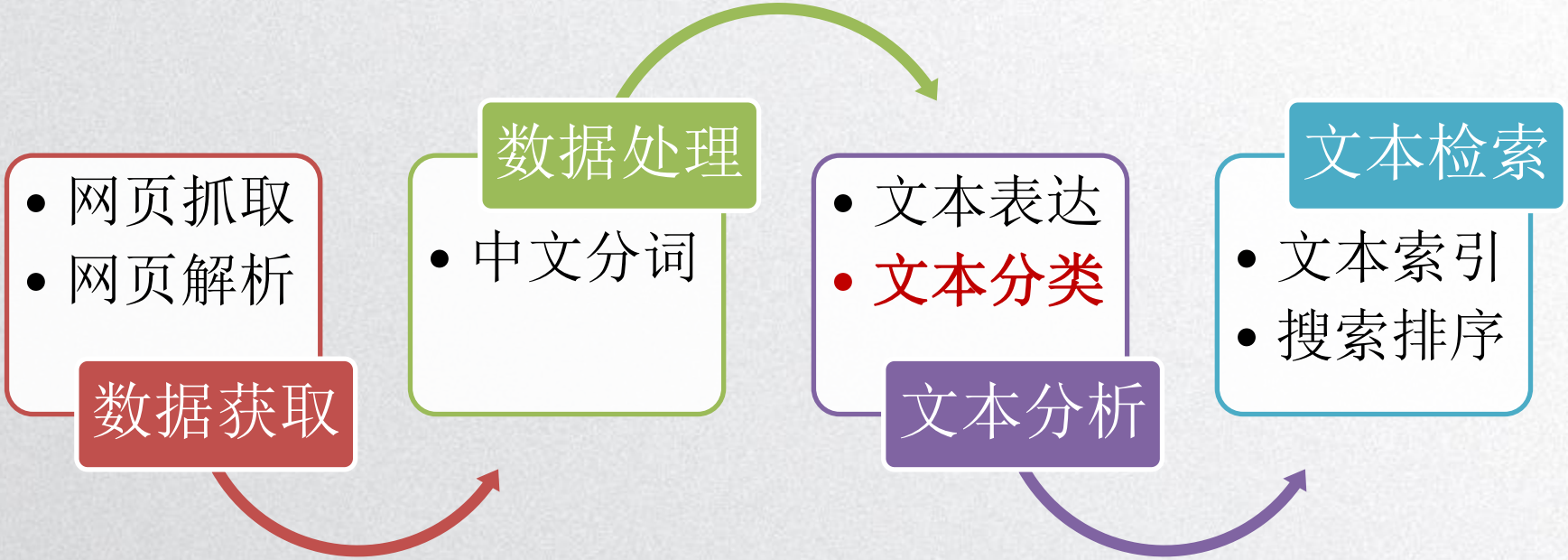
文本分类

覃雄派

文本模块涉及的内容



文本分类

提纲

- 文本分类应用
- 文本分类形式化定义
- 文本分类方法介绍
- 贝叶斯公式介绍
- 基于朴素贝叶斯分类器的文本分类
- 一个实例
- 另一个实例（课堂练习）
- 朴素贝叶斯的问题与效果
- NLTK文本分类实例

文本分类

- 文本分类应用：垃圾信息过滤

```
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
=================================================
```

垃圾Email

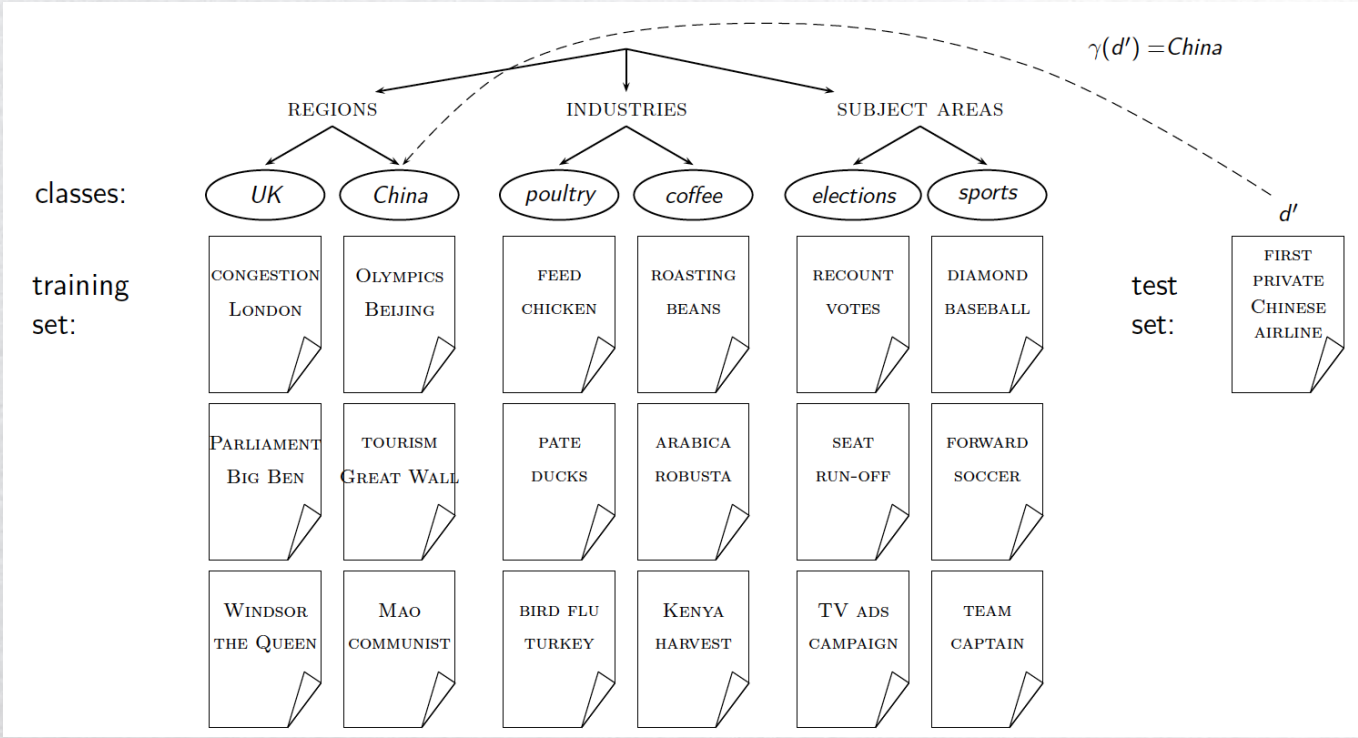
标签	短信内容
0	商业秘密的秘密性那是维系其商业价值和垄断地位的前提条件之一
1	南口阿玛施新春第一批限量春装到店啦　春暖花开淑女裙、冰蓝色公主衫　气质粉小西装、冰丝女王长半裙
0	带给我们大常州一场壮观的视觉盛宴
0	有原因不明的泌尿系统结石等
0	23年从盐城拉回来的麻麻的嫁妆
1	感谢致电杭州萧山全金釜韩国烧烤店，本店位于金城路xxx号。韩式烧烤等，价格实惠、欢迎惠顾【全金釜韩国烧烤店】
0	这款UVe智能杀菌机器人是扫地机的最佳伴侣
1	一次价值xxx元王牌项目；可充值xxx元店内项目卡一张；可以参与V动好生活百分百抽奖机会一次！预约电话：xxxxxxxxxx
0	此类皮肤特别容易招惹粉刺、黑头等
1	(长期诚信在本市作各类资格职称（以及印 /章、牌、 ……等。祥：ｘｘｘｘｘｘｘｘｘｘ李伟%

