出“标签”
  - 你以为我穷，不好看，就没有感情吗？我也会有的。如果上帝赋予我财富和美貌，我一定要让你难于离开我，就像我现在难于离开你。上帝没有这样，我们的精神是同等的，就如同你跟我经过坟墓，将同样地站在在上帝面前 (From 《简爱》)

多标签分类

课堂思考题

- 从如下叙述中找出“标签”
  - 你以为我穷，不好看，就没有感情吗？我也会有的。如果上帝赋予我财富和美貌，我一定要让你难于离开我，就像我现在难于离开你。上帝没有这样，我们的精神是同等的，就如同你跟我经过坟墓，将同样地站在在上帝面前 (From 《简爱》)

课堂思考题

- 从如下叙述中找出“标签”
  - 你以为我穷，不好看，就没有感情吗？我也会有的。如果上帝赋予我财富和美貌，我一定要让你难于离开我，就像我现在难于离开你。上帝没有这样，我们的精神是同等的，就如同你跟我经过坟墓，将同样地站在在上帝面前 (From 《简爱》)

穷、不好看，有感情，精神高尚

分类是有监督机器学习的一种

机器学习(Machine Learning)

- **有监督学习(Supervised Learning)**
- **无监督学习(Unsupervised Learning)**
- **半监督学习（Semi-supervised Learning）**
- **强化学习(Reinforcement Learning)**

有监督学习

有监督学习（supervised learning）：从给定的**有标注的训练数据集**中学习出一个函数（模型参数），当新的数据到来时可以根据这个函数预测结果。 常见任务包括**分类**与**回归**

Classification: Y is discrete

Y：年轻人(1)，老年人(-1)

X： x_1 黑头发的比例，　值域（0，1）；

x_2 行走速度，　值域（0，100）米/每分钟.

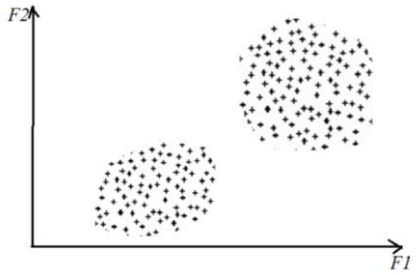
Training Data:

Y=1：(1，99)、(0.9，80)、(0.80，100) …

Y=-1：(0.2，30)、(0.5，50)、(0.4，30) …

Test:

X=(0.85, 98),　Y=?



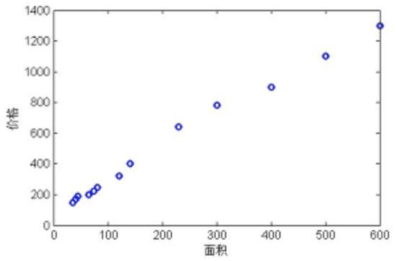
Regression: Y is continue

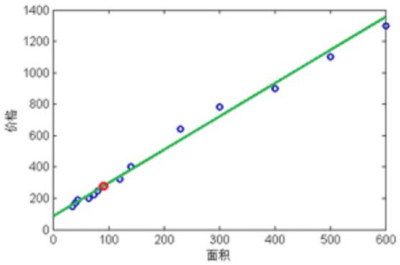
Y：房屋价钱（万元），值域 Y>=0.

X: x1=房屋面积 m^2.

Training Data:

35	150
40	170
45	190
65	200
74	224
80	245
120	320
140	400
230	640
300	780
400	900
500	1100
600	1300

