Natural Language Processing and Its Applications

#L8

Text Classification

袁彩霞

yuancx@bupt.edu.cn

智能科学与技术中心

Overview

- 目前为止：
  - 语言模型可以计算p(s)
    - 用于衡量语言的通顺性（“出现的可能性”）
  - 句子句法分析p(t)：
    - 表示语言的结构信息
  - 但都不能表示语言的主题、类别、语义等

- 文本分类
  - 将文本划分到特定的类别或主题下，例如政治、经济、体育等

特朗普表示，再次感谢习近平主席对我到访中国给予的热情接待，中国悠久灿烂的历史文化也给我留下深刻印象。

傅里叶分析不仅仅是一个数学工具，更是一种可以彻底颠覆一个人以前世界观的思维模式。但不幸的是，傅里叶分析的公式看起来太复杂了，所以很多大一新生上来就懵圈并从此对它深恶痛绝。

- 问题：
  - 如何判断文档的主题类别？
  - 做这种判断需要哪些语言分析技术？

文本分类：例子

- 新闻文本 → 商业、科技、娱乐、健康
- 校布告栏信息 → 学生、老师、教辅人员
- 学生作文 → 等级A，B，C，D
- Email → 垃圾邮件、正常邮件
- 科技文献 → 感兴趣、不感兴趣
- Movie → 好评、中评、差评
- 商品 → 推荐、不推荐
- ……

如何进行分类？

- （手工）规则：Hand-coded rules
  - 一些垃圾邮件过滤系统：
    - 例如，如果email里包含$、RMB、发票、免费等，则判断为spam
  - 如果规则定义好，准确率往往很高
  - 然而，编制及维护规则的成本很大

如何进行分类？

- 机器学习：
  - 学习一个从文档到文档类别的映射函数
  - **BUT no free lunch**: 一般需要人工分好类的文本作为学习样本（训练数据）
    - 标注数据相较于编制规则较为容易（一般不需要专家直接参与，例如：众包）
  - 机器学习
    - 有监督学习、无监督学习？

如何进行分类？

- 有监督学习模型：
  - 学习器：
    - *输入*：m 个手工标注了类别的文本集合 $(d_1,c_1),\ldots,(d_m,c_m)$ (training data)
    - *输出*：训练好的分类器 $f: d \rightarrow c$
  - 分类器：
    - *输入*: 文档 d (test data)
    - *输出*: 从类别集合 $c_1,\ldots,c_K$ 中选择一个类别作为 d 的类别输出

Outline

- 文本分类
  - Step 1：预处理
  - Step 2：文本表示
  - Step 3：分类模型
  - Step 4：评价
  - 其它：特征选择

Step 1: 预处理

- 依据具体的文本形式及任务而定：
  - 去除HTML (or other)标签
  - Stop-words(停用词):
    - 高频词往往携带较少信息
    - E.g.,“a，an，the，this，for，at，on“,”的，得，地，这，尽管，但是” etc.
  - Word stemming(词干):
    - 词的后缀及变形处理
    - 将具有相同概念意义的词进行合并，例如 walk, walker, walked, and walking

Step 2: 文本表示

(或文本索引，后续再提)

f(ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS
BUENOS AIRES, Feb 26
Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:
- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....)=c

?

简单、好用

What is the best representation for the document *d* being classified?

向量空间模型

- 一种常用的文本表示方法：Vector Space Model (VSM, 向量空间模型)
  - 将文本表示为由词条构成的向量
    - 不仅是文本，广泛适用于其它数据对象
  - 理论上假设词条之间统计独立（不考虑词在文本中的出现顺序），即文本是由词构成的词集合（词袋， bag-of-words ）

向量空间模型

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS

BUENOS AIRES, Feb 26

Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

→ Categories: grain, wheat

向量空间模型

xxxxxxxxxxxxxxxxxxx GRAIN/OILSEED xxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxx grain xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx grains, oilseeds xxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxx tonnes, xxxxxxxxxxxxxxxx shipments
xxxxxxxxxxxx total xxxxxxxxx total xxxxxxxx xxxxxxxxxxxxxxxxxxx:

- Xxxxx wheat xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, total xxxxxxxxxxxxxxx
- Maize xxxxxxxxxxxxxxxxx
- Sorghum xxxxxxxxxx
- Oilseed xxxxxxxxxxxxxxxxxxxx
- Sunflowerseed xxxxxxxxxxxxxx
- Soybean xxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx....

→ Categories: grain, wheat

向量空间模型

xxxxxxxxxxxxxxxxxx GRAIN/OILSEED xxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxx grain xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx grains, oilseeds
xxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxx tonnes,
xxxxxxxxxxxxxxxxx shipments xxxxxxxxxxxx total xxxxxxxxx total
xxxxxxxx xxxxxxxxxxxxxxxxxxxx:

- Xxxxx wheat xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, total xxxxxxxxxxxxxxxx
- Maize xxxxxxxxxxxxxxxxx
- Sorghum xxxxxxxxxx
- Oilseed xxxxxxxxxxxxxxxxxxxxx
- Sunflowerseed xxxxxxxxxxxxxx
- Soybean xxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx....

word	freq	Other statistics
grain(s)	3	…
oilseed(s)	2	…
total	3	…
wheat	1	…
maize	1	…
soybean	1	…
tonnes	1	…
…	…	…

