信息组织与检索

第13讲：文本分类

主讲人：张蓉

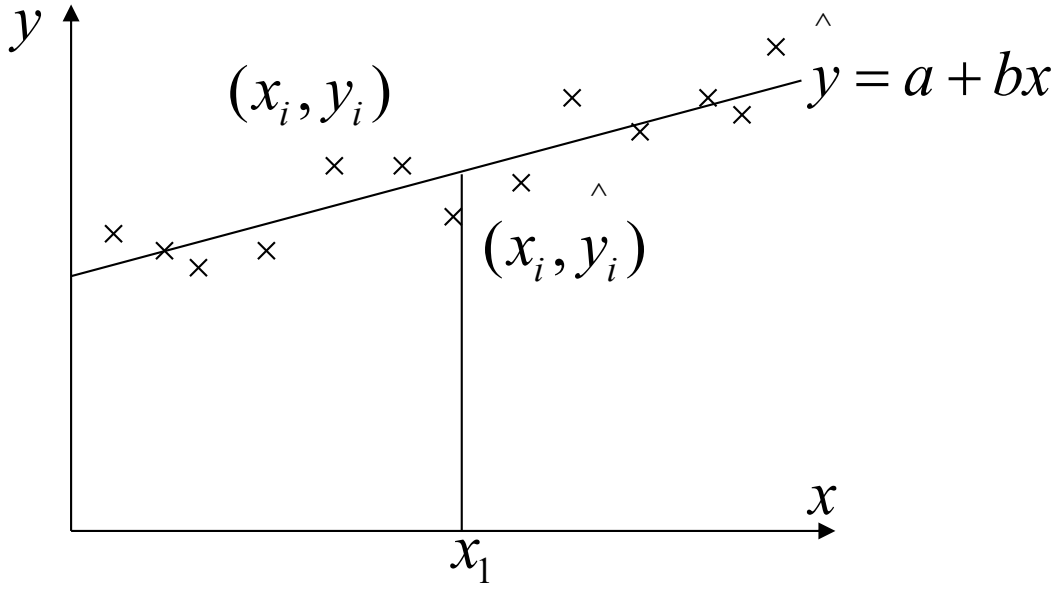
华东师范大学 数据科学与工程学院

提纲

1. 上一讲回顾
2. 文本分类
3. 朴素贝叶斯
4. 文本分类评价

回归(Regression)

- 回归分析：回归分析是处理变量之间相关关系的一种工具，回归的结果可以用于预测或者分类
- 一元线性回归：根据观测点，拟合出一条直线，使得某种损失(如离差平方和)最小

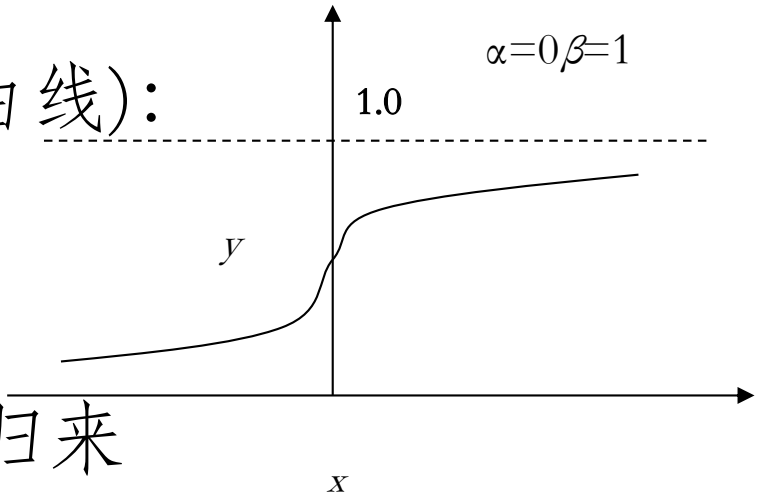


- 多元线性回归：

$$y = \beta_0 + \sum_i \beta_i x_i$$

Logistic 回归

- Logistic回归是一种非线性回归
- Logistic (也叫Sigmoid)函数(*S*型曲线):



$$y = f(x) = \frac{1}{1+e^{-(\alpha+\beta x)}} = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$