垃圾短信

文本分类

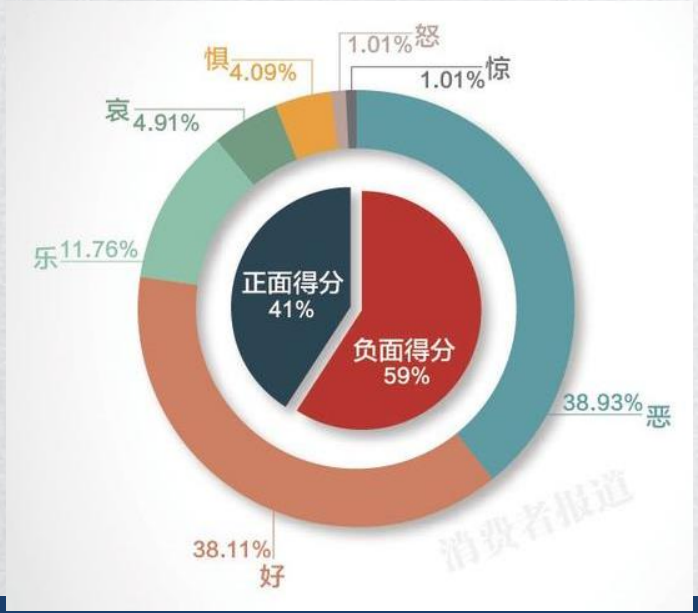
poultry 英 ['pəʊltri] 美 ['poʊltri]
n. 家禽; 禽的肉;

- 文本分类应用：话题分类(注意不是话题建模)



文本分类

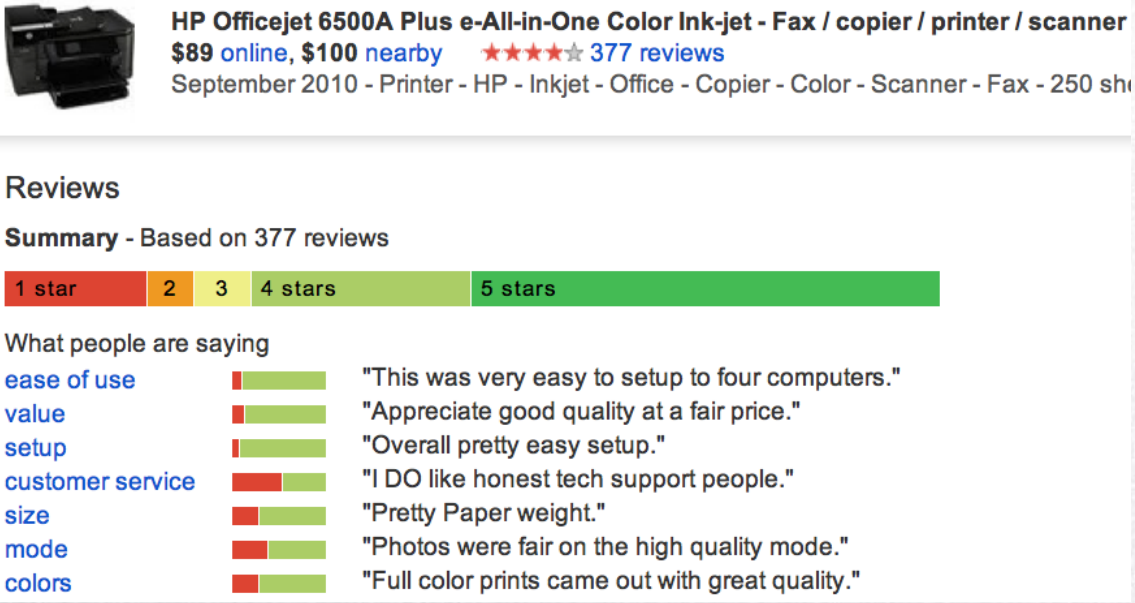
- 什么是情感分析?
- 情感分析是让计算机……识别人的情感



- Sentiment Analysis
- 对带有情感色彩的主观性文本进行分析、处理、归纳和推理，
- 最后得出文本的二元化或者多元化极性分类。

文本分类

- 案例：商品评论情感分析



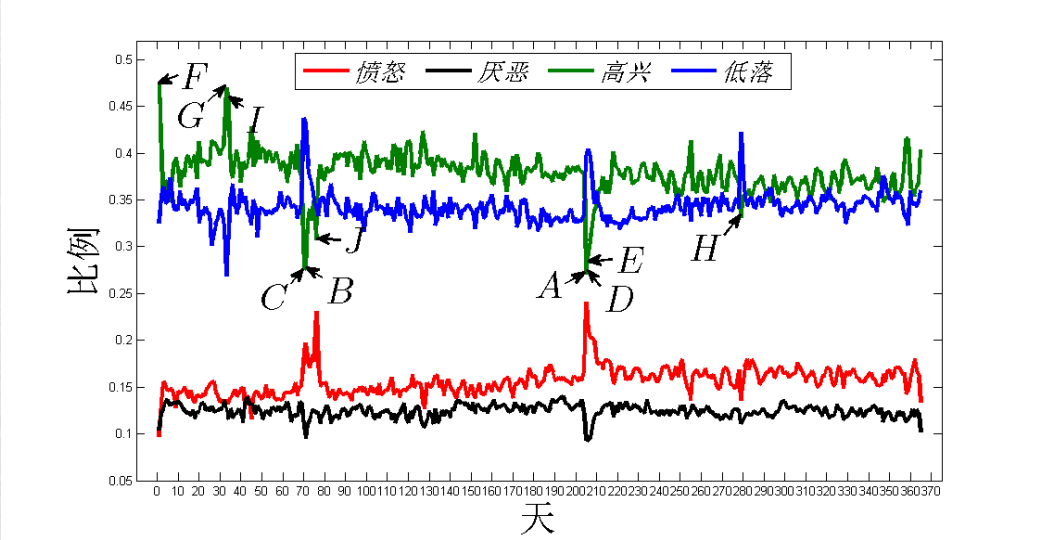
文本分类

- 案例：饭店评论情感分析

评论	打分
口味：不知道是我口高了，还是这家真不怎么样。??我感觉口味确实很一般很一般。上菜相当快，我敢...	2
菜品丰富质量好，服务也不错！很喜欢！	4
说真的，不晓得有人排队的理由，香精香精香精香精，拜拜！	2
菜量实惠，上菜还算比较快，疙瘩汤喝出了秋日的暖意，烧茄子吃出了大阪烧的味道，想吃土豆片也是口...	5
先说我算是娜娜家风荷园开业就一直在这里吃??每次出去回来总想吃一回??有时觉得外面的西式简餐...	4
哎，冲着推荐来的，后悔死了，上菜巨慢不说，服务态度还超级差，东西咸的要死，而且普遍的贵，谁来...	1
超级差评????上菜慢??服务态度超级差、先把汤上来喝饱了??完事菜就上了一个，找服务员催半...	1