向量空间模型

- 文档-词条矩阵 $A=(a_{ik})$
  - 每个文档表示为由词构成的列向量
  - a_{ik} 表示词 k 在文档 i 中的权重

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1M} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2M} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{N1} & a_{N2} & \cdots & \cdots & a_{NM} \end{pmatrix}$$

- 如何计算权重a_{ik}?
  - 一个词条在某个文档中出现的次数越多，则这个词条与此文档的类别相关性越大
  - 一个词条在所有文档中出现的次数都很多，则这个词条对于文档的类别区分性就越低

几个符号

- f_{ik} 词条 k 在文档i中的出现次数
- n_k 词条 k 在文档集合中的出现总次数
- N 文档集合包含的文档个数
- M 预处理后文档集合包含的词条个数

词的权重

- 布尔权重：
  - 如果词在文档中出现，则权重为1；否则为0

    $a_{ik}=1$， if $f_{ik}>0$;

    $a_{ik}=0$， otherwise

- 词条频次权重(term frequency weighting, tf)
  - 使用词条在文档中的出现次数作为词的权重
  - $a_{ik}=f_{ik}$

词的权重

- 逆文档频次(inverse document frequency, idf)
  - 考虑包含某词条的文档个数
  - $a_{ik} \propto 1/n_k$

- tf × idf 权重：
  - 同时考虑词条频次和逆文档频次

$$a_{ik} = f_{ik} * \log(\frac{N}{n_k})$$

  - 正比于词条在文档中的出现频次
  - 反比于包含词条的文档个数

词的权重

- tf-idf的一些变形：
  - tfc (term frequency component)权重：
    - 对文档长度进行正则化

$$a_{ik} = \frac{f_{ik} * \log(\frac{N}{n_k})}{\sqrt{\sum_{j=1}^{M}[f_{ij} * \log(\frac{N}{n_j})]^2}}$$

  - ltc 权重：
    - 减小绝对频次的差异带来的影响

$$a_{ik} = \frac{\log(f_{ik}+1) * \log(\frac{N}{n_k})}{\sqrt{\sum_{j=1}^{M}[\log(f_{ij}+1) * \log(\frac{N}{n_j})]^2}}$$

词的权重

- tf-idf的其它变形：

Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1+\log(\mathrm{tf}_{t,d})$	t (idf)	$\log\frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5+\frac{0.5\times\mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0,\log\frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases}1 & \text{if } \mathrm{tf}_{t,d}>0\\ 0 & \text{otherwise}\end{cases}$			b (byte size)	$1/CharLength^{\alpha}$, $\alpha<1$
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t\in d}(\mathrm{tf}_{t,d}))}$				

词的权重

- 熵权重(Entropy weighting)：

$$a_{ik} = \log(f_{ik} + 1) * (1 + \frac{1}{\log(N)} \sum_{j=1}^{N} \frac{f_{jk}}{n_k} \log(\frac{f_{jk}}{n_k}))$$

  - 词条 k 的信息熵:

$$(1 + \frac{1}{\log(N)} \sum_{j=1}^{N} \frac{f_{jk}}{n_k} \log(\frac{f_{jk}}{n_k}))$$

    - 如果在所有文档中的分布相等，则熵为-1；如果只在一个文档中出现，则熵为0

Step 3: 分类模型

- 任务：通过学习得到一个映射：$f(x) \rightarrow y$

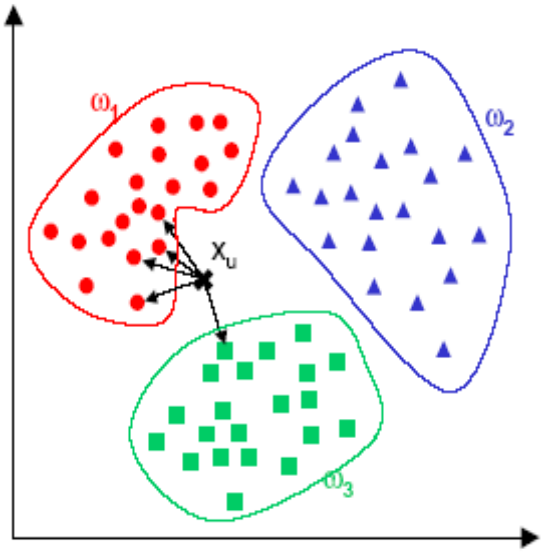
- 如何构建计算模型？
  - 最近邻模型
  - 概率模型
  - 回归模型
  - 其它的决策模型？

Step 3: 分类模型

I) Instance-based methods:
   1) Nearest neighbor
II) Probabilistic models:
   1) Naïve Bayes
   2) Maximum Entropy Model
   3) Neural Network
III) Linear Models:
   1) Linear Regression/Perceptron
   2) Support Vector Machine
IV) Decision Models:
   1) Decision Trees
   2) Random Forest

