

# THE TASK OF TEXT CLASSIFICATION

## 文本分类

- 文本分类的基本概念
- 文本表示
- 特征选择
- 分类器设计

### Is this spam?

Subject: **Important notice!**  
From: Stanford University <newsforum@stanford.edu>  
Date: October 28, 2011 12:34:16 PM PDT  
To: undisclosed-recipients;;

Greats News!

You can now access the latest news by using the link below to login to Stanford University News Forum.

<http://www.123contactform.com/contact-form-StanfordNew1-236335.html>

Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information about the new services.

© Stanford University. All Rights Reserved.

### Who wrote which Federalist papers?

1787-8: anonymous essays try to convince New York to ratify U.S Constitution: Jay, Madison, Hamilton.

Authorship of 12 of the letters in dispute

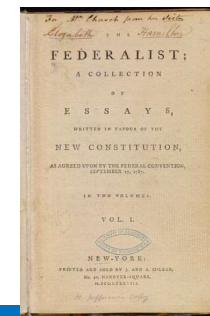
1963: solved by Mosteller and Wallace using Bayesian methods



James Madison



Alexander Hamilton







## Male or female author?

1. By 1925 present-day Vietnam was divided into three parts under French colonial rule. The southern region embracing Saigon and the Mekong delta was the colony of Cochinchina; the central area with its imperial capital at Hue was the protectorate of Annam...
2. Clara never failed to be astonished by the extraordinary felicity of her own name. She found it hard to trust herself to the mercy of fate, which had managed over the years to convert her greatest shame into one of her greatest assets...

S. Argamon, M. Koppel, J. Fine, A. R. Shimon, 2003. "Gender, Genre, and Writing Style in Formal Written Texts," Text, volume 23, number 3, pp. 321–346

## Positive or negative movie review?

-  unbelievably disappointing
-  Full of zany characters and richly applied satire, and some great plot twists
-  this is the greatest screwball comedy ever filmed
-  It was pathetic. The worst part about it was the boxing scenes.

## What is the subject of this article?

### MEDLINE Article



### MeSH Subject Category Hierarchy

Antagonists and Inhibitors  
Blood Supply  
Chemistry  
Drug Therapy  
Embryology  
Epidemiology  
...

?

## Text Classification

Assigning subject categories, topics, or genres  
Spam detection  
Authorship identification  
Age/gender identification  
Language Identification  
Sentiment analysis

## Text Classification: definition

### Input:

- a document  $d$
- a fixed set of classes  $C = \{c_1, c_2, \dots, c_j\}$

Output: a predicted class  $c \in C$

## Classification Methods: Supervised Machine Learning

### Input:

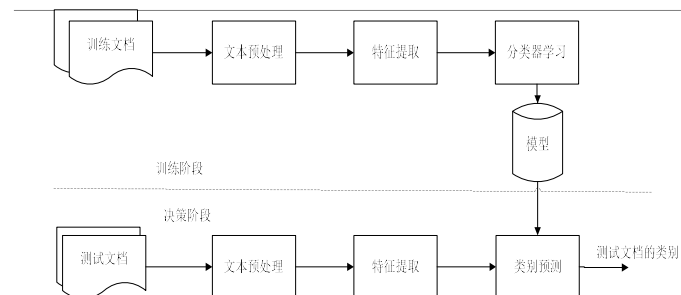
- a document  $d$
- a fixed set of classes  $C = \{c_1, c_2, \dots, c_j\}$
- A training set of  $m$  hand-labeled documents  $(d_1, c_1), \dots, (d_m, c_m)$

### Output:

- a learned classifier  $\gamma: d \rightarrow c$

10

## 文本分类



11

## 文本预处理

去掉Tag标签，如：HTML文档

停用词去除

(英文) 词根还原

(中文) 分词、词性标注、短语识别\.....

数据清洗：去掉不合适的噪声文档或文档内垃圾数据

12

## The bag of words representation

$Y($

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.