文本分类

- 案例：微博情感分析



北航的先进网络分析研究小组(GANA) 对收集到的2011年的近7000万条微博进行情感分析

文本分类

- 案例：微博情感分析

  - 如F对应新年，以高兴的情绪为主；
  - G，I对应春节，也以高兴的情绪为主；
  - A、D、E则对应动车事故，明显看到用户以低落悲伤的情感为主，同时愤怒的情绪比例也明显上长升至全年最高；
  - C、B和J对应日本3月份的地震，以低落的情绪为主，但对J点，即2011年03月17日，愤怒的情感比例突然增加，这与当时的盛传碘盐被污染、盐荒等谣言有关；
  - H对应苹果前CEO乔布斯逝世，以低落的情绪为主，但同时，愤怒的情绪比例极低，与前面的动车事件和碘盐谣言有明显的区别。

文本分类

- 情感分析
- 澄清一些认识
- 情感分析是让计算机拥有人的情感吗？NO!

- 情感分析是让计算机超越人的情感吗？ NO!

爱，只是一个字而已
但人类千秋和万代
不明白一直到现在
但 A.I.
能克服所有问题

——《A.I.爱》

文本分类

- 列举文本分类应用案例
  - 语言识别（中文，英文，日文，...）
  - 垃圾网页识别（垃圾网页，正常网页）
  - 情感识别（正面、负面）
  - 主页识别（主页、一般网页）
  - ……

文本分类

- 文本分类的形式化定义
- 训练过程：
  - 给定文档空间 X
    - 所有的文档都在此空间进行表示，一般表示为高维向量
  - 文档类别 $C = \{c_1, \cdots, c_J\}$
    - 依据目标分类任务人工定义，比如$\mathbb{C} = \{\text{垃圾信息，正常信息}\}$
  - 带标签的训练集合$\mathrm{D} = \{(d_i, c_i)\}_{i=1}^{N}, (d_i, c_i) \in X \times C$
  - 基于学习算法，训练得到一个分类器 γ

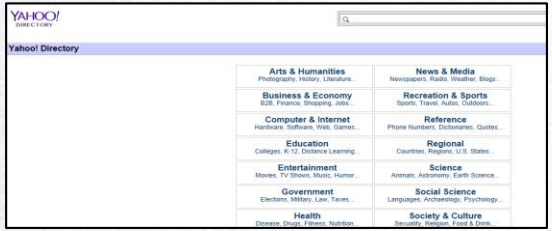
$$\gamma: X \mapsto C$$

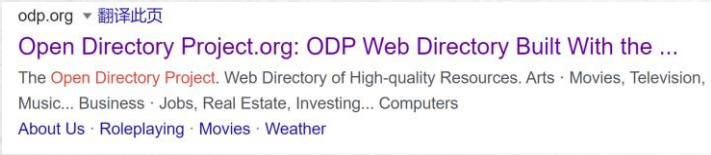
- 应用/测试过程
  - 给定：任意一个（可能在训练集合D中未出现）文档$d \in X$
  - 确定：$\gamma(d) = c_i \in C$，即对d而言最合适的类别

文本分类

文本分类

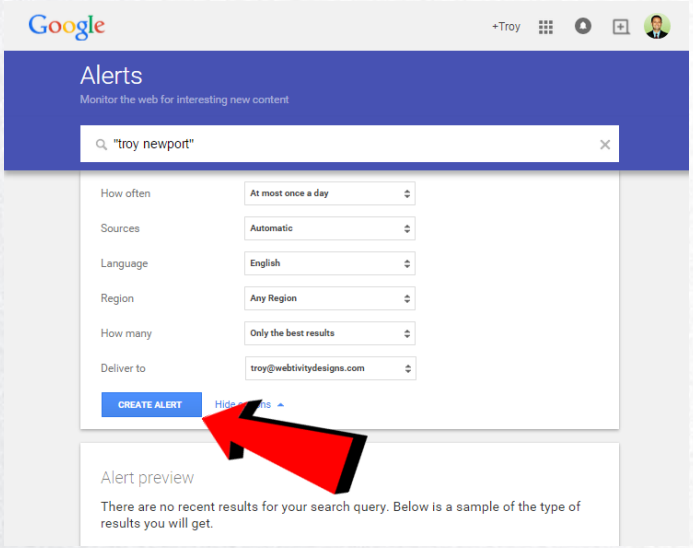
- 文本分类方法介绍：1#人工分类
- 雅虎最先提出将网页进行手工分类，ODP、PubMed也进行了类似的实践
  - 准确率高（如果雇佣专家进行分类）
  - 稳定（如果问题和团队规模可控）
- 规模化难题
  - 费用太高
  - 速度慢
  - 专家团队人数太多后，准确率难以保证
  - 于是需要自动化的分类方法！