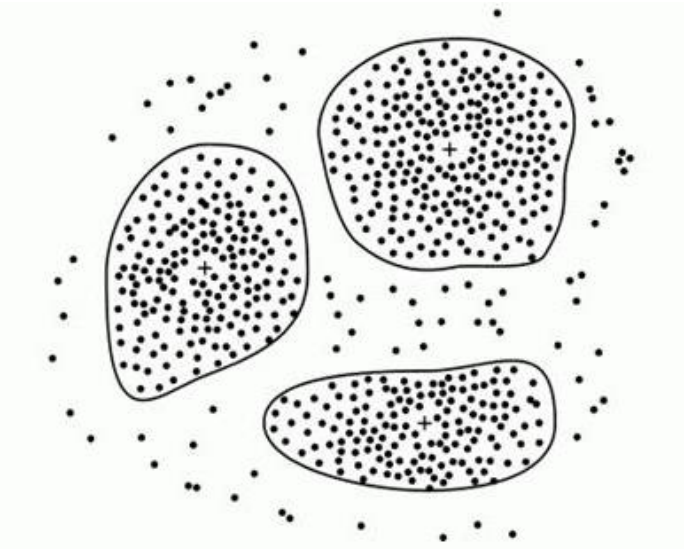


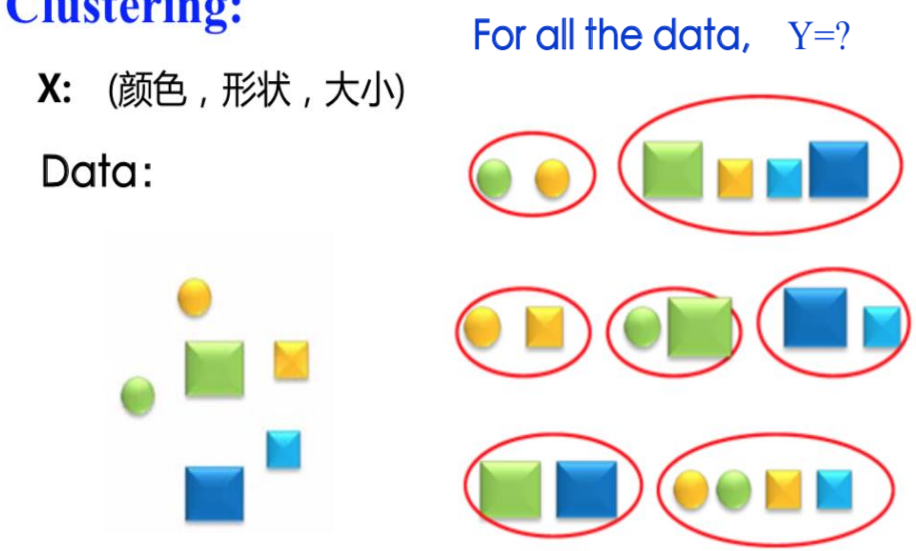


Test:　X=90　　Y= ?　　$y=ax+b$

无监督学习

无监督学习（unsupervised learning）：没有标注的训练数据集，需要根据样本间的统计规律对样本集进行分析，常见任务如**聚类**等



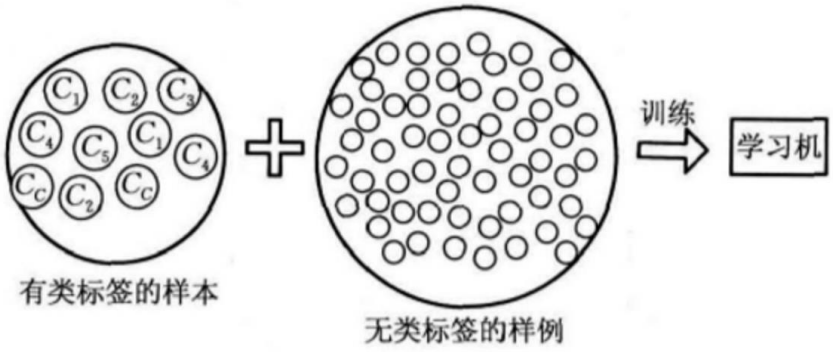


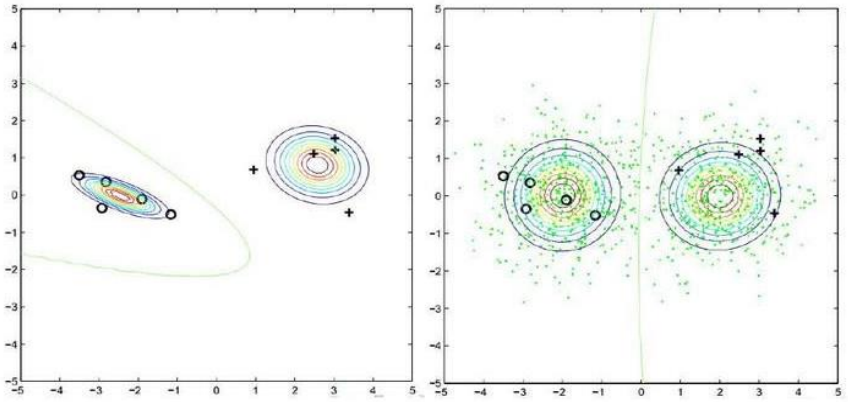
半监督学习

半监督学习（Semi-supervised learning）：结合**（少量的）标注训练数据**和**（大量的）未标注数据**来进行数据的分类学习。

两个基本假设：

- **聚类假设：**处在相同聚类中的样本示例有较大的可能拥有相同的标记。根据该假设，决策边界就应该尽量通过数据较为稀疏的地方。
- **流形假设：**处于一个很小的局部区域内的样本示例具有相似的性质，因此，其标记也应该相似。在该假设下，大量未标记示例的作用就是让数据空间变得更加稠密，从而有助于更加准确地刻画局部特性，使得决策函数能够更好地进行数据拟合。





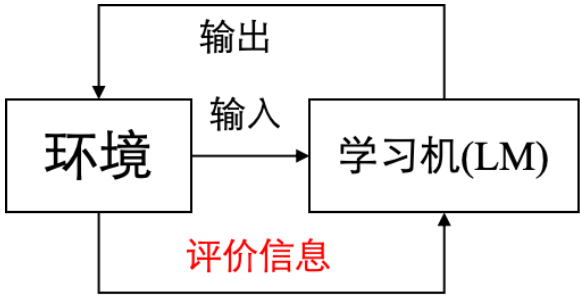
强化学习、多任务学习

强化学习（Reinforcement Learning）：外部环境对输出只给出评价信息而非正确答案，学习机通过强化受奖励的动作来改善自身的性能。

强化学习 VS 有监督学习

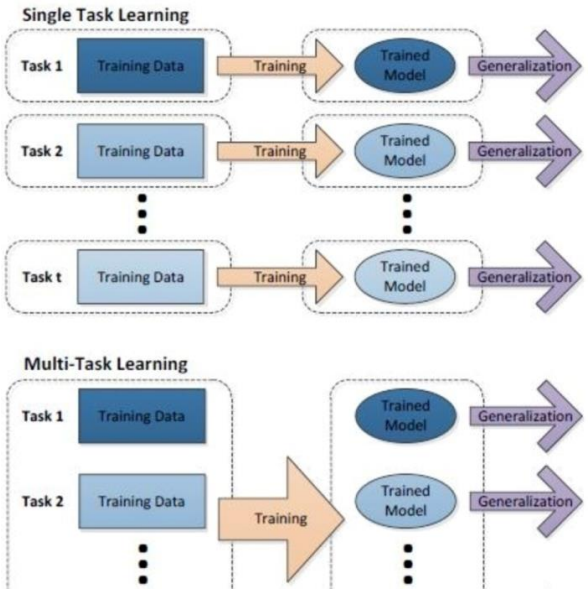
（1）目标不一样，强化学习看重的是行为序列下的长期收益，而监督学习往往关注的是标签和输出的误差。

（2）强化学习的奖惩概念是没有正确或错误之分的，而监督学习标签就是正确的，并且强化学习是一个学习+决策的过程，有和环境交互的能力（交互的结果以惩罚的形式返回）



多任务学习（Multi-task Learning）：把多个相关（related）的任务放在一起同时学习。

单任务学习时，各个任务之间的模型空间（Trained Model）是相互独立的，但现实世界中很多问题不能分解为一个一个独立的子问题，且这样忽略了问题之间所包含的丰富的关联信息。多任务学习就是为了解决这个问题而诞生的。多个任务之间共享一些因素，它们可以在学习过程中，共享它们所学到的信息，**相关联的多任务学习比单任务学习具备更好的泛化（generalization）效果。**



文本分类

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

- 分类体系：随应用不同而不同。比如：垃圾 vs. 非垃圾、体育/经济/军事 等等

- 文本分类的类型：
  - 按类别数目：
    - binary vs. multi-class：二类问题 vs. 多类问题
  - 按每篇文档赋予的标签数目：
    - sing label vs. multi label：单标签 vs. 多标签问题

一个文本分类任务：垃圾邮件过滤

```
From: ''' <takworlld@hotmail.com>
Subject: real estate is the only way... gem oalvgkay
Anyone can buy real estate with no money down
Stop paying rent TODAY !
There is no need to spend hundreds or even thousands for
similar courses
I am 22 years old and I have already purchased 6 properties
using the
methods outlined in this truly INCREDIBLE ebook.
Change your life NOW !
=================================================
Click Below to order:
http://www.wholesaledaily.com/sales/nmd.htm
=================================================
```

如何编程实现对上类信息的识别和过滤？

文本分类的形式化定义： 训练

给定:

- 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 (如, 相关类 vs. 不相关类)
- 训练集 D，文档d用c来标记，$\langle d, c \rangle \in X \times C$