$)=C$



## The bag of words representation: using a subset of words

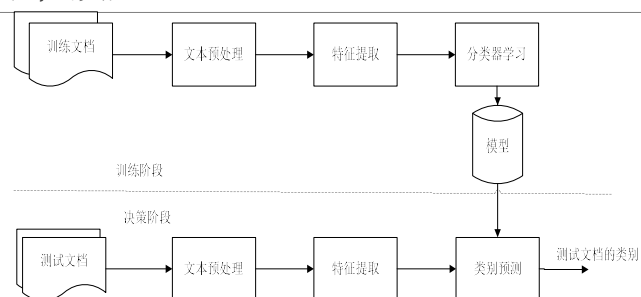
$Y($

x love xxxxxxxxxxxxxxxx sweet  
xxxxxxxx satirical xxxxxxxxxxxx  
xxxxxxxx great xxxxxxxx  
xxxxxxxxxxxxxxxxxxxxxxxx fun xxxx  
xxxxxxxxxxxxxxxx whimsical xxxx  
romantic xxxx laughing  
xxxxxxxxxxxxxxxxxxxxxxxxxxxx  
xxxxxxxxxxxxxxxx recommend xxxxx  
xxxxxxxxxxxxxxxxxxxxxxxxxxxx  
xx several xxxxxxxxxxxxxxxx  
xxxx happy xxxxxxxx again  
xxxxxxxxxxxxxxxxxxxxxxxxxxxx  
xxxxxxxxxxxxxxxxxxxxxxxx

$)=C$



## 文本预处理



## 文本预处理

去掉Tag标签，如：HTML文档

停用词去除

（英文）词根还原

（中文）分词、词性标注、短语识别\.....

词频统计

• TF: 词频

• DF: 文档频率

数据清洗：去掉不合适的噪声文档或文档内垃圾数据

## Classification Methods: Supervised Machine Learning

Any kind of classifier

- Naïve Bayes
- Logistic regression
- Support-vector machines
- k-Nearest Neighbors
- ...

## 向量空间模型

文档表示

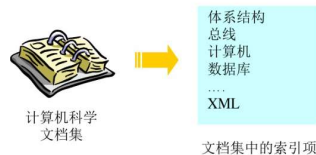
- 一个文档被表示为一个 $t$ 维的向量形式:

$$d_j = (w_{1j}, \dots, w_{tj})$$

- $w_{ij}$  表示文档  $d_j$  的第  $i$  个关键词的权重。

## 向量空间模型

- 若干独立的词项被选作索引项(*index terms*) or 词表 *vocabulary*
- 索引项代表了一个应用中的重要词项
  - 计算机科学图书馆中的索引项应该是哪些呢?



## 向量空间模型-TF

通常的权重计算方法主要有2种: 词频 (TF) 和TFIDF。

TF

- 文档中的每个单词的权重取决于该单词在文档中出现的次数。
- 简单的方式是将权重设置为  $t$  在文档  $d$  中的出现次数。这种权重计算的结果称为词项频率 (*term frequency*)，记为  $tf_{d,t}$ ，其中的两个下标分别对应词项  $t$  和文档  $d$ 。

## 向量空间模型-TF

d1: 爱吃苹果的人也爱玩苹果手机。

d2: 苹果手机比华为手机贵

d3: 今年苹果丰收

	爱	吃	苹果	玩	手机	比	华为	贵	今年	丰收
d1	2	1	2	1	1	0	0	0	0	0
d2	0	0	1	0	2	1	1	1	0	0
d3	0	0	1	0	0	0	0	0	1	1

## 向量空间模型-TF

TF函数背后的含义是在同一个文档中多次出现了单词比少数几次出现的单词的权重更高。但是它的值并不随着词频的增长而线性增长。

$$TF = \begin{cases} \log(f_{t,d}) & \text{如果 } f_{t,d} > 0 \\ 0 & \text{其他} \end{cases}$$