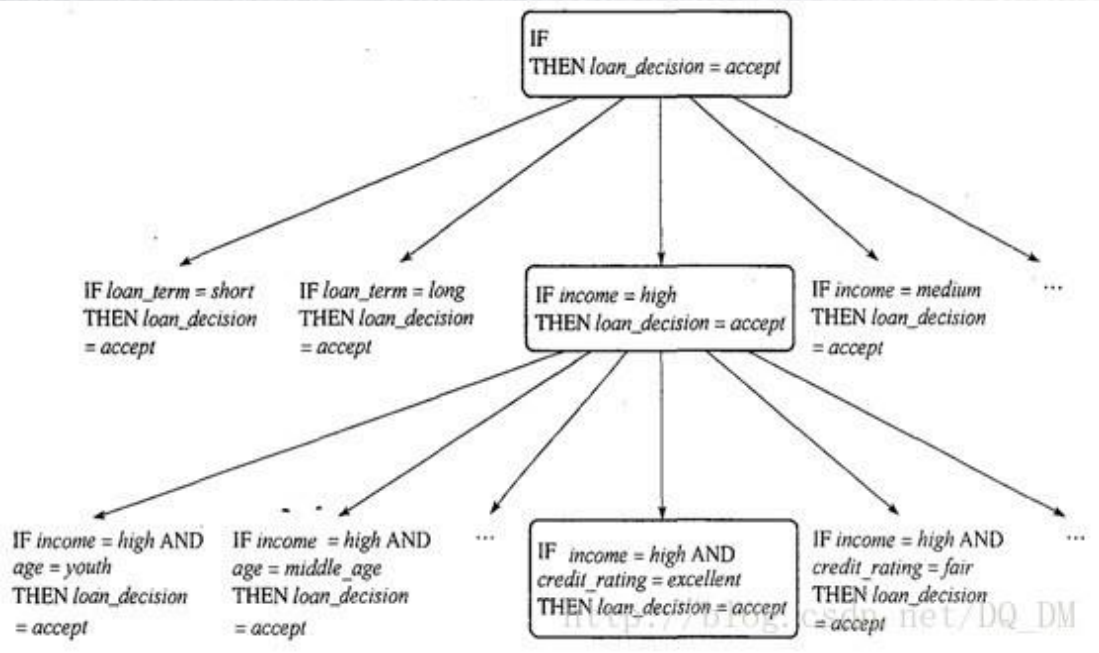
文本分类

- 文本分类方法介绍：2#基于规则的分类
- 例如：Google alerts是基于规则的分类
  - 通过辅助工具创建分类规则
  - 通常规则为布尔表达式
    - 如 ” 标题中含有 homepage”
    - AND ” URL中含有~”
- 优点：准确率高
  - 如果雇佣专家进行规则的编写和维护
  - 易于理解
- 缺点：规则维护任务繁重
  - 分类规则甚至类别经常发生变化
  - 规则冲突问题



文本分类

- 文本分类方法介绍：2#基于规则的分类
- 规则分类举例



文本分类

- 文本分类方法介绍：3#基于统计学习的分类
- 把文本分类问题看成一个机器学习问题
  - 基于监督学习算法，训练得到一个分类器 $\gamma: X \mapsto C$
  - 把γ应用于新文档d：$\gamma(d) \in C$

- 没有免费的午餐
  - 需要手工标注的训练数据（数据驱动）
  - 需要把文本表达为合适的**特征向量**

文本分类

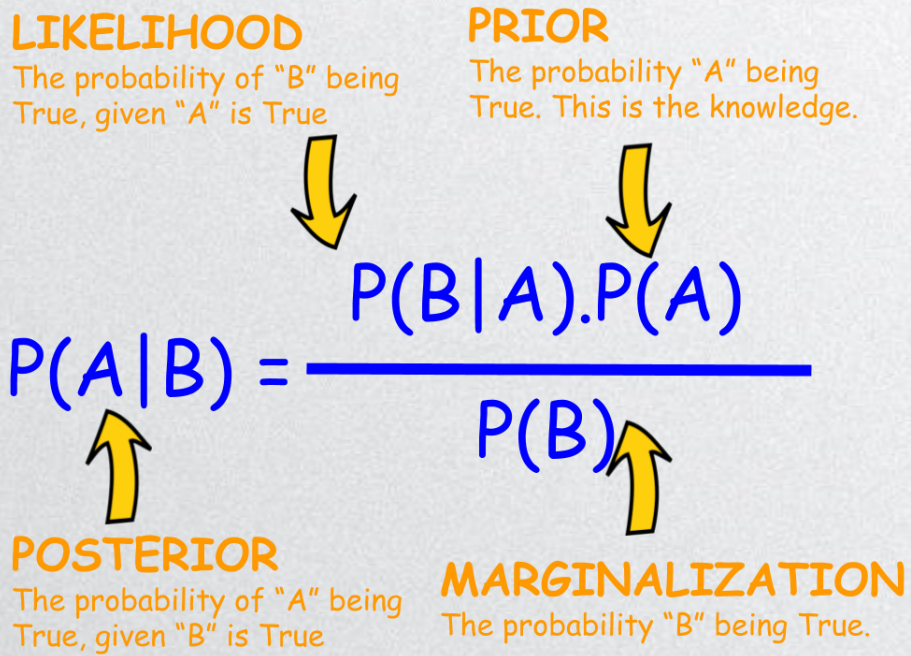
文本分类

- 贝叶斯公式介绍

https://en.wikipedia.org/wiki/Thomas_Bayes

文本分类

- 贝叶斯公式介绍



1，建立先验概率和后验概率的关系
2，提供计算后验概率的办法

文本分类

- 基于朴素贝叶斯分类器的文本分类
- 假设有文档d，文档类别集合C
- 分类目标：找出文档最可能属于的类别

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

- 应用贝叶斯公式$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

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

- 由于$P(d)$与类别选取没有关系，因此可以去除

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

- 朴素贝叶斯的目标：为文档找到“最好”的类
- 什么是最好的类？
- 能够最大化后验概率的类别 (maximum a posteriori (MAP) class)，记为c_{map}

文本分类

- 基于朴素贝叶斯分类器的文本分类
- 假设有文档d，文档类别集合C
- 逐个求解$P(\langle t_1,\cdots,t_k,\cdots,t_{n_d}\rangle|c)P(c)$非常困难
  - 数目过于庞大：$|V|^{n_d}$，$t_k \in V$可以为任意一个单词
  - 需要海量的训练数据去估计所有的$P(\langle t_1,\cdots,t_k,\cdots,t_{n_d}\rangle|c)$

- 数据稀疏问题

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

$$= \arg\max_{c\in\mathcal{C}} P(\langle t_1,\cdots,t_k,\cdots,t_{n_d}\rangle|c)P(c)$$

文本分类

- 基于朴素贝叶斯分类器的文本分类
- 假设一：条件独立假设
- 条件独立假设：在给定类别的条件下，文档中的单词出现概率独立

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

  - 独立：$P(AB) = P(A)P(B)$
- 给定类别c，文档d中所有的单词出现的联合概率$P(t_1, \cdots, t_{n_d}|c)$可以分解为各个单词出现概率$P(X_k = t_k|c)$的乘积，计算方法见后文

- Naïve Bayes独立性假设：
- 单词t在类别c中出现的概率其所出现的位置和上下文无关

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

- n_d：文档长度
- $\boldsymbol{P(t_k|c)}$：给定类别c，出现单词t_k的概率
- $\boldsymbol{P(c)}$：类别c的先验概率

文本分类

- 基于朴素贝叶斯分类器的文本分类：参数估计
- 给定训练数据之后，如何估计上述概率（参数）？

- 需要估计的参数包括
  - 先验概率：$P(\mathrm{c}), \text{for all } \mathrm{c} \in C = \{c_1, \cdots, c_J\}$
  - 条件概率：$P(\mathrm{t_k}|\mathrm{c}), \text{for all } \mathrm{t_k} \in V, \text{and } \mathrm{c} \in C = \{c_1, \cdots, c_J\}$, 其中V为所有可能的单词

$$
\begin{aligned}
c_{\text{map}} &= \arg\max_{c \in \mathcal{C}} P(d|c)P(c) \\
&= \arg\max_{c \in \mathcal{C}} P(\langle t_1, \cdots, t_k, \cdots, t_{n_d} \rangle | c)P(c) \\
&= \arg\max_{c \in \mathcal{C}} \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k|c)
\end{aligned}
$$