利用学习算法，可以学习一个分类器 γ，它可以将文档映射成类别：

$$\gamma : X \rightarrow C$$

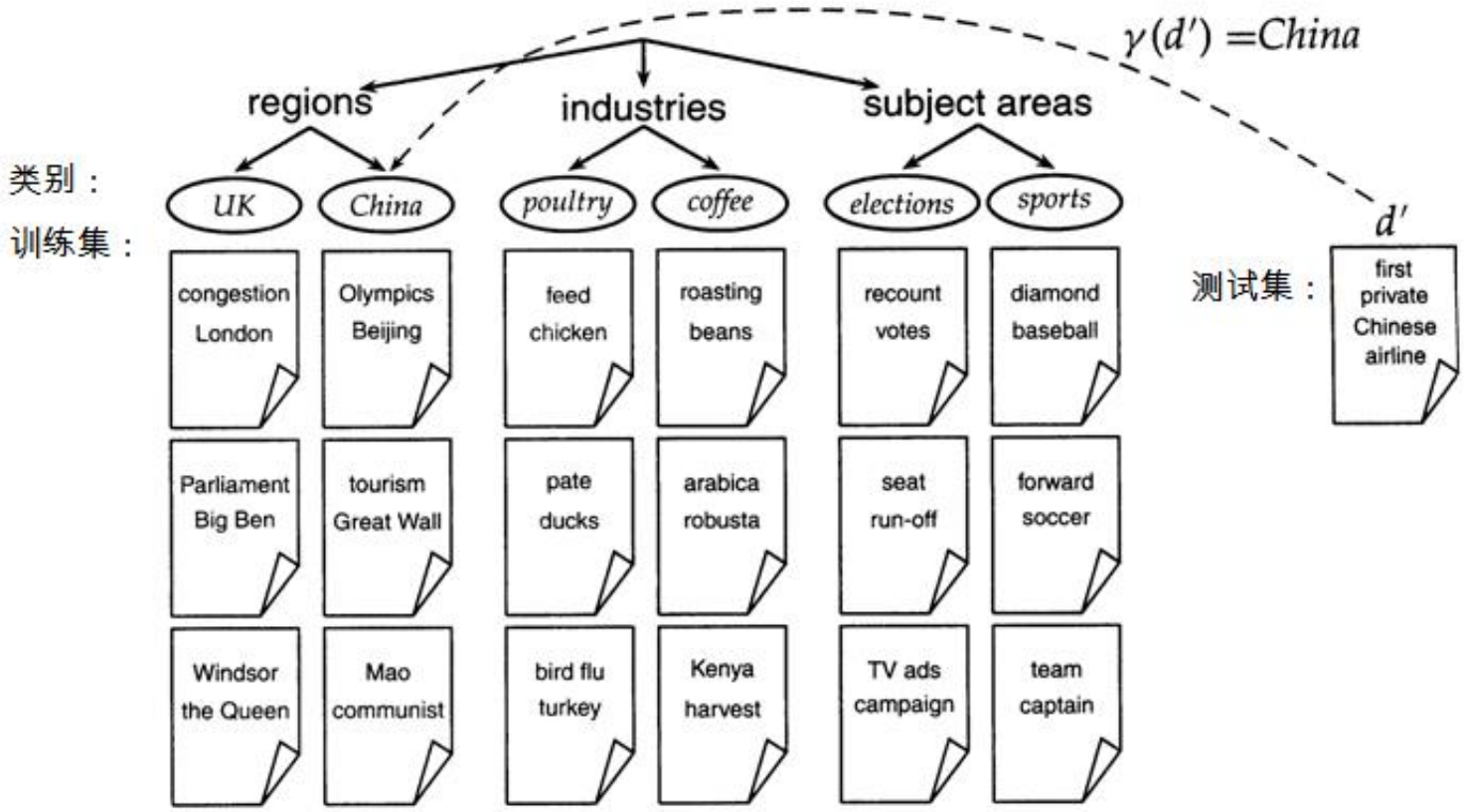
文本分类的形式化定义：应用/测试

给定： $d \in X$

确定： $\Upsilon(d) \in C$,

即确定d最可能属于的类别

主题分类——二级分类


$\gamma(d') = China$
regions
industries
subject areas
类别：
UK
China
poultry
coffee
elections
sports
训练集：
congestion London
Olympics Beijing
feed chicken
roasting beans
recount votes
diamond baseball
Parliament Big Ben
tourism Great Wall
pate ducks
arabica robusta
seat run-off
forward soccer
Windsor the Queen
Mao communist
bird flu turkey
Kenya harvest
TV ads campaign
team captain
测试集：
d'
first private Chinese airline

课堂练习

- 试举出文本分类在信息检索中的应用例子

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

- Google Alerts的例子是基于规则分类的
- 存在一些IDE开发环境来高效撰写非常复杂的规则 (如Verity)
- 通常情况下都是布尔表达式组合 (如Google Alerts)
- 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

一个Verity主题 (一条复杂的分类规则)

```
# Beginning of art topic definition
art ACCRUE
    /author = "fsmith"
    /date   = "30-Dec-01"
    /annotation = "Topic created
                   by fsmith"
* 0.70 performing-arts ACCRUE
** 0.50 WORD
    /wordtext = ballet
** 0.50 STEM
    /wordtext = dance
** 0.50 WORD
    /wordtext = opera
** 0.30 WORD
    /wordtext = symphony
* 0.70 visual-arts ACCRUE
** 0.50 WORD
    /wordtext = painting
** 0.50 WORD
    /wordtext = sculpture
* 0.70 film ACCRUE
** 0.50 STEM
    /wordtext = film
** 0.50 motion-picture PHRASE
*** 1.00 WORD
    /wordtext = motion
*** 1.00 WORD
    /wordtext = picture
** 0.50 STEM
    /wordtext = movie
* 0.50 video ACCRUE
** 0.50 STEM
    /wordtext = video
** 0.50 STEM
    /wordtext = vcr
# End of art topic
```

comment line
top-level topic
topic definition modifiers
subtopic topic
evidence topic
topic definition modifier
evidence topic
topic definition modifier
evidence topic
topic definition modifier
subtopic
subtopic
subtopic

分类方法: 3. 统计/概率方法

- 文本分类被定义为一个学习问题，这也是本书中的定义，包括：
  - (i) **训练(training)**：通过有监督的学习，得到分类函数γ，然后将其
  - (ii) **测试/应用/分类(test)**：应用于对新文档的分类
- 后面将介绍一系列分类方法: 朴素贝叶斯、Rocchio、kNN和SVM
- 当然，世上没有免费的午餐：需要手工构建训练集
- 但是，该手工工作一般人就可以完成，不需要专家。

提纲

朴素贝叶斯(Naïve Bayes)分类器

- 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（多项式模型）：

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

  - n_d 是文档的长度(词条的个数)
  - $P(t_k \mid c)$ 是词项t_k 出现在类别c中文档的概率，即类别c文档的一元语言模型！
  - $P(t_k \mid c)$ 度量的是当c是正确类别时t_k 的贡献
  - $P(c)$ 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

具有最大后验概率的类别

- 朴素贝叶斯分类的目标是寻找“最佳”的类别
- 最佳类别是指具有最大后验概率(**maximum a posteriori - MAP**)的类别 c_{map}:

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

对数计算

- 很多小概率的乘积会导致浮点数下溢出
- 由于 $\log(xy) = \log(x) + \log(y)$, 可以通过取对数将原来的乘积计算变成求和计算
- 由于log是单调函数，因此得分最高的类别不会发生改变
- 因此，实际中常常使用的是：