TF归一化

$$ntf_{t,d} = \alpha + (1 - \alpha) \frac{tf_{t,d}}{tf_{\max}(d)}$$

## 向量空间模型-IDF

文档频率：我们将文档集中有多少个文档中出现了单词 $t$ ，定义为文档频率，记作 $df_t$

d1: 爱吃苹果的人也爱玩苹果手机。

d2: 苹果手机比华为手机贵

d3: 今年苹果丰收

逆文档频率 (IDF)

$$idf_t = \log \frac{N}{df_t}$$

单词	$df_t$
苹果	3
手机	2
华为	1

单词	$idf_t$
苹果	0
手机	0.176
华为	0.477

某个单词在多个文档中出现次数较多，那么区分度就降低，重要度下降。

## 向量空间模型- TFIDF

### TFIDF权重机制

- 对于每一篇文档中的每个词项，可以将其的 $tf$ 值与 $idf$ 值组合在一起形成最终的权重。

$$tf-idf_{t,d} = tf_{t,d} \times idf_t$$

$$tf-idf_{t,d} = \log(f_{t,d}) \times idf_t$$

## 向量空间模型- TFIDF

d1: 爱吃苹果的人也爱玩苹果手机。

d2: 苹果手机比华为手机贵

d3: 今年苹果丰收

	爱	吃	苹果	玩	手机	比	华为	贵	今年	丰收
d1	2	1	2	1	1	0	0	0	0	0
d2	0	0	1	0	2	1	1	1	0	0
d3	0	0	1	0	0	0	0	0	1	1

	爱	吃	苹果	玩	手机	比	华为	贵	今年	丰收
D1	0.477	0.477	0	0.477	0.176	0	0	0	0	0
D2	0	0	0	0	0.352	0.477	0.477	0.477	0	0
D3	0	0	0	0	0	0	0	0	0.477	0.477

单词	idf <sub>t</sub>
爱	0.477
吃	0.477
苹果	0
玩	0.477
手机	0.176
比	0.477
华为	0.477
贵	0.477
今年	0.477
丰收	0.477

## 向量空间模型- TFIDF

d中某个单词t的词频很高，但是这个单词在文档集中的其他文档中很少出现，那么单词t的权重会较高。

如果d中单词的词频很高，但是这个单词在其他文档中也经常出现，那么单词t的权重会很低，因为该单词没有区分能力

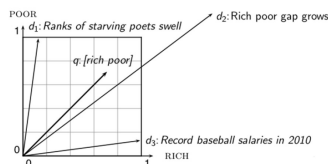
如果d中单词的词频很低，但是这个单词在文档集合D中的其他文档中很少出现，那么单词t的权重也会很低，因为该单词t对文档d的内容表现能力较差。

如果d中的单词词频很低，并且这个单词在文档集合D中的其他文档中经常出现，那么单词t的权重也会很低，因为该单词t不仅对文档d的内容表现能力较差，而且该单词的区分能力也差。

## 向量空间模型-余弦

采用余弦相似度计算2个文档的相似度

$$\text{sim}(d, q) = \frac{q \cdot d}{\|q\| \|d\|}$$



## 余弦相似度计算示例

计算下面三部小说相似度

term	SaS	PaP	WH
affection	115	58	20
jealous	10	7	11
gossip	2	0	6
wuthering	0	0	38

SaS: Sense and

Sensibility (理智与情感)

•PaP: Pride and

Prejudice (傲慢与偏见)

•WH: Wuthering

Heights? (呼啸山庄)

**词项频率 (tf)**

(暂不考虑idf权值)

### 词项频率取对数

term	SaS	PaP	WH
affection	3.06	2.76	2.30
jealous	2.00	1.85	2.04
gossip	1.30	0	1.78
wuthering	0	0	2.58

### 长度归一化

term	SaS	PaP	WH
affection	0.789	0.832	0.524
jealous	0.515	0.555	0.465
gossip	0.335	0	0.405
wuthering g	0	0	0.588

$$\cos(\text{SaS}, \text{PaP}) \approx? \quad 0.789 \times 0.832 + 0.515 \times 0.555 + 0.335 \times 0.0 + 0.0 \times 0.0 \approx 0.94$$