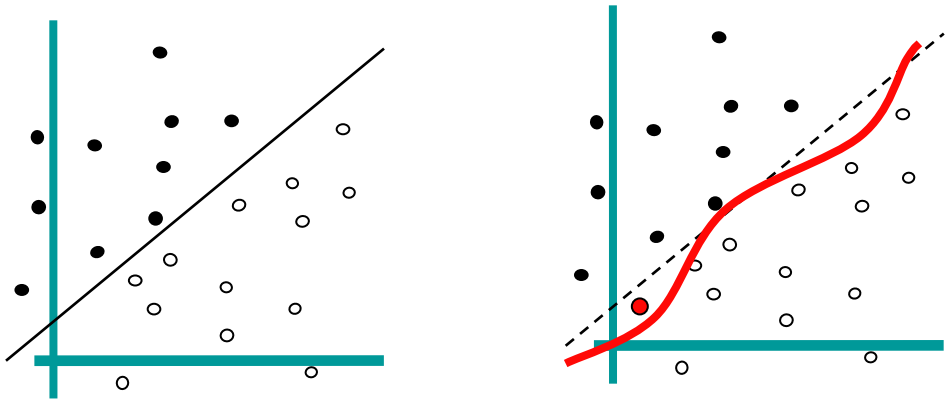
最近邻分类器

- Nearest neighbor classifier
- 思想：
  - 定义两个样本点之间的距离函数
    - e.g., $d(x_1,x_2)=||x_1-x_2||$
  - 将新的样本划分到距离它最近的样本（最近邻）所属的类别中
    - $f(x) = y_{i^*}$
    - where $i^*=\arg_i \min d(x, x_i)$



最近邻分类器的问题

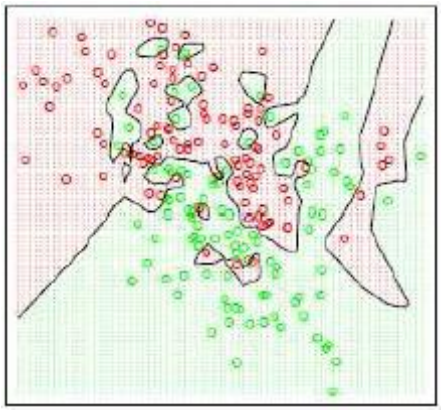
- 容易过度拟合数据(overfitting)
  - 忠实于每一个训练数据，包括噪声和错误数据



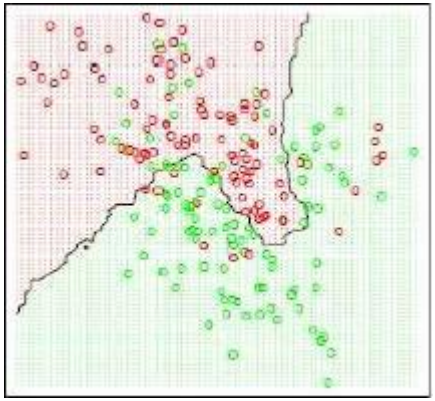
- 解决方案：1-近邻扩展到k-近邻（达到平滑分类边界的效果）

k-近邻分类器

- k-nearest neighbor classifier (kNN)
- 为新样本找到距离最近的k个样本点（k-近邻）
- 将其划分到k个最近邻中出现最多的类别（少数服从多数）



k=1



k=15

- 参数k：平滑或正则化（smoothing/regularization）参数

k-近邻分类器

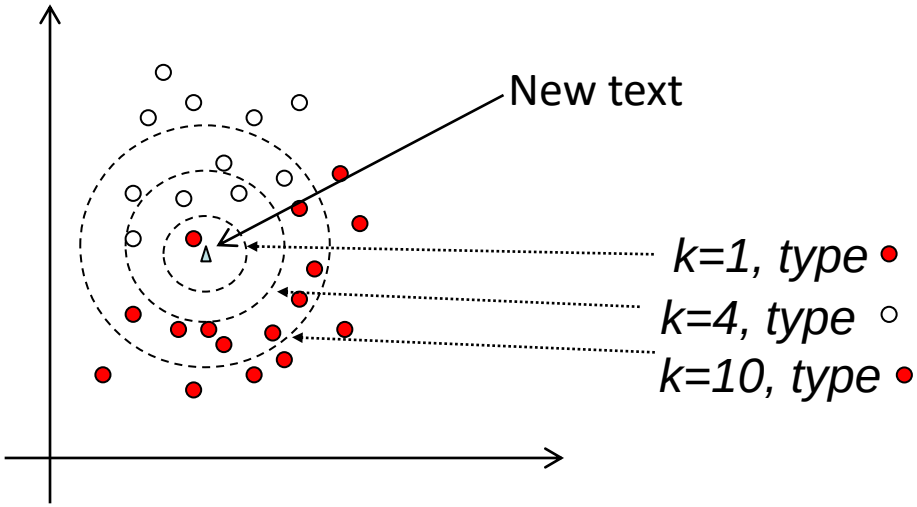
- 训练过程：
  - 对于每一个训练实例<x, y>，把这个样例加入training examples中
- 分类过程：
  - 给定一个要分类的实例x_q
  - 在training examples中选出最靠近x_q的k个实例，记为：$x_1, \ldots, x_k$
  - 返回：

$$\hat{y}(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} \delta(v, y(x_i))$$

    - 其中：
      - $V=\{v_1,\ldots,v_s\}$为有限的类别值集合
      - 如果a=b，则$\delta(a, b)=1$；否则$\delta(a, b)=0$

如何选择k?

- k值固定时，样本越多，越有可能找到x的正确类别



- k值过低？ k=1
- k值过高？ k=N

如何选择k?