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

朴素贝叶斯分类器

- 分类规则:

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

- 简单说明:
  - 每个条件参数 $\hat{P}(t_k|c)$ 是反映t_k 对c的贡献高低的一个权重
  - 先验概率 $\hat{P}(c)$ 是反映类别c的相对频率的一个权重
  - 因此，所有权重的求和反映的是文档属于类别的可能性
  - 选择最具可能性的类别

参数估计：极大似然估计

- 如何从训练数据中估计 $\hat{P}(c)$ 和 $\hat{P}(t_k|c)$ ？
- 先验:

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

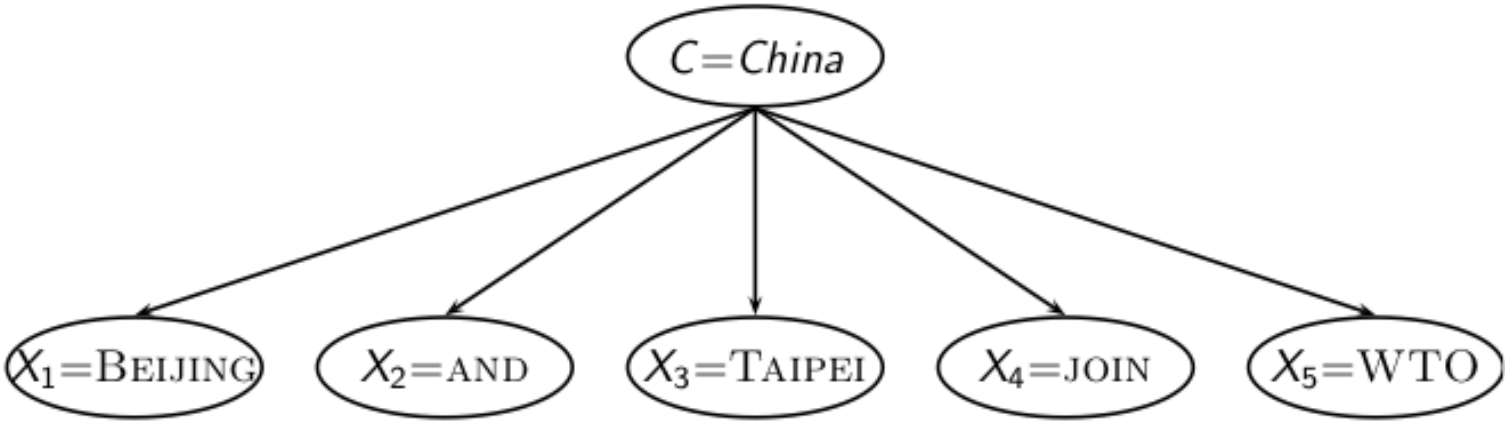
- N_c：类c中的文档数目；N: 所有文档的总数
- 条件概率:

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

- T_{ct} 是训练集中类别c中的词条t的个数 (多次出现要计算多次)
- 给定如下的 **位置独立性假设(positional independence assumption):**

$$\hat{P}(t_{k_1}|c) = \hat{P}(t_{k_2}|c)$$

MLE估计中的问题：零概率问题



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

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

MLE估计中的问题：零概率问题（续）

- 如果 WTO 在训练集中没有出现在类别 China中，那么就会有如下的零概率估计：

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

- 那么，对于任意包含WTO的文档d，$P(\mathrm{China}|d) = 0$。
- 一旦发生零概率，将无法判断类别

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

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

- B 是不同的词语个数 (这种情况下词汇表大小 $|V|$ = B)

避免零概率: 加一平滑（续）

- 利用加1平滑从训练集中估计参数
- 对于新文档，对于每个类别，计算
  - (i) 先验的对数值之和以及
  - (ii) 词项条件概率的对数之和
- 将文档归于得分最高的那个类

朴素贝叶斯: 训练过程

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** $V, prior, condprob$

朴素贝叶斯: 测试/应用/分类

```
ApplyMultinomialNB(ℂ, V, prior, condprob, d)
1  W ← ExtractTokensFromDoc(V, d)
2  for each c ∈ ℂ
3  do score[c] ← log prior[c]
4     for each t ∈ W
5     do score[c] += log condprob[t][c]
6  return arg max_{c∈ℂ} score[c]
```

课堂练习

	docID	words in document	in c = *China*?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

- 估计朴素贝叶斯分类器的参数
- 对测试文档进行分类

例子: 参数估计

先验 $\hat{P}(c) = 3/4$ 及 $\hat{P}(\overline{c}) = 1/4$ ，而条件概率如下：

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

上述计算中的分母分别是 (8 + 6) 和 (3 + 6)，这是因为$text_c$ 和 $text_{\overline{c}}$ (分别代表两类文档集的大小)的大小分别是8和3，而词汇表大小是6。上述概率还是平滑后的概率。

例子: 分类

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

因此, 分类器将测试文档分到c = *China*类，这是因为d_5中起正向作用的CHINESE出现3次的权重高于起反向作用的 JAPAN和TOKYO的权重之和。

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， V: 词汇表 $\mathbb{C}$: 类别集合
- $\Theta(|\mathbb{D}|L_{ave})$是计算所有统计数字的时间
- $\Theta(|\mathbb{C}||V|)$ 是从上述数字计算参数的时间
- 通常来说: $|\mathbb{C}||V| < |\mathbb{D}|L_{ave}$
- 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。这在某种意义上是最优的。**

提纲

朴素贝叶斯: 分析

- 接下来对朴素贝叶斯的性质进行更深层次的理解
- 先形式化地推导出分类规则
- 然后介绍在推导中的假设

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

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

应用贝叶斯定律 $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$:

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

由于分母$P(d)$对所有类别都一样，因此可以去掉，有:

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

过多参数/稀疏性问题

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} P(d|c)P(c)$$
$$= \arg\max_{c \in \mathbb{C}} P(\langle t_1, \ldots, t_k, \ldots, t_{n_d} \rangle | c)P(c)$$

- 上式中存在过多的参数 $P(\langle t_1, \ldots, t_k, \ldots, t_{n_d} \rangle | c)$，每个参数都是一个类别和一个词语序列的组合
- 要估计这么多参数，必须需要大量的训练样例。但是，训练集的规模总是有限的
- 于是出现**数据稀疏性(data sparseness)**问题