$\cos(\text{SaS}, \text{WH}) \approx ? \quad 0.79$

$\cos(\text{PaP, WH}) \approx ?$  0.69

## 特征选择

- 文档频率DF
- 信息增益IG
- 互信息MI
- $\chi^2$ 统计量 (CHI-2)

中	文	停	用	词	表
但是	各自	即便	靠	哪个	
并且	给	即或	咳	哪	
并且	根据	即或	哪	哪	
不比	缘故	即或	可见	哪	
不成	故	即或	可见	哪	
不	当然	几	可以	哪	
不	的话	几	况且	哪	
不独	关于	既	啦	那	
不管	等等	管	已	那边	
不光	地	归	既然	那个	
不过	第	果然	来着	那个	
不	对	真	高	那个	
不	听	过	加	那里	
不仅	过	之	例如	那里	
			哩		

通过词频分析筛选：

1、 $N_{sd} \geq 70\%N_d$  (包含该词的文档数超过总文档数的70%)

2、Fs in TopN (该词出现的频率在前N个最大的频率中, N的取值是样本大小而定)

[返回](#)

## 互信息

一个常用的特征选择方法是计算词项t和类别c的MI (expected mutual information, 期望互信息)

$$I(t_i, c_p) = \log \left( \frac{p(t_i, c_p)}{p(t_i)p(c_p)} \right) = \log \left[ \frac{\frac{n_{ip}}{N_t}}{\frac{n_i}{N_t} \times \frac{n_p}{N_t}} \right]$$

$$MI(t_i, C) = \sum_{p=1}^L p(c_p) I(t_i, c_p) = \sum_{p=1}^L \frac{n_p}{N_t} \log \left[ \frac{\frac{n_{ip}}{N_t}}{\frac{n_i}{N_t} \times \frac{n_p}{N_t}} \right]$$



UK		China		poultry	
london	0.1925	china	0.0997	poultry	0.0013
uk	0.0755	chinese	0.0523	meat	0.0008
british	0.0596	beijing	0.0444	chicken	0.0006
stg	0.0555	yuan	0.0344	agriculture	0.0005
britain	0.0469	shanghai	0.0292	avian	0.0004
plc	0.0357	hong	0.0198	broiler	0.0003
england	0.0238	kong	0.0195	veterinary	0.0003
pence	0.0212	xinhua	0.0155	birds	0.0003
pounds	0.0149	province	0.0117	inspection	0.0003
english	0.0126	taiwan	0.0108	pathogenic	0.0003

coffee		elections		sports	
coffee	0.0111	election	0.0519	soccer	0.0681
bags	0.0042	elections	0.0342	cup	0.0515
growers	0.0025	polls	0.0339	match	0.0441
kg	0.0019	voters	0.0315	matches	0.0408
colombia	0.0018	party	0.0303	played	0.0388
brazil	0.0016	vote	0.0299	league	0.0386
export	0.0014	poll	0.0225	beat	0.0301
exporters	0.0013	candidate	0.0202	game	0.0299
exports	0.0013	campaign	0.0202	games	0.0284
crop	0.0012	democratic	0.0198	team	0.0264