文本分类

- 基于朴素贝叶斯分类器的文本分类：参数估计（最大似然估计）
- 先验概率：

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

  - N_c: 训练集合中属于类别c的文档数目；N: 训练集合中所有文档数目

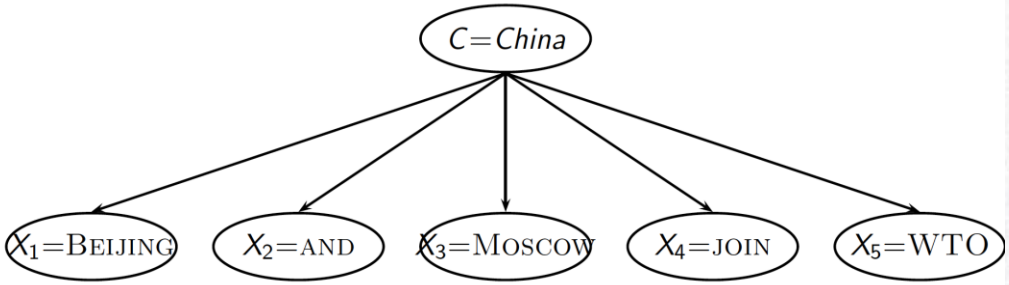
- 条件概率：

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

  - T_{ct}: 在训练集合中，属于类别c的文档中包含单词t的次数**（包括重复出现）**

文本分类

- 基于朴素贝叶斯分类器的文本分类：参数估计（最大似然估计的问题：零概率）



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

- 如果单词WTO从来没有在类别China中出现过

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

文本分类

- 基于朴素贝叶斯分类器的文本分类：参数估计（最大似然估计的问题：零概率）
- 在训练数据中，属于China的文档中从来没有出现过单词WTO，则

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

- 依据分类规则，则所有含有WTO的文档属于China类别的概率为0：
  - $P(China|d) = 0$，如果d中含有单词WTO

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

文本分析

- 基于朴素贝叶斯分类器的文本分类：参数估计（最大似然估计的问题：零概率）
- 避免零概率：概率平滑
- 最大似然条件概率估计：

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

- 平滑化：在所有的统计次数上加一次

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

  - B=|V|：字典中所含有**不同的单词**数目

文本分析

- 一个实例

	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	?

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

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

文本分析

- 一个实例

	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	?

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

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

字典表：|V|=6
Chinese
Beijing
shanghai
Macao
Tokyo
japan

B=|V|：字典中所含有不同的单词数目

文本分析

- 一个实例

	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	?

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

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

字典表：|V|=6
Chinese,Beijing,shanghai,
macao,Tokyo, japan

B=|V|：字典中所含有不同的单词数目

Priors: $\hat{P}(c) = 3/4$

$\hat{P}(\bar{c}) = 1/4$

Conditional probabilities:

Chinese Chinese Chinese Tokyo Japan

这是新文档，需要确定类别

$$\hat{P}(\text{CHINESE}|c) = (5+1)/(8+6) = 6/14 = 3/7$$
$$\hat{P}(\text{TOKYO}|c) = (0+1)/(8+6) = 1/14$$
$$\hat{P}(\text{JAPAN}|c) = (0+1)/(8+6) = 1/14$$

$$\hat{P}(\text{CHINESE}|\bar{c}) = (1+1)/(3+6) = 2/9$$
$$\hat{P}(\text{TOKYO}|\bar{c}) = (1+1)/(3+6) = 2/9$$
$$\hat{P}(\text{JAPAN}|\bar{c}) = (1+1)/(3+6) = 2/9$$

文本分析

- 一个实例

	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	?

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

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

字典表：|V|=6
Chinese,Beijing,shanghai,
macao,Tokyo, japan

B=|V|：字典中所含有不同的单词数目

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

Priors: $\hat{P}(c) = 3/4$

$\hat{P}(\bar{c}) = 1/4$

Conditional probabilities:

Chinese Chinese Chinese Tokyo Japan

这是新文档，需要确定类别

$\hat{P}(\text{CHINESE}|c) = (5+1)/(8+6) = 6/14 = 3/7$

$\hat{P}(\text{TOKYO}|c) = (0+1)/(8+6) = 1/14$

$\hat{P}(\text{JAPAN}|c) = (0+1)/(8+6) = 1/14$

$\hat{P}(\text{CHINESE}|\bar{c}) = (1+1)/(3+6) = 2/9$

$\hat{P}(\text{TOKYO}|\bar{c}) = (1+1)/(3+6) = 2/9$

$\hat{P}(\text{JAPAN}|\bar{c}) = (1+1)/(3+6) = 2/9$

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

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

文本分类

文本分析

- 另一个实例
- 有如下文档及其分类，请使用朴素贝叶斯方法对第6个文档进行分类。

1	人民大学明德楼	人大
2	明德楼办公室	人大
3	北京大学未名湖	北大
4	一勺池是小湖泊	人大
5	未名湖是大湖泊	北大
6	明德楼未名湖湖泊	?

词汇表如下： **10个词汇**
人民大学、明德楼、办公室、北京大学、未名湖、一勺池、是、小、湖泊、大

文本分析

- 另一个实例
- 有如下文档及其分类，请使用朴素贝叶斯方法对第6个文档进行分类。

1	人民大学明德楼	人大
2	明德楼办公室	人大
3	北京大学未名湖	北大
4	一勺池是小湖泊	人大
5	未名湖是大湖泊	北大
6	明德楼未名湖湖泊	?

计算先验概率

$P(\text{人大})=\frac{3}{5}$

$P(\text{北大})=\frac{2}{5}$

文本分析

- 另一个实例
- 有如下文档及其分类，请使用朴素贝叶斯方法对第6个文档进行分类。

1	人民大学明德楼	人大
2	明德楼办公室	人大
3	北京大学未名湖	北大
4	一勺池是小湖泊	人大
5	未名湖是大湖泊	北大
6	明德楼未名湖湖泊	?

P(人民大学、明德楼、办公室、北京大学、未名湖、一勺池、是、小、湖泊、大|人大)=

$\frac{1+1}{8+10}$、$\frac{2+1}{8+10}$、$\frac{1+1}{8+10}$、$\frac{0+1}{8+10}$、$\frac{0+1}{8+10}$、$\frac{1+1}{8+10}$、$\frac{1+1}{8+10}$、$\frac{1+1}{8+10}$、$\frac{1+1}{8+10}$、$\frac{0+1}{8+10}$

文本分析

- 另一个实例
- 有如下文档及其分类，请使用朴素贝叶斯方法对第6个文档进行分类。

1	人民大学明德楼	人大
2	明德楼办公室	人大
3	北京大学未名湖	北大
4	一勺池是小湖泊	人大
5	未名湖是大湖泊	北大
6	明德楼未名湖湖泊	?