- 采用验证数据（validation data）
  - 将训练数据分成两个不同的部分：训练数据、验证数据
  - 通过选择不同的k，得到不同的kNN分类器
  - 将不同的分类器用于分类验证数据
  - 选取具有最高分类性能的分类器对应的k值

距离测度

- 传统意义上，距离测度可以表示为一个函数，满足：

$$1.\ \mathbf{d(x,y)} \geq 0,\ \mathbf{d(x,y)} = \mathbf{0} \Leftrightarrow \mathbf{x} = \mathbf{y}$$

$$2.\ \mathbf{d(x,y)} + \mathbf{d(y,z)} \geq \mathbf{d(x,z)}$$

$$3.\ \mathbf{d(x,y)} = \mathbf{d(y,x)}$$

- 但对于聚类问题，采用的距离测度（或相似度）可以不严格满足以上三个约束

距离测度

几种常用的距离测度：

- Euclidean Distance:

$$d(x,y) = \sqrt{(x-y)^2} = \sqrt{(x-y)^T(x-y)} = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$$

L_2

- Manhattan Distance:

$$d(x,y) = |x-y| = \sum_{i=1}^{d}|x_i - y_i|$$

L_1

- Chebyshev Distance:

$$d(x,y) = \max_{1\le i\le d}|x_i - y_i|$$

L_∞

- 注意: 对于欧氏距离d(x,y)，$d^2(x,y)$ 不是一个测度（metric），但可用于距离度量（measure）（不满足三角不等式）

距离测度

几种常用的距离测度：

- Mahalanobis Distance:

$$\mathbf{d(x,y)} = \sqrt{(\mathbf{x}-\mathbf{y})^{\mathbf{T}}\Sigma^{-1}(\mathbf{x}-\mathbf{y})}$$

  - 其中 Σ 表示协方差矩阵

- 余弦距离：

$$\cos\theta = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^{\,2}}\sqrt{\sum_{i=1}^{n} y_i^{\,2}}}$$

其它问题

- 设计支持快速找到最近邻的数据结构
  - 树结构：较少计算量
  - 剪辑近邻、压缩近邻：减少存储量

kNN总结

- **Pros**:
  - 可以描述很复杂的分类边界 (非参数)
  - 训练快速 (仅需要构建数据结构)
  - 简单且好用 (e.g., 在computer vision中使用很多)
  - 模型结果可解释性好(通过近邻观察)
- **Cons**:
  - 存储开销大
  - 样本不均衡的影响
  - 参数空间很大时会导致搜索近邻很慢
  - 不是“最好”的分类器（性能上看）

Methods

I) Instance-based methods:
   1) Nearest neighbor
II) Probabilistic models:
   1) Naïve Bayes
   2) Maximum Entropy Model
   3) Neural Network
III) Linear Models:
   1) Linear Regression/Perceptron
   2) Support Vector Machine
IV) Decision Models:
   1) Decision Trees
   2) Random Forest

朴素贝叶斯模型

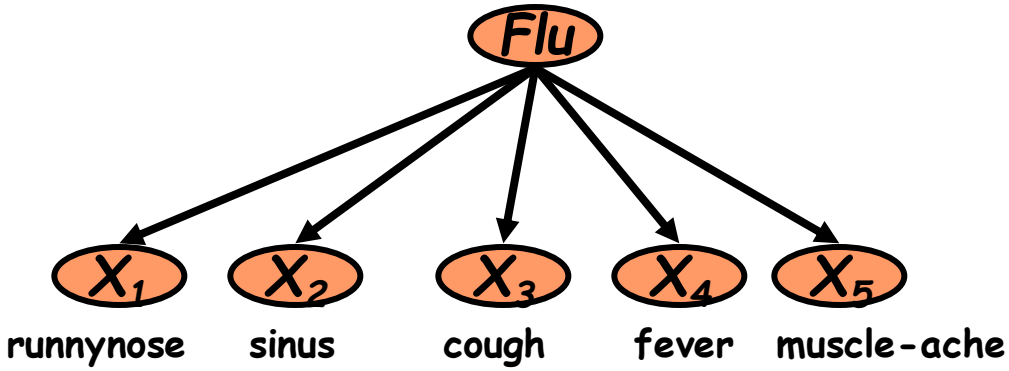
- **动机**：通过先验事件的知识来预测未来事件
- 贝叶斯定理：The Bayes Theorem

$$p(c_k \mid x_i) = \frac{p(x_i \mid c_k)\,p(c_k)}{p(x_i)}$$

  - $P(c_k)$: Prior probability of hypothesis c_k
  - $P(x_i)$: Prior probability of training data x_i
  - $P(c_k \mid x_i)$: Probability of c_k given x_i (posteriori)
  - $P(x_i \mid c_k)$: Probability of x_i given c_k (likelihood)