朴素贝叶斯条件独立性假设

为减少参数数目，给出朴素贝叶斯条件独立性假设：

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

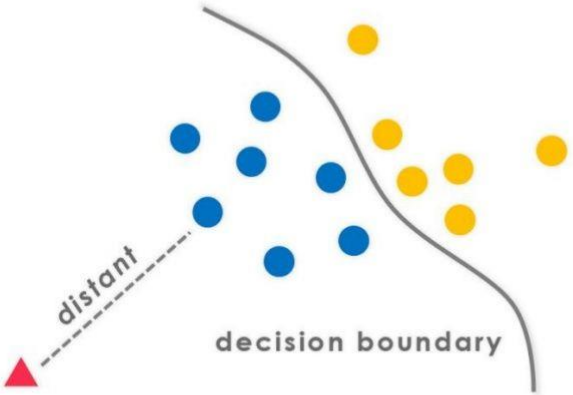
假定上述联合概率等于某个独立概率$P(X_k = t_k | c)$的乘积。前面我们提到可以通过如下方法来估计这些先验概率和条件概率：

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

生成模型 VS 判别模型

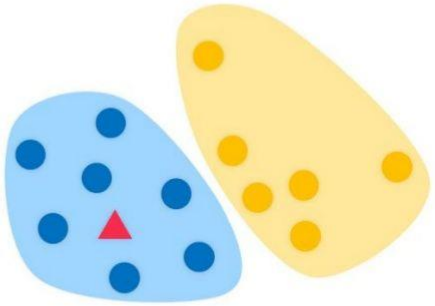
Discriminative vs. Generative

Discriminative



- Only care about estimating the conditional probabilities
- Very good when underlying distribution of data is really complicated (e.g. texts, images, movies)

Generative

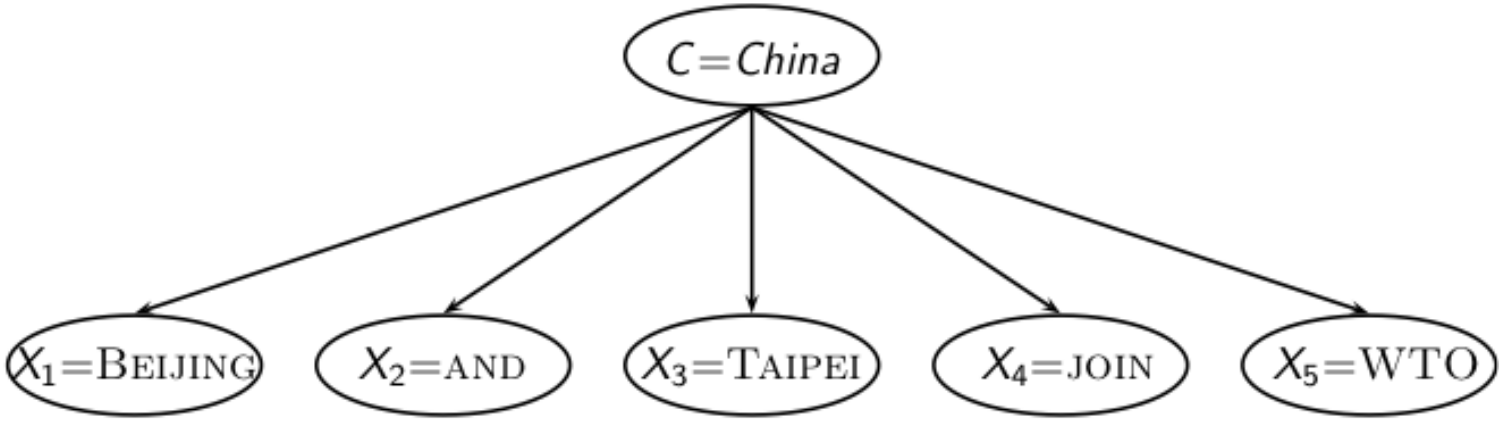


- Model observations (x,y) first, then infer p(y|x)
- Good for missing variables, better diagnostics
- Easy to add prior knowledge about data

机器学习的任务是从属性X预测标记Y，即求概率P(Y|X)

- 判别式模型求的是P(Y|X)，对未见示例X，根据P(Y|X)可以求得标记Y
- 生成式模型求的是P(Y, X)，对于未见示例X，要求出X与不同标记之间的联合概率分布，然后大的获胜

生成式(Generative)模型



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

生成过程：

- 利用概率$P(c)$产生一个类
- 以该类为条件，（在各自位置上）基于概率$P(t_k|c)$产生每个词语，这些词语之间相互独立
- 对文档分类时，找出最有可能生成该文档的类别

第二个独立性假设：位置独立性假设

- $\hat{P}(t_{k_1}|c) = \hat{P}(t_{k_2}|c)$
- 例如，对于*UK*类别中的一篇文档，在第一个位置上生成QUEEN的概率和在最后一个位置上生成它的概率一样
- 上述两个独立性假设实际上是**词袋模型(bag of words model)**

朴素贝叶斯独立性假设不成立的情况

- 自然语言文本中，上述独立性假设并不成立
- 条件独立性假设:

$$P(\langle t_1, \ldots, t_{n_d} \rangle | c) = \prod_{1 \leq k \leq n_d} P(X_k = t_k | c)$$

- 位置独立性假设:

$$\hat{P}(t_{k_1} | c) = \hat{P}(t_{k_2} | c)$$

- 课堂练习
  - 给出条件独立性假设不成立的例子
  - 给出位置独立性假设不成立的例子
- 在这些假设都不成立的情况下，为什么朴素贝叶斯方法有用？

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
- 例子

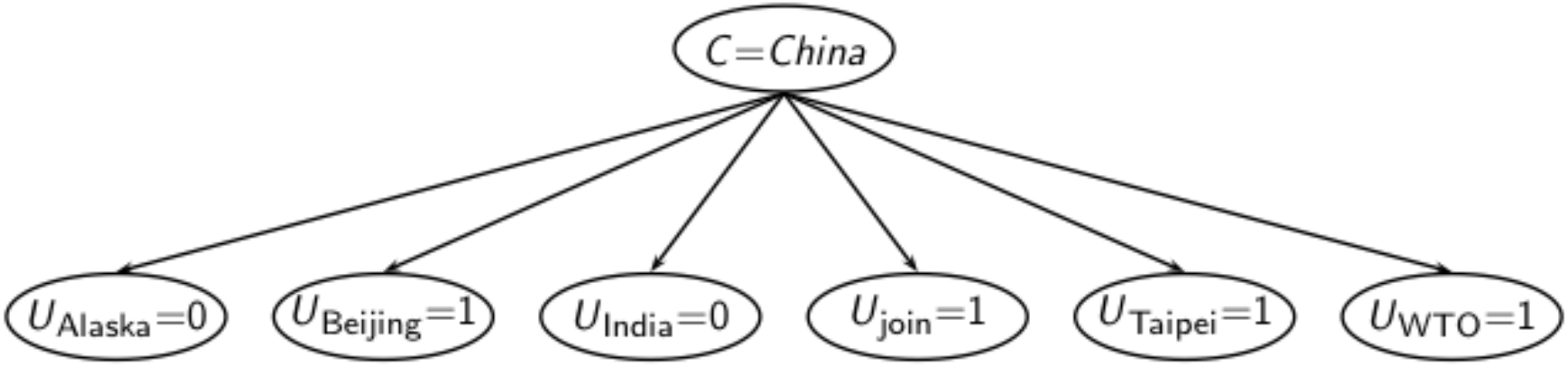
	c_1	c_2	class selected
true probability $P(c\|d)$	0.6	0.4	c_1
$\hat{P}(c)\prod_{1\leq k\leq n_d}\hat{P}(t_k\|c)$	0.00099	0.00001	
NB estimate $\hat{P}(c\|d)$	0.99	0.01	c_1

- 概率P($c_2|d$)被过低估计(0.01)，而概率P($c_1|d$)被过高估计(0.99)。
- 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!