- Logistic回归可以转化成线性回归来实现

$$\frac{y}{1-y} = e^{\alpha+\beta x}, \quad \ln\frac{y}{1-y} = \alpha+\beta x$$

Logistic 回归IR模型

- 基本思想：为了求Q和D相关的概率$P(R=1|Q,D)$，通过定义多个特征函数$f_i(Q,D)$，认为$P(R=1|Q,D)$是这些函数的组合。
- Cooper等人提出一种做法*：定义$\log(P/(1-P))$为多个特征函数的线性组合。则P是一个Logistic函数，即：

$$\log\frac{P}{1-P}=\beta_0+\sum_i\beta_i f_i(Q,D)$$

$$P=\frac{1}{1+e^{-\beta_0-\sum_i\beta_i f_i(Q,D)}}$$

*William S. Cooper , Fredric C. Gey , Daniel P. Dabney, Probabilistic retrieval based on staged logistic regression, Proceedings of ACM SIGIR'92, p.198-210, June 21-24, 1992, Copenhagen, Denmark

特征函数f_i的选择

$$X_1 = \frac{1}{M}\sum_{1}^{M}\log QAF_{t_j}$$

$$X_2 = \sqrt{QL}$$

$$X_3 = \frac{1}{M}\sum_{1}^{M}\log DAF_{t_j}$$

$$X_4 = \sqrt{DL}$$

$$X_5 = \frac{1}{M}\sum_{1}^{M}\log IDF_{t_j}$$

$$IDF = \frac{N - n_{t_j}}{n_{t_j}}$$

$$X_6 = \log M$$

Logistic 回归IR模型(续)

- 求解和使用过程：
  - 通过训练集合拟和得到相应系数 $\beta_0 \sim \beta_6$，对于新的文档，代入公式计算得到概率 P
  - *Learning to Rank*中 *Pointwise*方法中的一种
  - 判别式(discriminate)模型
- 优缺点：
  - 优点：直接引入数学工具，形式简洁。
  - 缺点：特征选择非常困难，实验中效果一般。

应用

- 如何应用Logistic 回归?
  - 预测一个人是否喜欢X?
  - 是否喜欢某人?
  - 是否喜欢某电影?
  - 是否喜欢某餐馆?

特征设计

- 问题：是否喜欢某人？ Yes / No
- 特征
  - F1：是不是长时间注视着某人?
  - F2:
  - F3:
  - F4:
  - F5:
  - F6:

$$y = \beta_0 + \sum_i \beta_i x_i$$

$$y = \beta_0 + \beta_1 F_1 + \beta_2 F_2 + \beta_3 F_3 + \beta_4 F_4 + \beta_5 F_5 + \beta_6 F_6$$

训练集表达

$$y = \beta_0 + \sum_i \beta_i x_i$$

$$y = \beta_0 + \beta_1 F_1 + \beta_2 F_2 + \beta_3 F_3 + \beta_4 F_4 + \beta_5 F_5 + \beta_6 F_6$$