P(人民大学、明德楼、办公室、北京大学、未名湖、一勺池、是、小、湖泊、大|人大)=

$\frac{0+1}{6+10}$、$\frac{0+1}{6+10}$、$\frac{0+1}{6+10}$、$\frac{1+1}{6+10}$、$\frac{2+1}{6+10}$、$\frac{0+1}{6+10}$、$\frac{1+1}{6+10}$、$\frac{0+1}{6+10}$、$\frac{1+1}{6+10}$、$\frac{1+1}{6+10}$

文本分析

- 另一个实例
- 有如下文档及其分类，请使用朴素贝叶斯方法对第6个文档进行分类。

1	人民大学明德楼	人大
2	明德楼办公室	人大
3	北京大学未名湖	北大
4	一勺池是小湖泊	人大
5	未名湖是大湖泊	北大
6	明德楼未名湖湖泊	?

计算后验概率

P(人大|明德楼、**未名湖**、湖泊)∝P(人大)P(明德楼|人大) P(**未名湖**|人大) P(湖泊|人大)

$\frac{3}{5}\frac{3}{18}\frac{1}{18}\frac{2}{18}$=18/29160=0.00062

P(北大|明德楼、**未名湖**、湖泊)∝P(北大)P(明德楼|北大) P(**未名湖**|北大) P(湖泊|北大)

$\frac{2}{5}\frac{1}{16}\frac{3}{16}\frac{2}{16}$=12/20480=0.00059

该文档分类为人大

文本分类

文本分类

- 朴素贝叶斯训练算法

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** $V, prior, condprob$

文本分类

- 朴素贝叶斯应用/测试

```
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]
```

参考下页

通过log把乘法变成加法

- 计算机表达概率相乘所遇到的问题
  - 如果把很多个概率（0~1之间的数字）相乘，会很快得到低于计算机所能表达精度的数字（**floating point underflow**），从而得到0概率
- 解决方案：log(XY)=log(X) + log(Y)
  - 通过求log和的方式表达积的log
  - 注意：log是一个单增函数，取log后的函数并不改变其最大值点
  - 因此，Naïve Bayes实际去计算

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

朴素贝叶斯时间复杂度

- L_{ave}: 平均文档长度；D：训练文档集合；V：字典集合；C：类别集合
  - $\Theta(|D|L_{ave})$: 训练文档中所有的统计计数
  - $\Theta(|C||V|)$: 基于统计计数计算参数值
  - 一般而言： $|C||V| < |D|L_{ave}$

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_a: 测试文档长度； M_a: 测试文档中不同的单词数；
  - 测试时间复杂度：对文档长度而言为线性
  - 结论：朴素贝叶斯在训练和测试中均为线性时间复杂度（对文档长度而言）

文本分类

文本分类

- 朴素贝叶斯的问题与效果
- 假设二：位置无关假设
- $P(X_{k_1} = t|c) = P(X_{k_2} = t|c)$
  - 假设对于属于类别“China”的一个文档d，其第1个位置出现Beijing和最后一个位置出现Beijing，所产生的效果是一样的

条件独立假设前文已经介绍

词袋假设(Bag of words, BOW)
假设1：不同单词之间独立
假设2：与位置无关

文本分类

- 朴素贝叶斯的问题与效果
- 上述独立性假设是否合理?
- 条件独立:

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

- 位置独立:

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

- 在实际的文本写作中，上述假设并不成立
  - 请举出一个例子，说明假设一（条件独立假设）并不成立。
    - China: Chinese great wall
    - Japan: japan great Tokyo
  - 请举出一个例子，说明假设二（位置无关假设）并不成立。
    - Japan: japan attack alien
    - Alien: alien attack japan

- 问题：既然独立性假设被违背，为何朴素贝叶斯仍然有效?

文本分类

- 朴素贝叶斯的问题与效果——为何依然有效?
- 朴素贝叶斯在条件独立假设被严重违背的情况下，任然有效!
- 举例:

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

  - 违背条件独立性假设，会重复利用单词的分类贡献，导致对概率的估计要么不足(underestimation, 0.01)，要么太过 overestimation (0.99)
  - 但是：分类问题的目标为正确预测类别，而不是精确估计概率
  - 在条件独立假设被严重违背的情况下，朴素贝叶斯
    - 不能够精确估计文档属于某一个类别的概率
    - 但是通常还是能够“猜”对类别（违背独立性假设并未严重到颠覆类别的**对比**）

文本分类

- 并不“朴素”的朴素贝叶斯
- 赢得KDD-CUP 1997年冠军
- 相对于大部分传统的机器学习模型
  - 对无关的特征比较鲁棒
  - 对topic drift（比如：类别的定义随着时间而变化）鲁棒
- 相对决策树
  - 如果一些特征同样重要，则朴素贝叶斯表现更好
- 如果独立性假设成立，会有惊艳的表现

- 是文本分类中一类重要的模型，简单有效，经常被用作基线模型

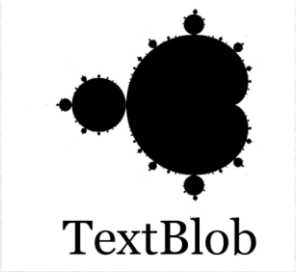
简单：没有超参数
时间：速度快、效率高
空间：需要存储空间少

文本分类

文本分类

- NLTK文本分类实践
- NLTK：处理自然语言数据的Python程序和平台
- Gensim：可扩展的统计语义、分析纯文本文档的语义结构、检索语义相似的文档
- TextBlob：用于处理文本数据的Python库

NLTK

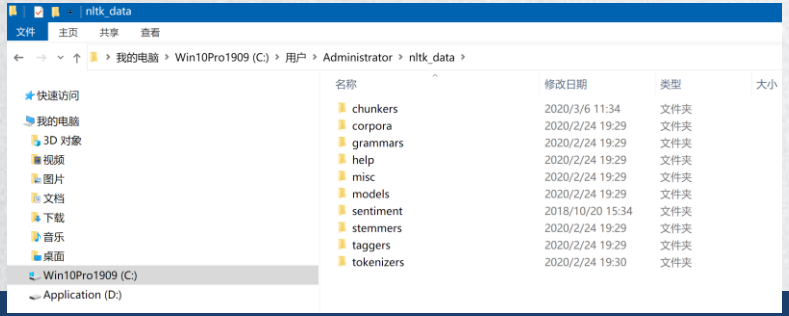


8种Python文本处理工具集
https://www.jiqizhixin.com/articles/2018-10-29-26

文本分类

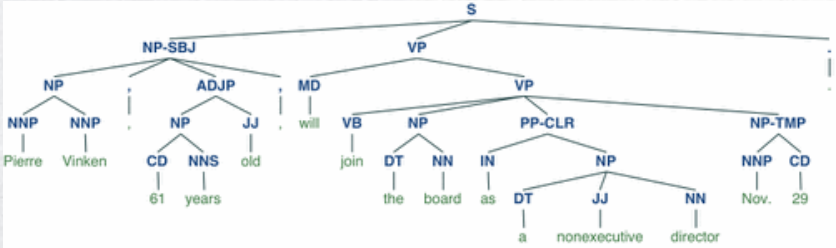
- NLTK文本分类实践
- NLTK安装
- pip install nltk –i https://pypi.douban.com/simple