朴素贝叶斯并不朴素

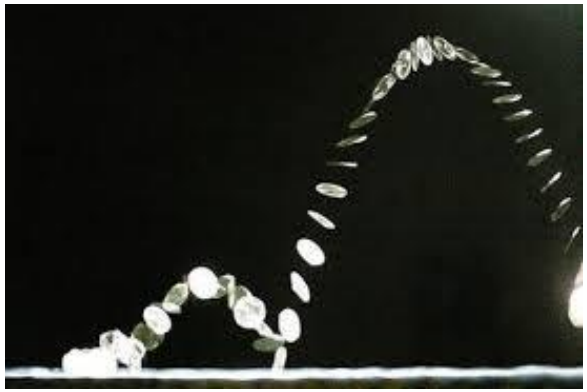
- 朴素贝叶斯在多次竞赛中胜出 (比如 KDD-CUP 97)
- 相对于其他很多更复杂的学习方法，朴素贝叶斯对不相关特征更具鲁棒性
- 相对于其他很多更复杂的学习方法，朴素贝叶斯对概念漂移(concept drift)更鲁棒(概念漂移是指类别的定义随时间变化)
  - 概念漂移：垃圾邮件的主题随时间变化
  - 基于反馈的动态贝叶斯算法：对新样本或者错误分类样本进行重点学习）
- 当有很多同等重要的特征时，该方法优于决策树类方法
- 一个很好的文本分类基准方法 (当然，不是最优的方法)
- (训练和测试)速度非常快
- 存储开销少

另一个朴素贝叶斯的实现： 贝努利模型



- 回想一下BIM模型，此时每个类别c生成文档d是基于贝努利模型的$P(d|c)$
- 此时，词项在文档中只有出现与不出现两种
- 词项的出现之间仍然相互独立，计算时要考虑词项不出现的概率

朴素贝叶斯的两种实现方式-贝努利模型



- 基于贝努利模型的实现方法：贝努利模型不考虑词在文档中出现的次数，只考虑出不出现，因此在这个意义上相当于假设词是等权重的。贝努利模型是一种以文档作为计算粒度的方法。
  - 每个类对应一堆硬币
  - 每个硬币代表一个词
  - 硬币的个数等于单词的个数
  - 类中的一篇文本是通过投掷对应类的所有硬币产生的

$$P(c|d) = P(d|c)P(c)/P(d) \propto P(d|c)P(c) = P(c)\prod_{t\in d} P(t|c)\prod_{t\notin d}\left(1-P(t|c)\right)$$

$$\hat{P}(c) = \frac{N_c}{N} \qquad \hat{P}(t|c) = \frac{N_{ct}+1}{N_c+2}$$

朴素贝叶斯的两种实现方式-多项式模型

- 基于多项式模型的实现方法：多项式模型中各单词类条件概率计算考虑了词出现的次数，多项式模型是一种以词作为计算粒度的方法。前面讨论的就是这种方法。
  - 每个不规则骰子代表一个类
  - 骰子的每个面代表一个词
  - 面的个数等于文本中的单词个数
  - 每个类的一篇文本是通过投掷上述对应骰子产生的

$$P(c \mid d) = P(d \mid c)P(c) / P(d) \propto P(d \mid c)P(c)$$

$$P(d|c) = P(< tf_{t_1,d}, \ldots, tf_{t_M,d} > |c) \propto \prod_{1 \le i \le M} P(X = t_i|c)^{tf_{t_i,d}}$$

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

多项式NB vs. 贝努利NB

- 基于多项式模型的实现方法：多项式模型中各单词类条件概率计算考虑了词出现的次数，**多项式模型是一种以词作为计算粒度的方法**。
  - 每个类对应一个不规则骰子
  - 每个词对应骰子的一个面
  - 面的个数等于单词的个数
  - 每个类的一篇文本是通过投掷上述对应骰子产生的

- 基于贝努利模型的实现方法：贝努利模型不考虑词在文档中出现的次数，只考虑出不出现，因此在这个意义上相当于假设词是等权重的。**贝努利模型是一种以文档作为计算粒度的方法**。
  - 每个类对应一堆硬币
  - 每个词对应一枚硬币
  - 硬币的个数等于单词的个数
  - 每个类的一篇文本是通过投掷对应类的所有硬币产生的