- 测试例子：郭靖是否喜欢黄蓉？

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
- 本讲义主要介绍查询似然模型

提纲

1. 上一讲回顾
2. 文本分类
3. 朴素贝叶斯
4. 文本分类评价

文本分类

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

- 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

人的分类

- 给定分类体系，将一个人分到其中一个或者多个类别中的过程。
  - 比如，男/女；教师/学生/工人/。。。

- 按类别数目： binary vs. multi-class
- 按每个人赋予的标签数目： sing label vs. multi label

出勤问题

- 给定分类体系，将一个学生分到其中一个类别：来/没来。

- 按类别数目： binary

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， $<d, c> \in X \times C$

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： $d \in X$

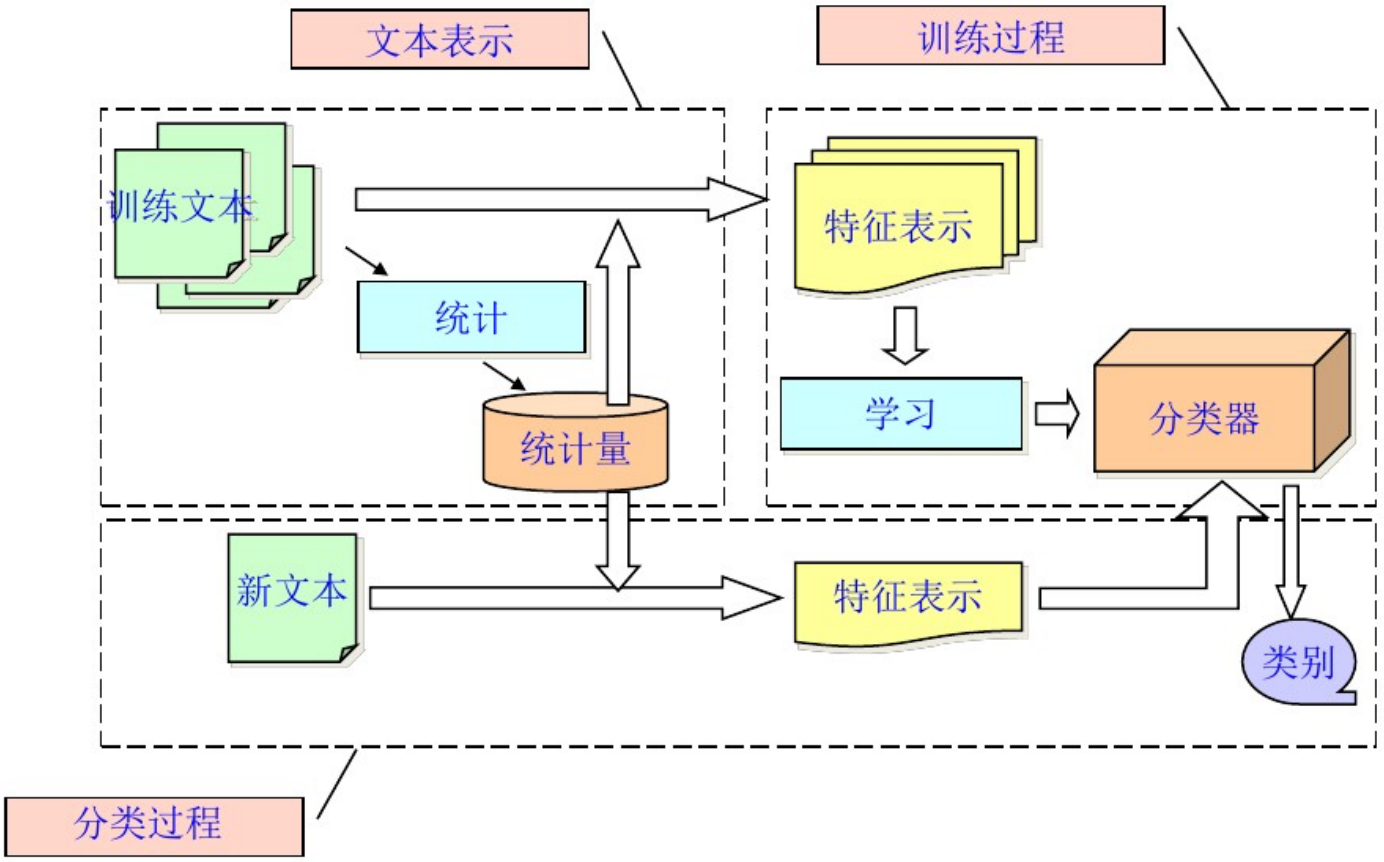
确定： $\Upsilon(d) \in C$,

即确定d最可能属于的类别

文本分类：训练/测试 类别

举例，类比

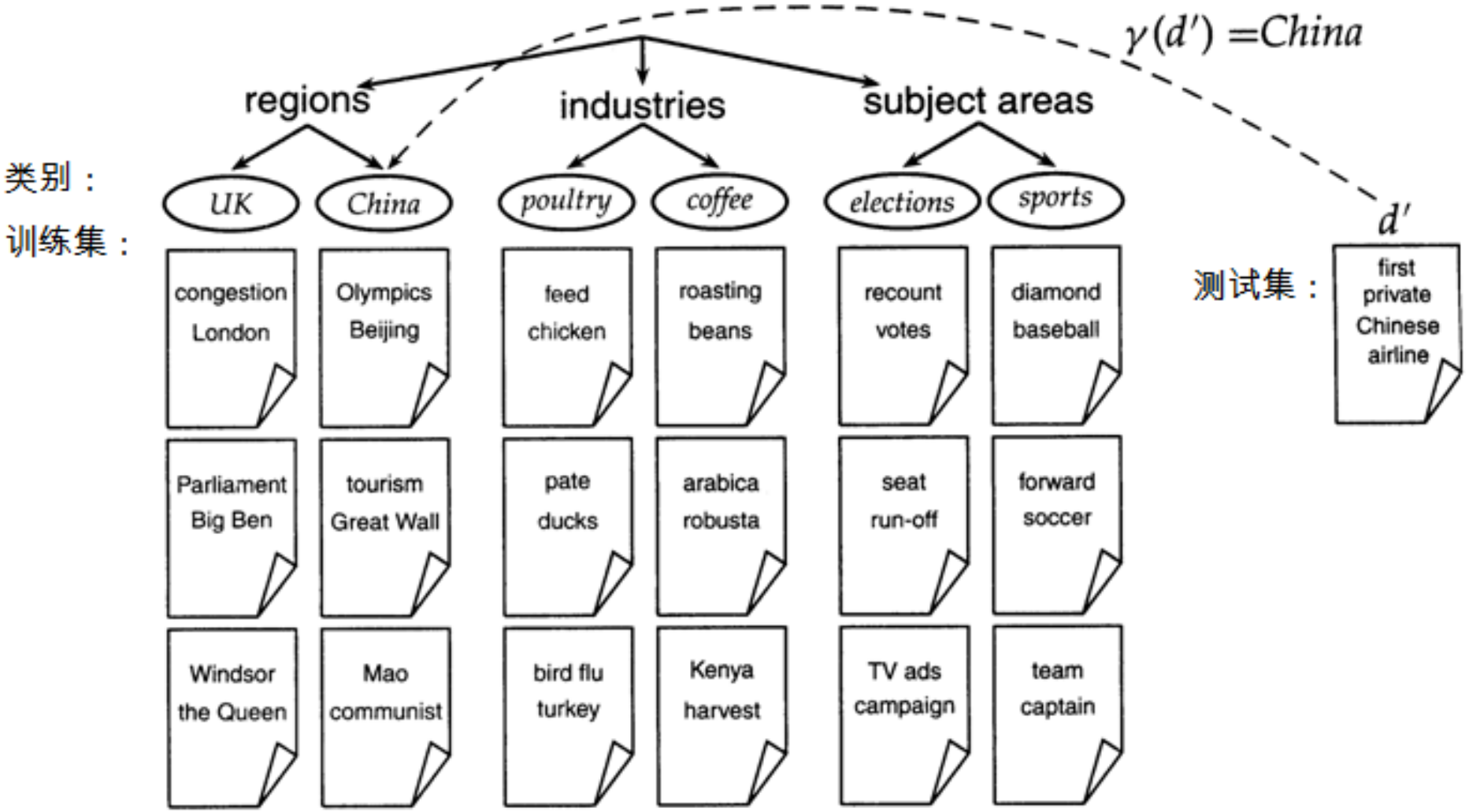
文本分类的过程



自动文本分类方法

- Rocchio方法
- Naïve Bayes
- kNN方法
- 决策树方法decision tree
- Decision Rule Classifier
- The Widrow-Hoff Classifier
- 神经网络方法Neural Networks
- 支持向量机SVM
- 基于投票的方法(voting method)

主题分类


$\gamma(d') = China$
regions
industries
subject areas
类别：
训练集：
UK
China
poultry
coffee
elections
sports
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
d'
测试集：
first
private
Chinese
airline