朴素贝叶斯假设

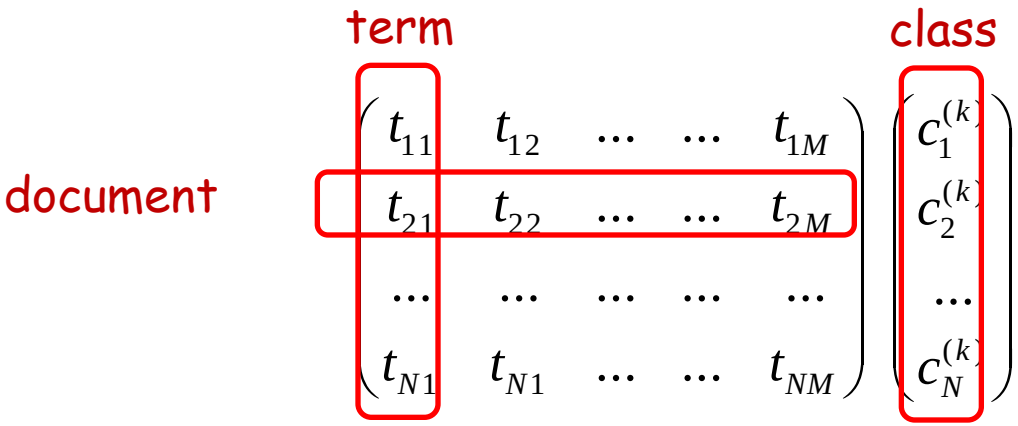
- **条件独立性：**
  - 给定类别C，变量$X_1,X_2...X_n$之间互相独立
  - 则$P(X_1,X_2...X_n|C) = P(X_1|C)\times P(X_2|C)\times...\times P(X_n|C)$



P(runny nose, sinus, cough, fever, muscle-ache | Flu) =
P(runny nose |Flu)×P(sinus | Flu)×…×P(muscle-ache | Flu)

朴素贝叶斯分类器

- e.g., spam detection:
- 训练数据：$D^{(k)} = \{(x_i, c_i^{(k)})\}$



朴素贝叶斯分类器

- 目标：给定email x_i，为其找到最可能的类别
- 由最大化后验概率（maximum a posteriori）：

$$\hat{c} = \arg\max_k \{P(c_k|x_i)\}$$

假设每个文档只属于一个类别

$$= \arg\max_k \left\{ \frac{P(c_k)P(x_i \mid c_k)}{P(x_i)} \right\}$$

Bayes法则

$$= \arg\max_k \{ P(c_k)P(x_i \mid c_k) \}$$

假设terms服从多项式分布且相互独立

$$= \arg\max_k \{ \log P(c_k) + \log P(x_i \mid c_k) \}$$

$$= \arg\max_k \left\{ \log P(c_k) + \log \prod_{j=1}^{M} P(t_j \mid c_k)^{n_{ij}} \right\}$$

where n_{ij} is the count of term t_j in document x_i.

Why?

朴素贝叶斯分类器

- $P(c_j)$
  - 可以根据训练语料中类别频率进行估计
- $P(t_1, t_2, \ldots, t_n | c_j)$
  - 参数规模：$O(|T|^n \bullet |C|)$
  - 依赖于规模巨大的训练数据，才可能对其进行有效估计
- 根据朴素贝叶斯假设：
  - 假设$P(t_1, t_2, \ldots, t_n | c_j) = \prod_{i=1,\ldots n} P(t_i | c_j)$
  - 参数规模： $O(|T| \bullet |C|)$
  - 意味着：
    - 类别与词的出现位置无关
    - 亦即：bag of words 模型

NBC：学习过程

- 给定训练数据，从中抽取出vocabulary，计算模型参数
  - 类别c_k的先验概率：

$$\hat{P}(c_k) = \frac{\#\text{ of training documents in } c_k}{N}$$

  - 已知类别c_k时terms t_j的出现概率：

$$\hat{P}(t_j \mid c_k) = \frac{\sum_{x_i \in c_k} n_{ij}}{\sum_{j=1}^{J} \sum_{x_i \in c_k} n_{ij}} \qquad j = 1, \cdots, M$$

→极大似然估计法（*maximum likelihood estimates*, MLE）

NBC：分类过程

- 计算测试文本x_i属于每个类别c_k的概率，返回概率最大时对应的类别

$$\hat{c} = \arg\max_{k=1,\ldots,K} \left\{ \hat{P}(c_k | x_i) \right\}$$

- 实际操作中：

$$f(c_k \mid x_i) = \underbrace{\log \hat{P}(c_k)}_{w_0^{(k)}} + \sum_{j=1}^{M} n_{ij} \underbrace{\log \hat{P}(t_j \mid c_k)}_{w_j^{(k)}} = w^{(k)} \cdot x_i$$

$$\text{Category profile} \quad w^{(k)} = (w_0^{(k)}, w_1^{(k)}, \cdots, w_M^{(k)})^T$$

$$\text{Input vector} \quad x_i = (1, n_{i1}, \cdots, n_{iM})^T$$

  - 最可能的类别：

$$\hat{c} = \arg\max_k \{ w^{(k)} \cdot x) \}$$

其它问题

- MLE的问题：

$$\hat{P}(t_5 = t \mid C = c) = \frac{N(t_5 = t, C = c)}{N(C = c)}$$

what if $N(t_5=t, C=c)$ is zero?

  - 数据稀疏（data sparseness）
  - 过拟合（overfitting）

- 属性为连续值时：如果属性t_i是一个连续值，如何计算$P(t_i \mid c)$?