连续值的情况处理—高斯朴素贝叶斯

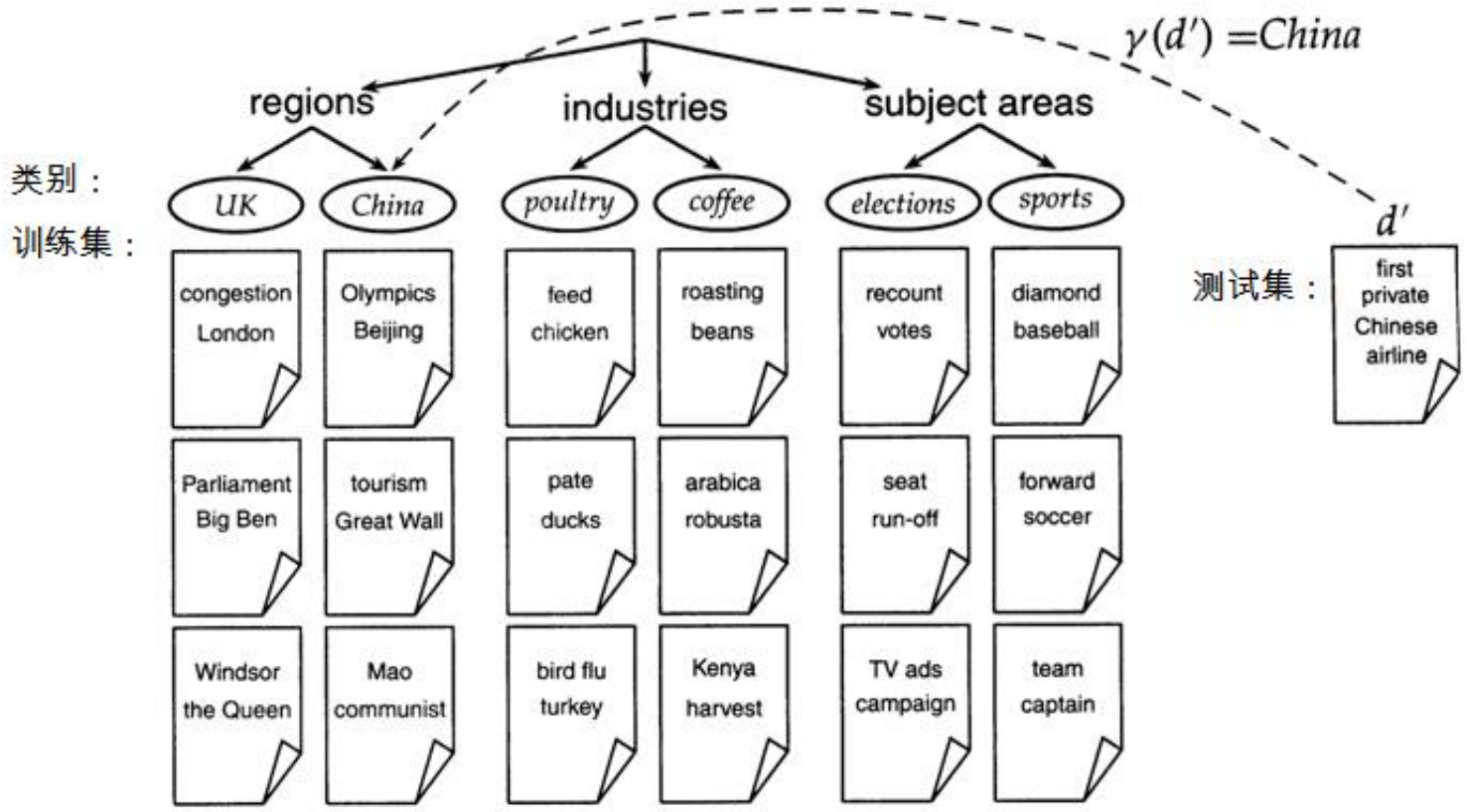
- 以上模型中，可以看成输入为特征向量$\{x_1, x_2, \ldots, x_n\}$
- 对于多项式模型，每个x_i的值可以看成对应特征t出现的次数
- 对于贝努利模型，每个x_i的值可以看成一个布尔值，分别对应t出现还是不出现
- 如果特征x_i取连续值，比如温度，可以采用高斯模型来求解

$$P(x_i \mid c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left(-\frac{(x_i - \mu_c)^2}{2\sigma_c^2}\right)$$

- 其中均值和方差可以利用MLE估计求解

提纲

Reuters语料上的评价



例子：Reuters语料

symbol	statistic	value
N	documents	800,000
L	avg. # word tokens per document	200
M	word types	400,000
	avg. # bytes per word token (incl. spaces/punct.)	6
	avg. # bytes per word token (without spaces/punct.)	4.5
	avg. # bytes per word type	7.5
	non-positional postings	100,000,000

type of class	number	examples
region	366	UK, China
industry	870	poultry, coffee
subject area	126	elections, sports

一篇Reuters文档

REUTERS

You are here: Home > News > Science > Article

Go to a Section: U.S. International Business Markets Politics Entertainment Technology Sports Oddly Enou

Extreme conditions create rare Antarctic clouds

Tue Aug 1, 2006 3:20am ET

Email This Article | Print This Article | Reprints

[-] Text [+]

SYDNEY (Reuters) - Rare, mother-of-pearl colored clouds caused by extreme weather conditions above Antarctica are a possible indication of global warming, Australian scientists said on Tuesday.

Known as nacreous clouds, the spectacular formations showing delicate wisps of colors were photographed in the sky over an Australian

分类评价

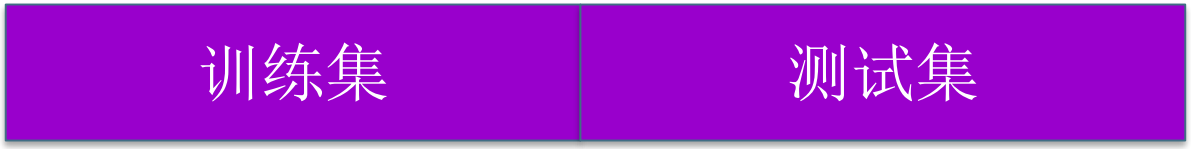
- 评价必须基于测试数据进行，而且该测试数据是与训练数据完全独立的 (通常两者样本之间无交集)
- 很容易通过训练可以在训练集上达到很高的性能 (比如记忆所有的测试集合)
- 指标：正确率、召回率、 F_1值、分类精确率(classification accuracy)等等

关于训练集和测试集

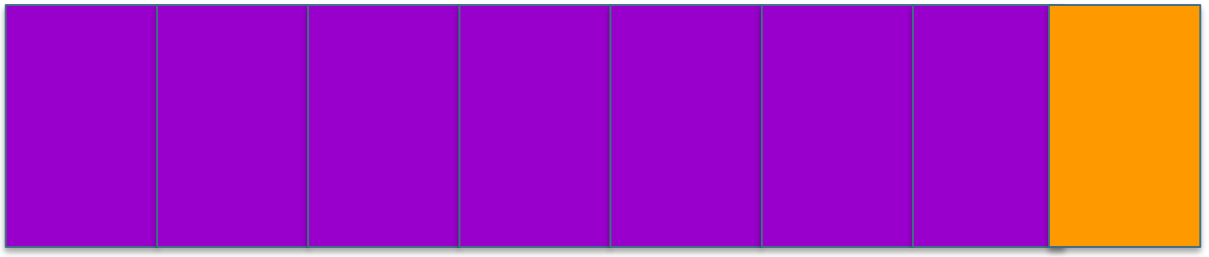
- 给定一个已标注好的数据集，将其中一部分划为训练集(training set)，另一部分划为测试集(test set)。在训练集上训练，训练得到的分类器用于测试集，计算测试集上的评价指标。
- 另一种做法：将上述整个数据集划分成k份，然后以其中k-1份为训练集训练出一个分类器，并用于另一份(测试集)，循环k次，将k次得到的评价指标进行平均。这种做法称为k交叉验证(k-cross validation)
- 有时，分类器的参数需要优化。此时，可以仍然将整个数据集划分成训练集和测试集，然后将训练集分成k份(其中k-1份作为训练，另一份称为验证集validation set，也叫开发集)，在训练集合上进行k交叉验证，得到最优参数。然后应用于测试集。