课堂练习

- 试举出文本分类的应用例子
- *对研究生找导师的Email进行分类：导师是否会回复？*

搜索引擎中的文本分类应用

- 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)

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大的时候，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（用处？）

- 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

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) 通过有监督的学习，得到分类函数Y，然后将其
  (ii) 应用于对新文档的分类
- 一系列分类方法: 朴素贝叶斯, Rocchio, kNN, SVM
- 当然，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

1. 上一讲回顾
2. 文本分类
3. 朴素贝叶斯
4. 文本分类评价

朴素贝叶斯分类器

- 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：

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

- 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
- $P(t_k \mid c)$ 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
- 最佳类别史具有最大后验概率(maximum a posteriori -MAP)的类别 c_{map}:

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

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 1: 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:

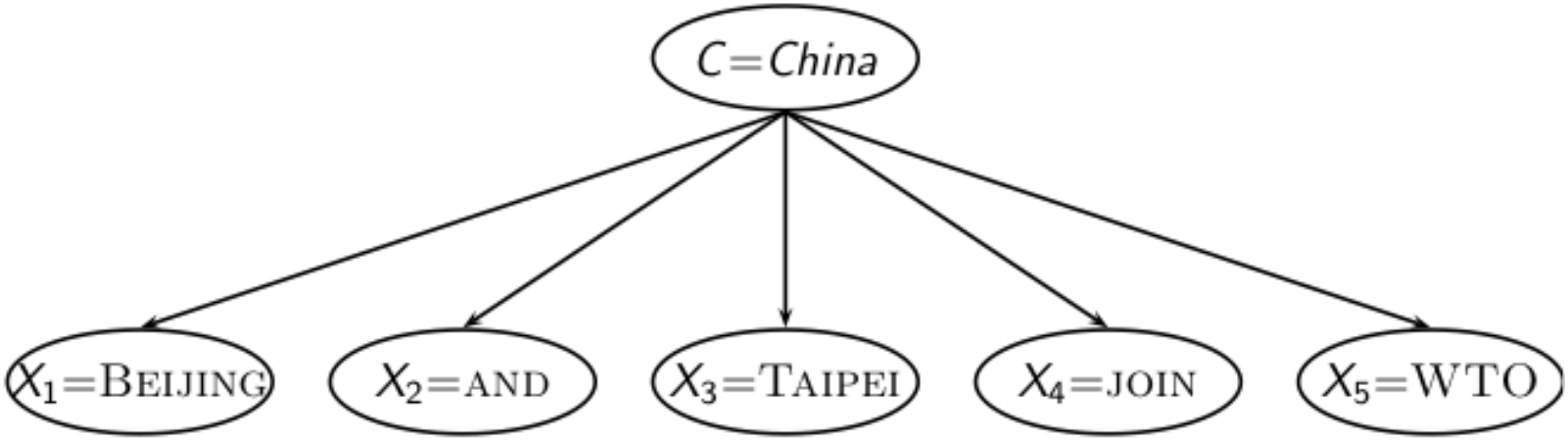
$$\hat{P}(c) = \frac{N_c}{N}$$

- 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)

MLE估计中的问题：零概率问题



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

- 如果 WTO 在训练集中没有出现在类别 China 中:

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

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：

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

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

What to do?

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

$\textsc{ApplyMultinomialNB}(\mathbb{C}, V, prior, condprob, d)$

1 $W \leftarrow \textsc{ExtractTokensFromDoc}(V, d)$
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** $\arg\max_{c \in \mathbb{C}} score[c]$

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

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

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

上述计算中的分母分别是 $(8+6)$ 和 $(3+6)$，这是因为 $text_c$ 和 $text_{\overline{c}}$ 的大小分别是8和3，词汇表大小是6。