图 13-7 Reuters-RCV1 语料 6 个类别中互信息值较高的部分词项列表

## $\chi^2$ 统计量 (CHI-2)

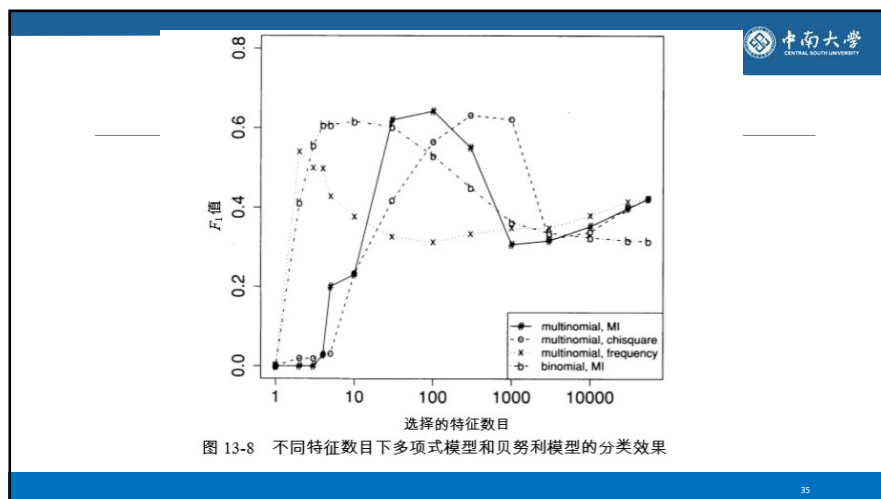
卡方校验是对索引项 $t_i$ 和类别 $c_p$ 独立性的缺失所做的度量。

$$\chi^2(t_i, c_p) = \frac{N_t(N_t n_{ip} - n_p n_i)^2}{n_p n_i (N_t - n_p)(N_t - n_i)}$$

$$\chi_{avg}^2(t_i) = \sum_{p=1}^k p(c_p) \chi^2(t_i, c_p) \quad (9-16)$$

$$\chi_{max}^2(t_i) = \max_{p=1} \chi^2(t_i, c_p)$$

<https://scikit-learn.org/stable/>  
[https://scikit-learn.org/stable/modules/feature\\_selection.html#feature-selection](https://scikit-learn.org/stable/modules/feature_selection.html#feature-selection)



## K近邻算法-KNN

- 基本思想是：
  - 在给定的新文本后，考虑在训练文本集中与该新文本距离最近(最相似)的K篇文本
  - 根据这K篇文本所属的类别判定新文本所属的类别

## K近邻算法-KNN

- 具体的算法步骤：
  - 根据特征项集合重新描述训练文本向量
  - 在新文本到达后，根据特征词，确定新文本的向量表示
  - 在训练文本集中选出与新文本最相似的K个文本，计算公式为：

$$sim(d_i, d_j) = \frac{\sum_{k=1}^M w_{ik} \times w_{jk}}{\sqrt{(\sum_{k=1}^M w_{ik}^2)(\sum_{k=1}^M w_{jk}^2)}}$$

其中，K值的确定目前并没有很好的方法，一般先定一个初始值，然后根据试验测试的结果调整K值，一般初始值定在几百到几千之间

37

## K近邻算法-KNN

- 在新文本的k个邻居中，依次计算每类的权重，计算公式如下：

$$p(\vec{x}, c_j) = \sum_{\vec{d}_i \in KNN} sim(\vec{x}, \vec{d}_i) y(\vec{d}_i, c_j)$$

其中， $\vec{x}$ 为新文本的特征向量， $sim(\vec{x}, \vec{d}_i)$ 为相似度计算公式，与上一步骤的计算公式相同，而  $y(\vec{d}_i, c_j)$  为类别属性函数，即如果  $\vec{d}_i$  属于类  $c_j$ ，那么函数值为1，否则为0；

- 比较每类的权重，将文本分到权重最大的那个类别中

38

## 基于贝叶斯的文本分类

$$c^* = \arg \max_{c_i \in \{c_1, c_2, \dots, c_k\}} p(c_i | d)$$

判断决策：  $p(c_1 | d) > p(c_2 | d)$ , 那么  $d \in c_1$

$$p(c_i | d_j) = \frac{p(c_i) p(d_j | c_i)}{p(d_j)} \propto p(c_i) p(d_j | c_i)$$

39

## 基于贝叶斯的文本分类

假设文档  $d_j$  由索引项  $\{t_1, \dots, t_n\}$ ，那么：

$$p(d_j | c_i) = \prod_{k=1}^n p(t_k | c_i)$$

朴素贝叶斯分类器的目标函数为：

$$c^* = \arg \max_{c_i \in \{c_1, c_2, \dots, c_k\}} \left[ \log p(c_i) + \sum_{k=1}^n \log p(t_k | c_i) \right]$$