模型参数的平滑

- Laplace:

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + k}$$

k: # of values of X_i

- Bayesian Unigram Prior:

$$\hat{P}(x_{i,k} \mid c_j) = \frac{N(X_i = x_{i,k}, C = c_j) + mp_{i,k}}{N(C = c_j) + m}$$

$p_{i,k}$: overall fraction in data where $X_i = x_{i,k}$

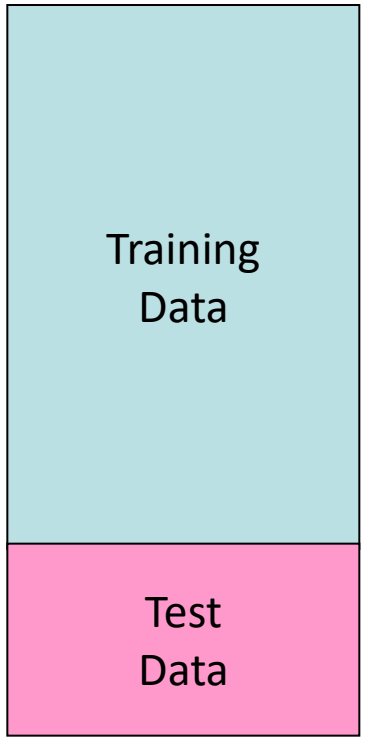
m: extent of "smoothing"

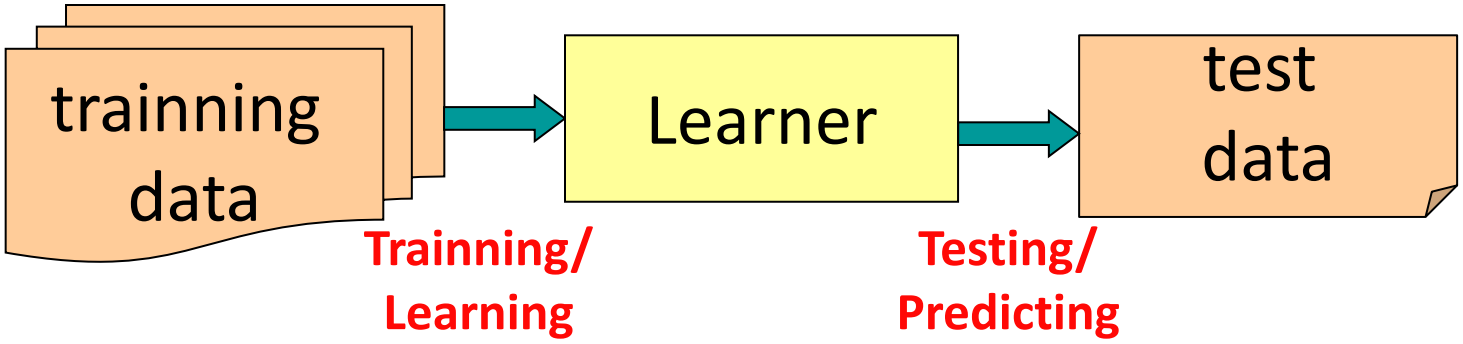
Summary for Naive Bayes

- **Pros**:
  - 原理简单
  - 易于处理大规模训练数据（只需要计数运算）
  - 实际应用中表现良好 (e.g. text classification)
- **Cons**:
  - 糟糕的条件独立性假设
  - 词汇位置无关假设
  - 不能融入较复杂的特征

Step 4: 评价

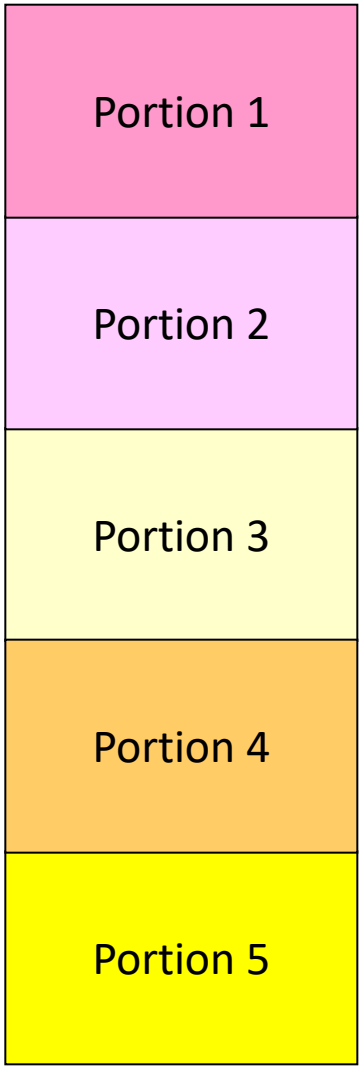
- 一般实验设置：
  - 数据集：有标记数据（例如标注了垃圾或正常标签的邮件）
  - 将数据分为训练数据和测试数据（不交叉）
    - 在训练数据上进行参数学习
    - 在测试数据上进行测试
    - 用在测试数据上的错误表示模型的错误