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

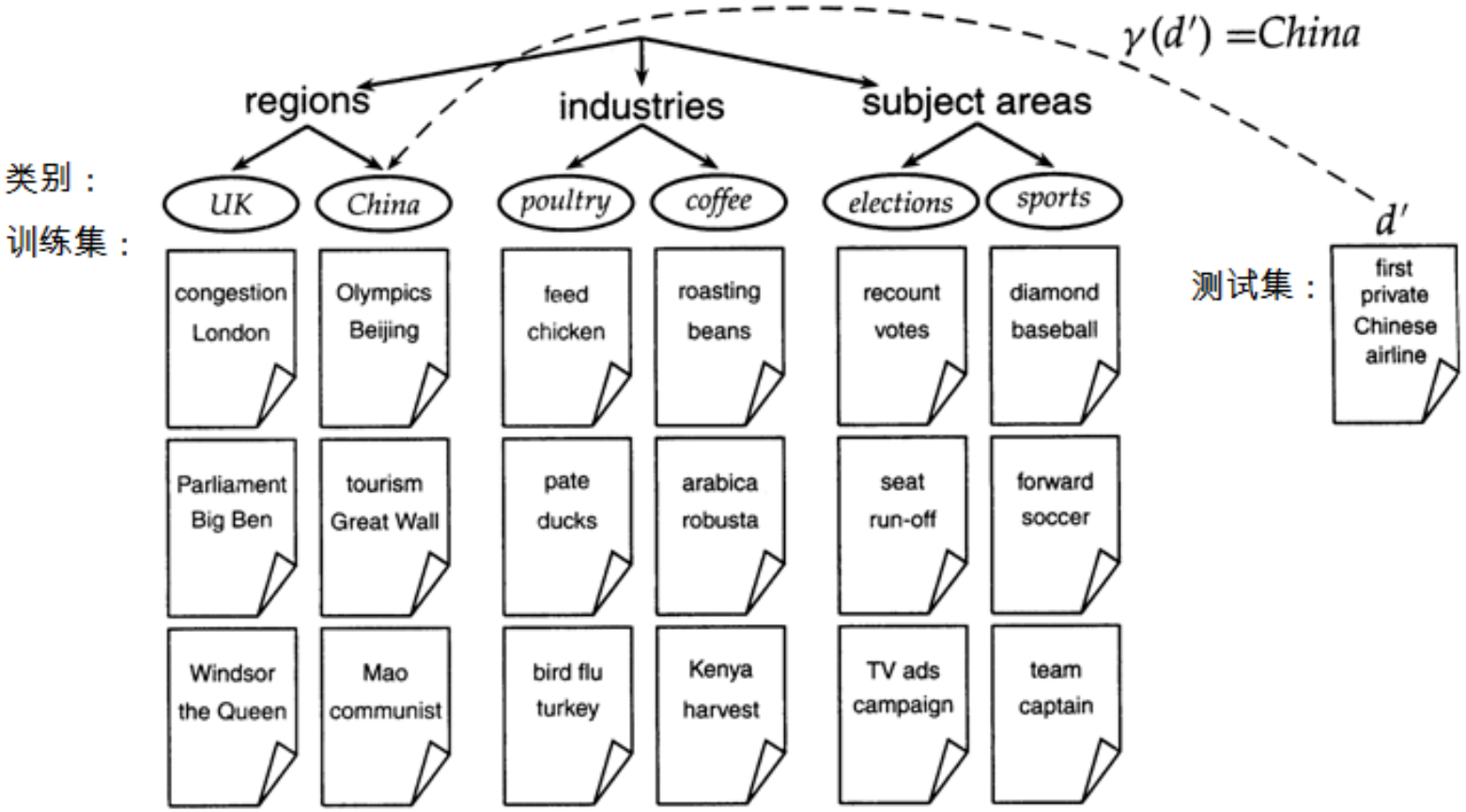
mode	time complexity
training	$\Theta(\lvert\mathbb{D}\rvert L_{ave} + \lvert\mathbb{C}\rvert\lvert V\rvert)$
testing	$\Theta(L_a + \lvert\mathbb{C}\rvert M_a) = \Theta(\lvert\mathbb{C}\rvert M_a)$

- 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
- $\Theta(\lvert\mathbb{D}\rvert L_{ave})$ 是统计所有词语的出现次数的时间
- $\Theta(\lvert\mathbb{C}\rvert\lvert V\rvert)$ 是从上述次数计算参数的时间
- 通常来说: $\lvert\mathbb{C}\rvert\lvert V\rvert < \lvert\mathbb{D}\rvert L_{ave}$
- 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。这是最优的