40

## 基于贝叶斯的文本分类

训练集用于估计参数

MLE 估计下的类别先验概率为:

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

条件概率 $P(t|c)$ 的估计值为 $t$ 在 $c$ 类文档中出现的相对频率:

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

41

## 基于贝叶斯的文本分类

如果在训练集上, WTO仅仅在China类文档中出现, 那么对于其他类(如UK), 采用MLE估计的概率值就会为0, 即:

$$P(WTO|UK) = 0$$

假定有一篇单句文档为Britain is a member of WTO, 那么按照公式(13-2)来计算其属于UK类的条件概率值就为0

平滑:

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

42

## 基于贝叶斯的文本分类

	测试集ID	文档中的词	属于c=“中国”类?
训练集	1	中国 北京 中国	是
	2	中国 中国 上海	是
	3	中国 澳门	是
测试集	4	东京 日本 中国	否
	5	中国 中国 中国 东京 日本	?

$$p(\text{“中国”} | c) = (5 + 1) / (8 + 6) = 3/7$$

$$p(\text{“东京”} | c) = p(\text{“日本”} | c) = (0 + 1) / (8 + 6) = 1/14$$

$$p(\text{“中国”} | \bar{c}) = (1 + 1) / (3 + 6) = 2/9$$

$$p(\text{“东京”} | \bar{c}) = p(\text{“日本”} | \bar{c}) = (1 + 1) / (3 + 6) = 2/9$$

43

## 基于贝叶斯的文本分类

测试文档:

$$p(c|d_5) \propto 3/4 \cdot \left(\frac{3}{7}\right)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003$$

$$p(\bar{c}|d_5) \propto 1/4 \cdot \left(\frac{2}{9}\right)^3 \cdot 2/9 \cdot 2/9 \approx 0.0001$$

因此文档 $d_5$ 属于类别 $c$ =“中国”。

44

建立 NB 分类器有两种不同的方法。

- 一种是上节介绍的基于多项式的方法，它基于一个生成模型：在文档的每个位置上生成词表中的一个词项。
- 另外一种方法是多元贝努利模型（multivariate Bernoulli model）或者直接称为贝努利模型（Bernoulli model）。它基于二值独立模型：对于词汇表中的每个词项都对应一个二值变量，1和0分别表示词项在文档中出现和不出现。

区别是：p(t|c)的计算方法

## 伯努利模型

	测试集ID	文档中的词	属于c=“中国”类?
训练集	1	中国 北京 中国	是
	2	中国 中国 上海	是
	3	中国 澳门	是
	4	东京 日本 中国	否
测试集	5	中国 中国 中国 东京 日本	?

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{i \in V} (T_{ci} + 1)} = \frac{T_{ct} + 1}{\sum_{i \in V} T_{ci} + B} \quad B=2$$

$$p(\text{中国} | c) = (3+1)/(3+2) \quad p(\text{中国} | \neg c) = (1+1)/(1+2)$$

$$p(\text{日本} | c) = p(0+1)/(3+2) \quad p(\text{日本} | \neg c) = (1+1)/(1+2)$$

$$\dots \quad P(c|d_5) = ? \quad P(\neg c|d_5) = ?$$

	测试集ID	文档中的词	属于c=“中国”类?
训练集	1	中国 北京 中国	是
	2	中国 中国 上海	是
	3	中国 澳门	是
	4	东京 日本 中国	否
测试集	5	中国 中国 中国 东京 日本	?

$$P(c|d_5) \propto p(\text{中国} | c) p(\text{日本} | c) p(\text{东京} | c) (1-p(\text{北京} | c)) (1-p(\text{上海} | c)) (1-p(\text{澳门} | c))$$

$$= 3/4 * 4/5 * 1/5 * 1/5 * (1-2/5) * (1-2/5) * (1-2/5)$$