关于训练集和测试集

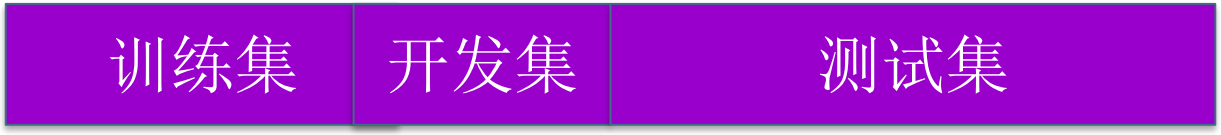
训练+测试

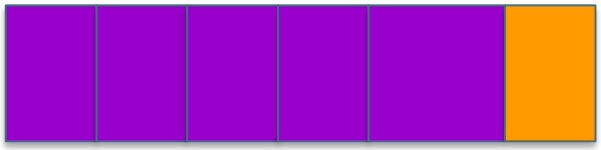


k交叉测试



参数确定





正确率P 及召回率 R

	in the class	not in the class
predicted to be in the class	true positives (TP)	false positives (FP)
predicted to not be in the class	false negatives (FN)	true negatives (TN)

$$P = TP / (TP + FP)$$

$$R = TP / (TP + FN)$$

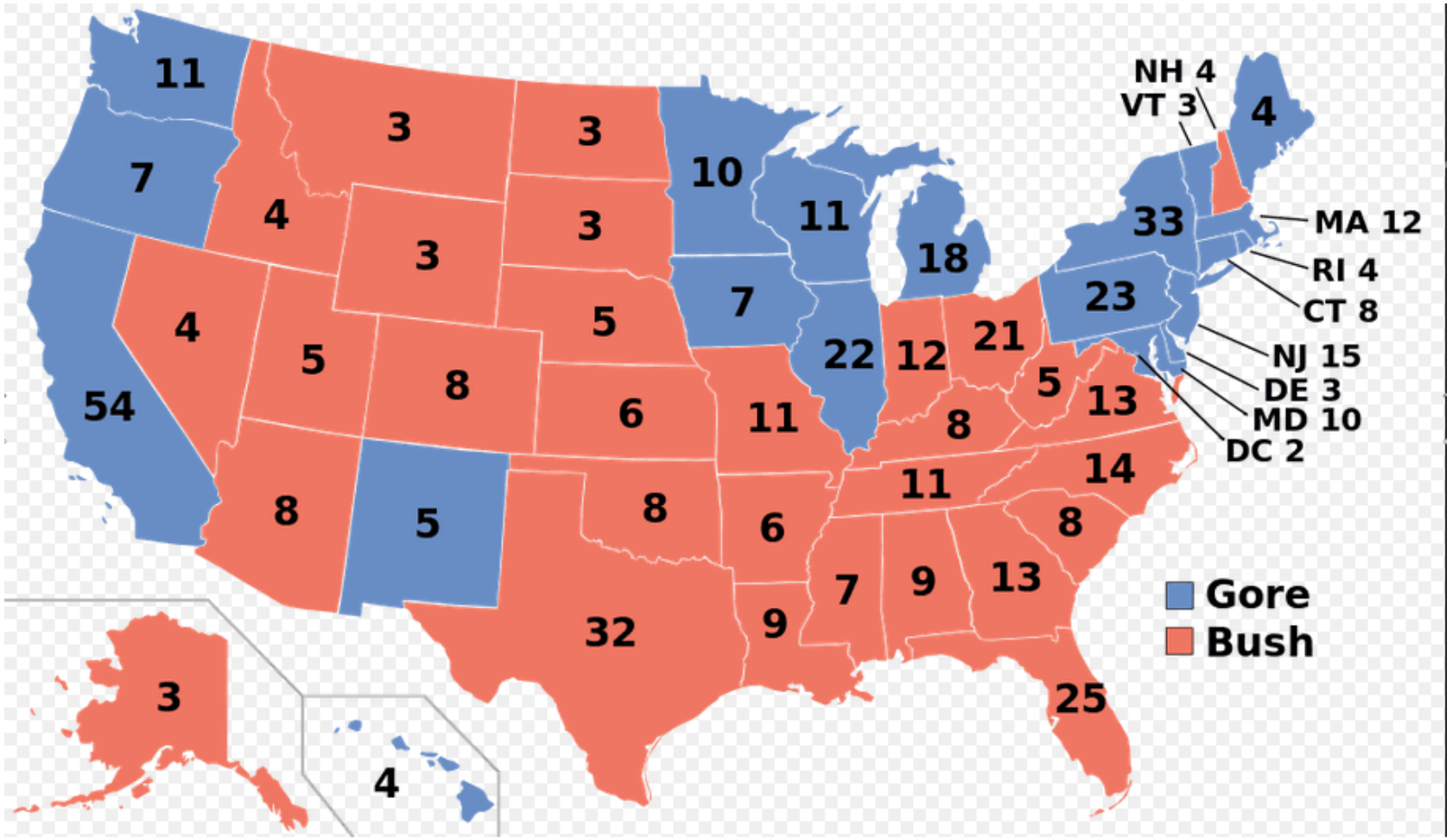
F值

- F_1 允许在正确率和召回率之间达到某种均衡
- $$F_1 = \frac{1}{\frac{1}{2}\frac{1}{P} + \frac{1}{2}\frac{1}{R}} = \frac{2PR}{P+R}$$
- 也就是P和R的调和平均值：$$\frac{1}{F} = \frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)$$

微平均 vs. 宏平均

- 对于一个类我们得到评价指标F_1
- 但是我们希望得到在所有类别上的综合性能
- **宏平均(Macroaveraging)：以类别为单位**
  - 对类别集合C 中的每个类都计算一个F_1 值
  - 对C个结果求算术平均
- **微平均(Microaveraging)：**以类别-文档对为单位
  - 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

美国大选规则：Macro/Micro的折中？



朴素贝叶斯 vs. 其他方法

(a)

	NB	Rocchio	kNN	SVM
micro-avg-L (90 classes)	80	85	86	89
macro-avg (90 classes)	47	59	60	60

(b)

	NB	Rocchio	kNN	trees	SVM
earn	96	93	97	98	98
acq	88	65	92	90	94
money-fx	57	47	78	66	75
grain	79	68	82	85	95
crude	80	70	86	85	89
trade	64	65	77	73	76
interest	65	63	74	67	78
ship	85	49	79	74	86
wheat	70	69	77	93	92
corn	65	48	78	92	90
micro-avg (top 10)	82	65	82	88	92
micro-avg-D (118 classes)	75	62	n/a	n/a	87

Evaluation measure: F_1 Naive Bayes does pretty well, but some methods beat it consistently (e.g., SVM).

本讲小结

- 文本分类的概念及其与IR的关系
- 朴素贝叶斯分类器(朴素贝叶斯)
- 文本分类的评价

扩展

- 贝叶斯分类器编程注意事项
- 常用半监督学习方法
- 如何对关系建模，如何对位置建模
- 自然语言太难了

参考资料

- 《信息检索导论》第13 章
- Weka: 一个包含了 朴素贝叶斯在内的数据挖掘工具包
- Reuters-21578 – 最著名的文本分类语料 (当然，当前已经显得规模太小)
- RCV1是更大的常用的分类数据集

- 一个著名的文本分类综述(引用5000多次)： F Sebastiani , Machine Learning in Automated Text Categorization, ACM Computing Surveys 34(1):1--47 (2002)