提纲

1. 上一讲回顾
2. 文本分类
3. 朴素贝叶斯
4. 文本分类评价

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

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

- F1 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)
  - 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

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).

biendata competitions

English version 登录 注册

问题

- 分类问题：判断是否营销意图
- 训练数据Train：
  - 标注数据（数据集规模为5万条新闻和35万张新闻配图，标注为有营销意图的新闻、文本片段和配图）
  - 未标注数据（数据规模为20万条新闻和100万张新闻配图）
- 测试数据Test：
  - 数据集规模为1万条新闻和7万张新闻配图
  - 标准答案未知，在测试平台服务器上

评价标准

- F-Measure
  - P为营销识别的准确率，R为营销识别的召回率。

$$\mathrm{F}-Measure=\frac{2PR}{P+R}$$

  - 随数据集公开评测程序的Python版本。

怎么保证能经常测试自己的系统?

- 规则：随时上传预测结果，一天不能超过5次
- 解决方案:
  - 按照规则，来不及就等明天 ➔ 太慢耽误事
  - 开个小号，偷偷的提交 ➔ X
  - 从训练集随机选20%数据作为内部评测数据(Dev)

基本步骤

- 选择自己的方法
- 在Dev测试自己系统的效果
  - 在Dev取得不错的效果
- 在Test测试自己系统，提交给平台

- 先提交个结果，得到一个分数
  - 非常重要，代表你已经了解评测流程了

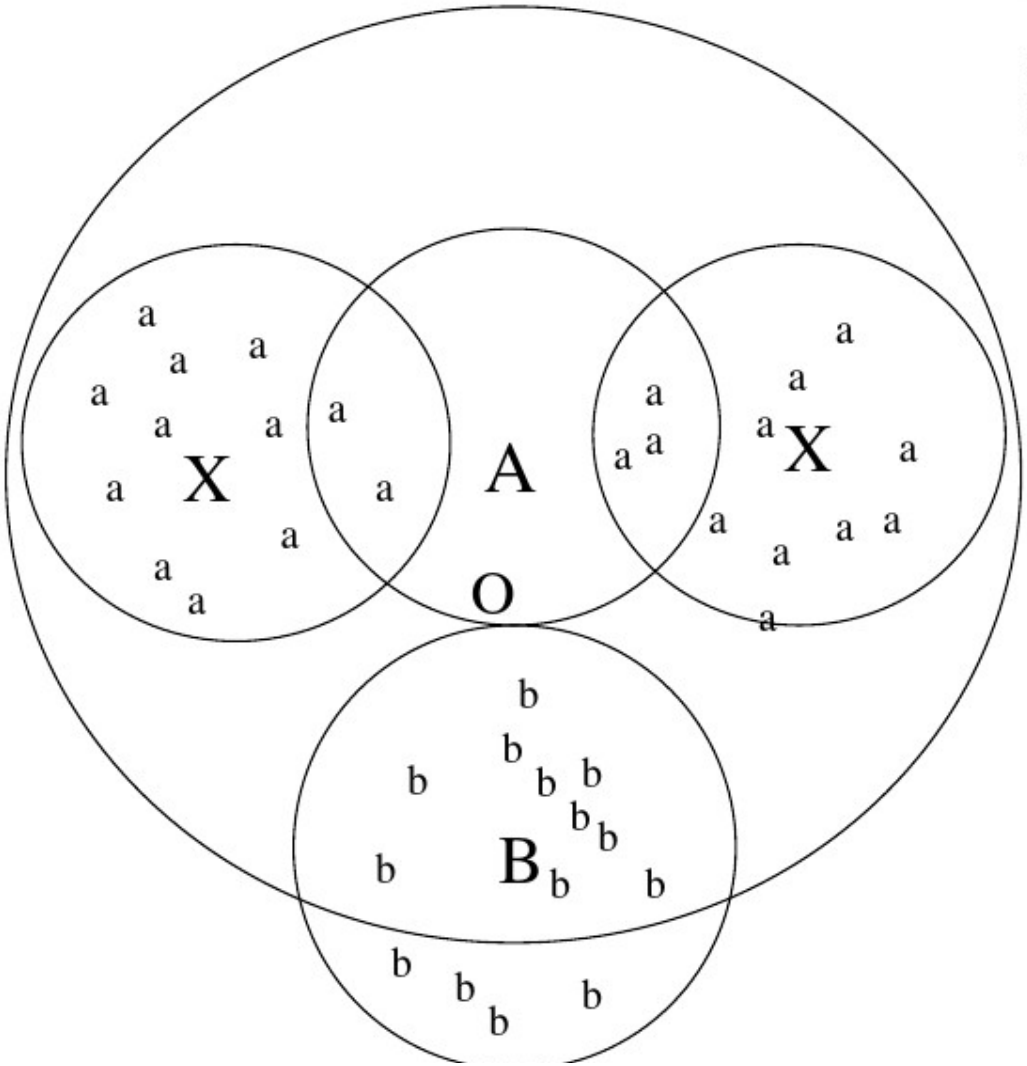
Rocchio分类: 基本思想

- 计算每个类的中心向量
  - 中心向量是所有文档向量的算术平均
- 将每篇测试文档分到离它最近的那个中心向量

Rocchio性质

- Rocchio简单地将每个类别表示成其中心向量
  - 中心向量可以看成类别的原型(prototype)
- 分类基于文档向量到原型的相似度或聚类来进行