- 数据下载（可选）
  - NLTK附带了许多语料库，toy grammar以及训练模型等。完整的列表发布在：http://nltk.org/nltk_data/
  - 要安装这些数据，首先安装NLTK，再利用NLTK的数据下载器进行下载



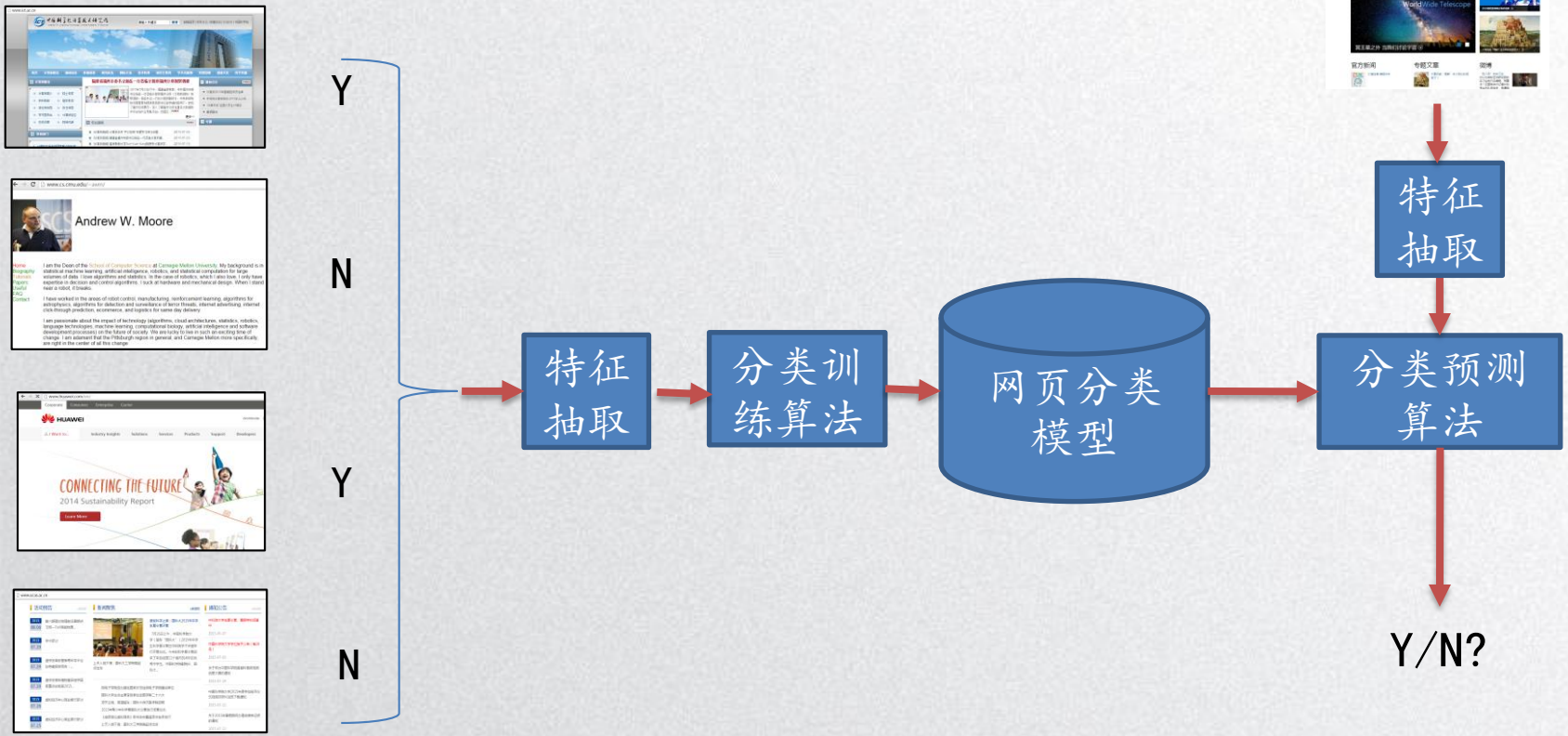
文本分类

- NLTK文本分类实践
- 自然语言处理
  - Tokenization and POS tagging
  - Identify named entities
  - parse tree
  - Stemming
  - 情感分类
  - 语言模型
- 机器翻译
- 分类
  - ConditionalExponentialClassifier
  - DecisionTreeClassifier
  - MaxentClassifier
  - NaiveBayesClassifier
- 聚类
  - EM
  - k-Means



Naïve Bayes仅仅是NLTK很小的一部分

监督学习模型训练使用流程



使用说明

- 使用NLTK朴素贝叶斯进行分类实验
- 训练和测试数据集合分别都被表示为list
  - List中每一个元素为一个pair：第一个为特征，第二个为分类标签
  - 每一个特征（一个文档）都被表示成一个dict

- 训练模型： classifier = NaiveBayesClassifier.train(train)
- 测试模型： labels = classifier.classify_many(test)

- 得到分类的概率： probs = classifier.prob_classify_many(test)
- 查看每一维特征的作用： classifier.show_most_informative_features()

文本分类

- 使用NLTK朴素贝叶斯进行分类实验

```
from nltk.classify import NaiveBayesClassifier
train = [
(dict(a=1,b=1,c=1), 'y'),
(dict(a=1,b=1,c=1), 'x'),
(dict(a=1,b=1,c=0), 'y'),
(dict(a=0,b=1,c=1), 'x'),
(dict(a=0,b=1,c=1), 'y'),
(dict(a=0,b=0,c=1), 'y'),
(dict(a=0,b=1,c=0), 'x'),
(dict(a=0,b=0,c=0), 'x'),
(dict(a=0,b=1,c=1), 'y'),
]
test = [
(dict(a=1,b=0,c=1)), # unseen
(dict(a=1,b=0,c=0)), # unseen
(dict(a=0,b=1,c=1)), # seen 3 times, labels=y,y,x
(dict(a=0,b=1,c=0)), # seen 1 time, label=x
]

#train
classifier = NaiveBayesClassifier.train(train)

#test
labels = classifier.classify_many(test)
print(labels)

#show probabilities
probs = classifier.prob_classify_many(test)
for pdist in probs:
    print('%.4f %.4f' % (pdist.prob('x'), pdist.prob('y')))

#how these features work
classifier.show_most_informative_features()
```

```
['y', 'x', 'y', 'x']
0.3203 0.6797
0.5857 0.4143
0.3792 0.6208
0.6470 0.3530
Most Informative Features
                    c = 0                x : y      =      2.0 : 1.0
                    c = 1                y : x      =      1.5 : 1.0
                    a = 1                y : x      =      1.4 : 1.0
                    b = 0                x : y      =      1.2 : 1.0
                    a = 0                x : y      =      1.2 : 1.0
                    b = 1                y : x      =      1.1 : 1.0
```