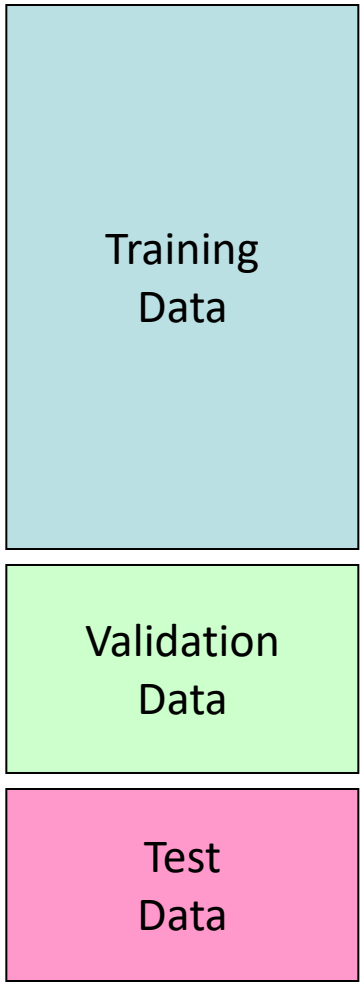
Step 4: 评价

- 另一种实验设置：n倍交叉验证（n-fold cross validation）
  - E.g., 5-fold cross validation
    - 把有标记数据等分为不交叉的5份，其中一份做测试，另外4份做训练
    - 独立地做5轮
    - 用5轮测试的平均性能作为模型的性能



Step 4: 评价

- 另一种实验设置：保留测试（hold-out test）
  - 在训练样本中学习模型参数
  - 通过验证集（发展集）来调整超参数（hyperparameters）
  - 在测试集中进行模型评价



Step 4: 评价

- 评价指标：precision and recall

困惑矩阵：

Category C_i		Expert Judgments YES	Expert Judgments NO
Classifier Judgments	YES	a (TP_i)	b (FP_i)
	NO	c (FN_i)	c (TN_i)

$$precision = \frac{a}{a+b} = \frac{TP_i}{TP_i + FP_i}, recall = \frac{a}{a+c} = \frac{TP_i}{TP_i + FN_i}$$

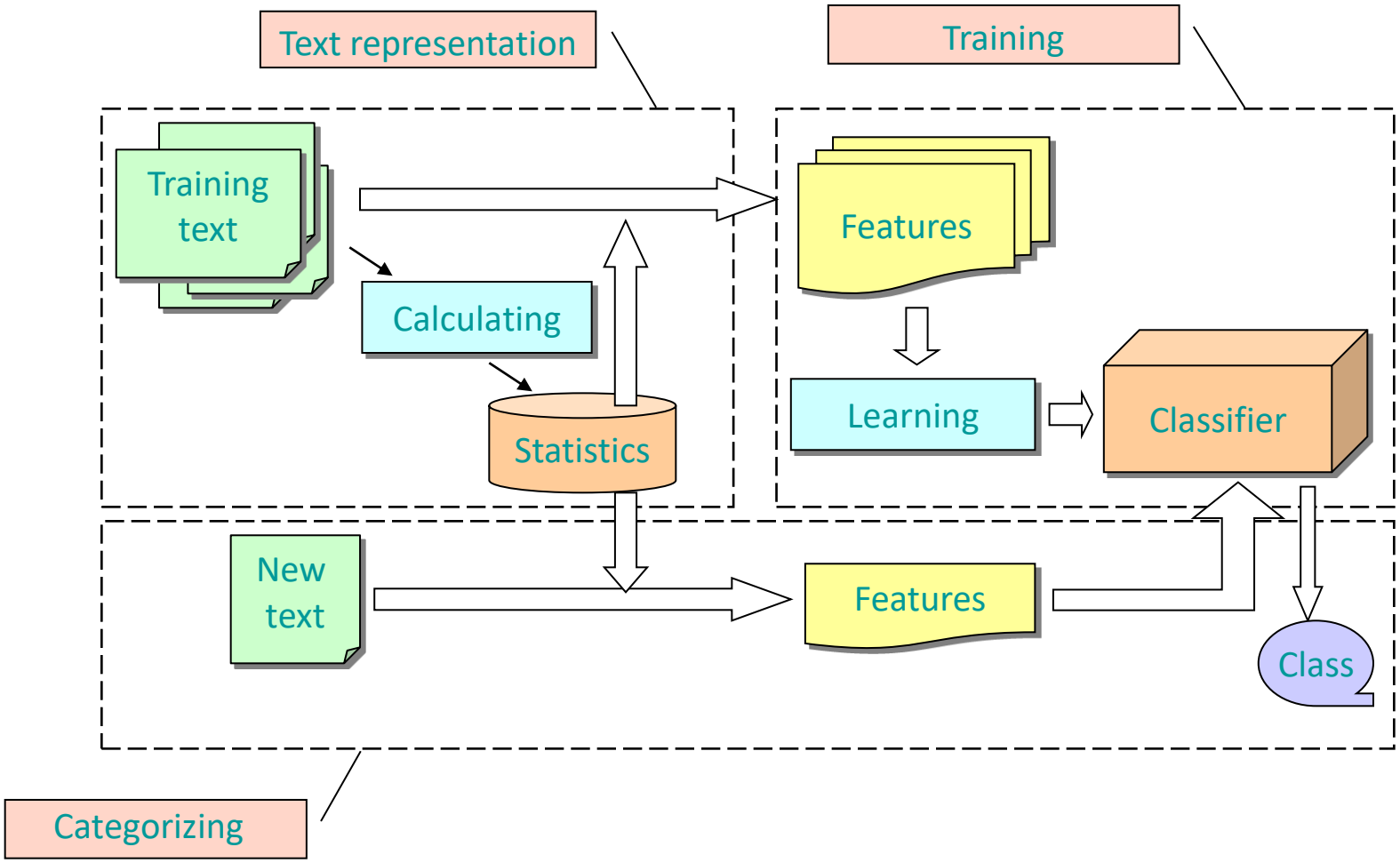
Step 4: 评价

- Presion: P = a/(a+b)
- Recall: R = a/(a+c)
- F measure: F_1 = 2PR/(P+R)
- Accuracy: Acc = (a+d)/(a+b+c+d)
- Miss = c/(a+c) = 1 - R