- 并不保证分类结果与训练集一致，即得到分类器后，不能保证训练集中的文档能否正确分类

Rocchio不能正确处理非凸、多模式类别问题



课堂练习: 对于左图的A/B分类问题，为什么Rocchio方法难以有效处理?

- A 是所有a的中心向量，B是所有b的中心向量
- 点o 离A更近
- 但是o更适合于b类
- A 是一个有两个原型多模式类别
- 但是，在Rocchio算法中，每个类别只有一个原型

kNN分类器

- kNN 是另外一种基于向量空间的分类方法
- 该方法非常简单，也容易实现
- 在大多数情况下，kNN的效果比朴素贝叶斯和Rocchio要好
- 如果你急切需要一种精度很高分类器并很快投入运行 . .
- . . . 如果你不是特别关注效率 . . .
- . . . 那么就使用kNN

kNN分类

- kNN = k nearest neighbors，k近邻
- $k = 1$ 情况下的kNN (最近邻): 将每篇测试文档分给训练集中离它最近的那篇文档所属的类别。
- 1NN 不很鲁棒—一篇文档可能会分错类或者这篇文档本身就很反常
- $k > 1$ 情况下的kNN: 将每篇测试文档分到训练集中离它最近的k篇文档所属类别中最多的那个类别
- kNN的基本原理: 邻近性假设
  - 我们期望一篇测试文档d与训练集中d周围邻域文档的类别标签一样。

kNN: 讨论

- 不需要训练过程
  - 但是，文档的线性预处理过程和朴素贝叶斯的训练开销相当
  - 对于训练集来说我们一般都要进行预处理，因此现实当中kNN的训练时间是线性的。
- 当训练集非常大的时候，kNN分类的精度很高
- 如果训练集很小， kNN可能效果很差。
- kNN倾向于大类，可以将相似度考虑在内来缓解这个问题。

线性分类器

- 定义：
  - 线性分类器计算特征值的一个线性加权和 $\sum_i w_i x_i$
  - 决策规则：$\sum_i w_i x_i > \theta$?
  - 其中，θ 是一个参数
- 首先，我们仅考虑二元分类器
- 从几何上说，二元分类器相当于二维平面上的一条直线、三维空间中的一个平面或者更高维下的超平面，称为分类面
- 基于训练集来寻找该分类面
- 寻找分类面的方法：感知机(Perceptron)、 Rocchio, Naïve Bayes – 我们将解释为什么后两种方法也是二元分类器
- 假设：分类是线性可分的

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