代码运行与分析

文本分类

文本分类

- 文本二值分类

```
train_corpus= [('The team dominated the game', True),
        ('The game was intense', True),
        ('The ball went off the court', True),
        ('They had the ball for the whole game', True),
        ('The President did not comment', False),
        ('The show is over', False),
       ]
#build features
v = voc(train_corpus)
#train
train_feature_label = feature(train_corpus, v)
NBC = NaiveBayesClassifier.train(train_feature_label)
#test
test_corpus = [('I lost the keys', False),
        ('The goalkeeper catched the ball', True),
        ('The other team controlled the ball', True),
        ('Sara has two kids', False),
        ('this is a book', True),
       ]
test_feature = []
test_labels=[]
for (ftr, label) in feature(test_corpus, v):
    test_feature.append(ftr)
    test_labels.append(label)

pred_labels = NBC.classify_many(test_feature)
print(test_labels)
print(pred_labels)

print('Precision = %.4f, Recall = %.4f, F-score = %.4f, Accuracy = %.4f'%
      classify_eval(test_labels, pred_labels))

# show probabilities
probs = NBC.prob_classify_many(test_feature)
for pdist in probs:
    print('%.4f %.4f' % (pdist.prob(True), pdist.prob(False)))
```

```
from nltk.classify import NaiveBayesClassifier

def voc(data):
    voc = {}
    for (sentence, val) in data:
        words = sentence.lower().split()
        for w in words:
            voc[w] = True
    return voc


def feature(data, v):
    ftr = []
    for (sentence, label) in data:
        f = dict((w, 0) for w in dict.keys(v))
        words = sentence.lower().split()
        for w in words:
            f[w] = 1
        ftr.append((f, label))
    return ftr
```

创建字典

1，一个样本(feature, label)
2，其中Feature是一个字典

文本分析

- 结果评价

```python
def classify_eval(truth, pred):
    idx = 0
    (TP, FP, TN, FN) = (0, 0, 0, 0)
    for truth_label in truth:
        pred_label = pred[idx]
        if (truth_label == True and pred_label == True):
            TP = TP + 1
        elif (truth_label == False and pred_label == False):
            TN = TN + 1
        elif (truth_label == True and pred_label == False):
            FN = FN + 1
        elif (truth_label == False and pred_label == True):
            FP = FP + 1
        idx = idx + 1
    P = 0 if TP == 0 else TP / (TP + FP)
    R = 0 if TP == 0 else TP / (TP + FN)
    F = 0 if (P == 0 or R == 0) else 2* P * R / (P + R)
    Acc = 0 if (TP + TN == 0) else (TP + TN) / (TP + TN + FP + FN)
    return (P, R, F, Acc)
```

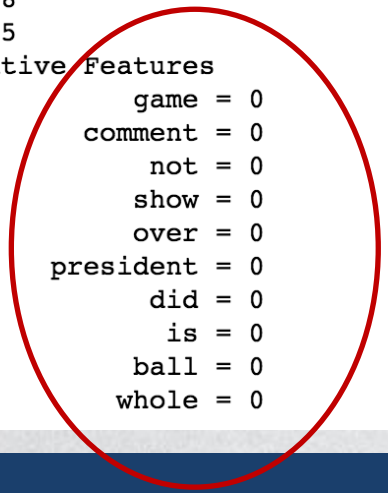
```
[False, True, True, False, True]
[True, True, True, True, False]
Precision = 0.5000, Recall = 0.6667, F-score = 0.5714, Accuracy = 0.4000
0.7776 0.2224
0.9459 0.0541
0.9740 0.0260
0.6602 0.3398
0.1775 0.8225
Most Informative Features
                    game = 0              False : True   =      2.8 : 1.0
                 comment = 0               True : False  =      1.8 : 1.0
                     not = 0               True : False  =      1.8 : 1.0
                    show = 0               True : False  =      1.8 : 1.0
                    over = 0               True : False  =      1.8 : 1.0
               president = 0               True : False  =      1.8 : 1.0
                     did = 0               True : False  =      1.8 : 1.0
                      is = 0               True : False  =      1.8 : 1.0
                    ball = 0              False : True   =      1.7 : 1.0
                   whole = 0              False : True   =      1.2 : 1.0
```

文本分类

- 哪些单词对分类最有意义?

NBC.show_most_informative_features(10)

```
[False, True, True, False, True]
[True, True, True, True, False]
Precision = 0.5000, Recall = 0.6667, F-score = 0.5714, Accuracy = 0.4000
0.7776 0.2224
0.9459 0.0541
0.9740 0.0260
0.6602 0.3398
0.1775 0.8225
Most Informative Features
                    game = 0              False : True    =      2.8 : 1.0
                 comment = 0               True : False   =      1.8 : 1.0
                     not = 0               True : False   =      1.8 : 1.0
                    show = 0               True : False   =      1.8 : 1.0
                    over = 0               True : False   =      1.8 : 1.0
               president = 0               True : False   =      1.8 : 1.0
                     did = 0               True : False   =      1.8 : 1.0
                      is = 0               True : False   =      1.8 : 1.0
                    ball = 0              False : True    =      1.7 : 1.0
                   whole = 0              False : True    =      1.2 : 1.0
```



文本分类

- 使用SVM进行文本分类

from sklearn.svm import LinearSVC

```
SVMC = nltk.classify.SklearnClassifier(LinearSVC())
SVMC.train(train_feature_label)
pred_labels_SVM = []
for f in test_feature:
    pred_labels_SVM.append(SVMC.classify(f))

print(pred_labels_SVM)
print('Precision = %.4f, Recall = %.4f, F-score = %.4f, Accuracy = %.4f'%
      classify_eval(test_labels, pred_labels_SVM))
```

文本分类

垃圾短信文本分类练习

- 要求
  - 1. 基于NLTK中的朴素贝叶斯对短信数据进行分类
    - 数据处理（中文分词）
    - 特征抽取
    - 模型训练
    - 模型评价
  - 2. 考虑优化特征，如：去除停用词等
  - 3. 提交对比报告

标签	短信内容
0	商业秘密的秘密性那是维系其商业价值和垄断地位的前提条件之一
1	南口阿玛施新春第一批限量春装到店啦　春暖花开淑女裙、冰蓝色公主衫　气质粉小西装、冰丝女王长半裙
0	带给我们大常州一场壮观的视觉盛宴
0	有原因不明的泌尿系统结石等
0	23年从盐城拉回来的麻麻的嫁妆
1	感谢致电杭州萧山全金釜韩国烧烤店，本店位于金城路xxx号。韩式烧烤等，价格实惠、欢迎惠顾【全金釜韩国烧烤店】
0	这款UVe智能杀菌机器人是扫地机的最佳伴侣
1	一次价值xxx元王牌项目；可充值xxx元店内项目卡一张；可以参与V动好生活百分百抽奖机会一次！预约电话：xxxxxxxxxx
0	此类皮肤特别容易招惹粉刺、黑头等
1	(长期诚信在本市作各类资格职称（以及印 /章、牌、 ……等。祥：ｘｘｘｘｘｘｘｘｘｘｘ李伟%

文本分类

- 参考资料

- 垃圾短信分类
- https://blog.csdn.net/qq_38525781/article/details/86599089
- 垃圾短信分类器
- https://github.com/hrwhisper/SpamMessage