  (false negative)
- F/A = b/(a+b+c+d)

  (false positive)

Text Classification



Other issues

- Dimension Reduction (or feature selection)
- Text re-parameterization:
  - LSA (Latent Semantic Analysis, will be discussed later)

特征选择（降维）

- 文档频次(DF, document frency)
- 信息增益(IG, information gain)
- 互信息(MI, mutual information)
- 统计量CHI (x^2)
- 期望交叉熵(ECE, expected cross-entropy)
- 文本证据权(WET, Weight of Evidence for Text)
- 几率比(OR, Odds Ratio)

文档频次

- 文档频次(DF):
  - 文档频次是指有该词条出现的文档数量
  - n_t 是出现词条t的文档数量，N为总的文档数

$$DF(t) = \frac{n_t}{N}$$

  - 在训练文本集中对每个词条计算它的文档频次
  - 剔除在特征空间中文档频次小于预先定义的阈值的词条

信息增益

- 信息增益(IG):
  - 特征能够为分类系统带来多少信息，带来的信息越多（信息增益越大），该特征越重要
  - $P(C_i|t)$ 表示文本中出现特征t时，文本属于C_i的概率
  - $P(C_i|\bar{t})$ 表示文本中不出现单词t时，文本属于C_i的概率
  - $P(C_i)$ 表示类别出现的概率
  - P(t)表示t在整个文本训练集中出现的概率

$$IG(t) = P(t)\sum_{i=1}^{N} P(C_i|t)\log\frac{P(C_i|t)}{P(C_i)} + P(\bar{t})\sum_{i=1}^{N} P(C_i|\bar{t})\log\frac{P(C_i|\bar{t})}{P(C_i)}$$

互信息

- MI是信息论中的概念，用于度量一个消息中两个信号之间的相互依赖程度
- 在特征选择时，特征t和类别C 的互信息体现了特征与类别的相关程度
- 在某个类别C 中出现的概率高，而在其它类别中出现的概率低的特征t将获得较高的互信息
  - $P(C_i)$ 表示类别出现的概率
  - $P(t|C_i)$ 表示类C_i中词t出现的概率
  - P(t) 是词t出现的概率

$$MI(t) = \sum_{i=1}^{N} P(C_i) \log \frac{P(t|C_i)}{P(t)}$$

x^2 Statistic

- CHI (X^2 Statistic）：
  - 假设：特征t与文本类别C之间的非独立关系类似于具有一维自由度的x分布。
  - 在指定类别C的文本中出现频率高的词语和在其他类的文本中出现频率高的词语，对判断文章是否属于类别C都有帮助
  - A是特征t和第i类文档共同出现的频度
  - B是特征t出现而第i类文档不出现的频度
  - C是第i类文档出现而特征t不出现的频度
  - D是第i类文档和特征t都不出现的频度
  - N为总共的文本数

$$CHI(F)=\sum_i P(C_i)\cdot\chi^2(t,C_i)$$

$$\chi^2(t,C_i)=\frac{N\times(AD-BC)^2}{(A+C)\times(B+D)\times(A+B)(C+D)}$$

期望交叉熵

- 期望交叉熵(ECE, Expected Cross Entropy)：
  - 如果特征t和类别C 强相关，那么P(C_i|t)就大， 若P(C_i)又很小，则说明该词对分类的影响大
  - ECE反映了文本类别的概率分布和出现了某种特征的条件下文本类别的概率分布之间的距离
  - P(t)：词条t出现的概率
  - P(C_i|t)：词条t出现属于类C_i的概率
  - P(C_i)：类C_i出现的概率

$$ECE(t) = P(t)\sum_{i=1}^{N} P(C_i \mid t)\log\frac{P(C_i \mid t)}{P(C_i)}$$

文本证据权

$$WET(t) = P(t)\sum_{i=1}^{N} P(C_i)\cdot\left|\log\frac{P(C_i\mid t)\left(1-P(C_i)\right)}{P(C_i)\left(1-P(C_i\mid t)\right)}\right|$$

- 文本证据权(WET，the Weight of Evidence for Text):
  - WET比较了$P(C_i)$与$P(C_i|t)$之间的差别
  - 如果t和类别强相关，即$P(C_i|t)$大，并且相应类别出现的概率小，说明t对分类的影响大，计算出来的函数值就大，可以选取作为特征项；反之，就不选其作为特征项

几率比

- 几率比(OR，Odds Ratio)：
  - 几率比法考察本类别和其它所有类别的差异，将其它类别全部看做负样本

$$OR(t)=\log\frac{P(t|C_{pos})(1-P(t|C_{neg}))}{P(t|C_{neg})(1-P(t|C_{pos}))}$$

  - C_{pos}表示正样本的情况，C_{neg}表示负样本的情况

Next lecture

- Text Clustering