



自然语言处理

L3: 朴素贝叶斯

第3章

CONTENTS

01

朴素贝叶斯分类器

02

训练朴素贝叶斯分类器

03

实例分析

04

评估



文本分类任务

垃圾邮件识别

匿名作者识别

文章主题识别

电影评论分类

.....

还有哪些？

情感分析(Sentiment Analysis)

为什么要情感分析？

电影是否好看？

产品是否好用？

社交媒体评论？

市场规律预测？

情感状态有哪些类别？

情感 emotion : 对事件的当下感受和评价

- angry, sad, joyful, fearful, ashamed, proud, elated

情绪 mood : 有弥漫性并且具有低强度长时间的主观感受

- cheerful, gloomy, irritable, listless, depressed, buoyant

人际立场 interpersonal stances : 与他人进行互动时的情感立场

- friendly, flirtatious, distant, cold, warm, supportive, contemptuous

态度 attitude : 持久且带情感色彩的信念，或对人对事的情感倾向

- liking, loving, hating, valuing, desiring

人格特质 Personality traits : 稳定的人格倾向和行为倾向

- nervous, anxious, reckless, morose, hostile, jealous

文本分类任务的定义

- *Input:*
 - a document d
 - a fixed set of classes $C = \{c_1, c_2, \dots, c_J\}$
- *Output:* a predicted class $c \in C$

分类方法：hand-crafted 规则

使用单词的一些组合或者其他特征作为规则，比如：

- 垃圾邮件识别 – 黑名单邮箱地址或者(“dollars” AND “you have been selected”)

识别的准确率可以很高

- 若由相关领域的专家仔细地定义识别规则

但建立和维持这些规则成本很高

- 为什么？

分类方法：有监督的机器学习

- *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 $\gamma: d \rightarrow c$

分类方法：有监督的机器学习

- 分类器
 - 朴素贝叶斯 Naïve Bayes
 - 逻辑回归 Logistic regression
 - 神经网络 Neural networks
 - K近邻 k-Nearest Neighbors
 - ...



Part.01

朴素贝叶斯分类器



什么是朴素贝叶斯分类器

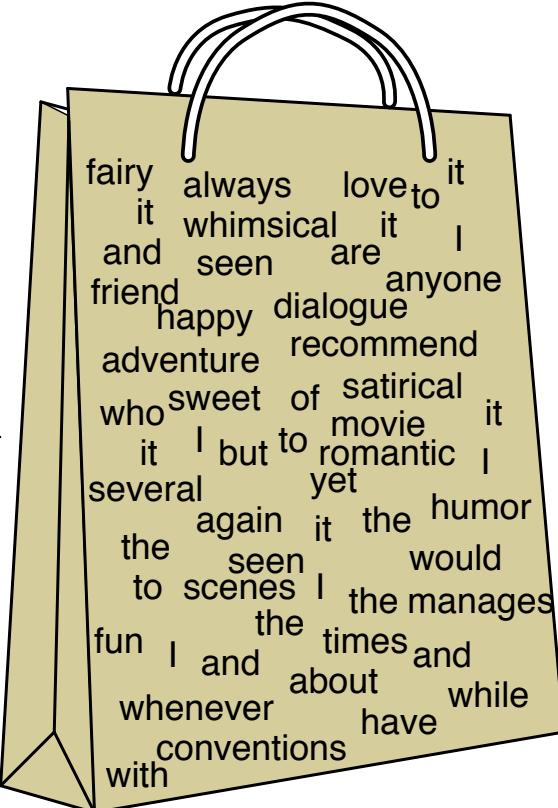
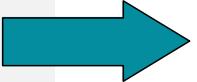
基于贝叶斯规则的简易（朴素或 naive）分类方法

依赖于较为简单的文本表达：

- 词袋 bag of words

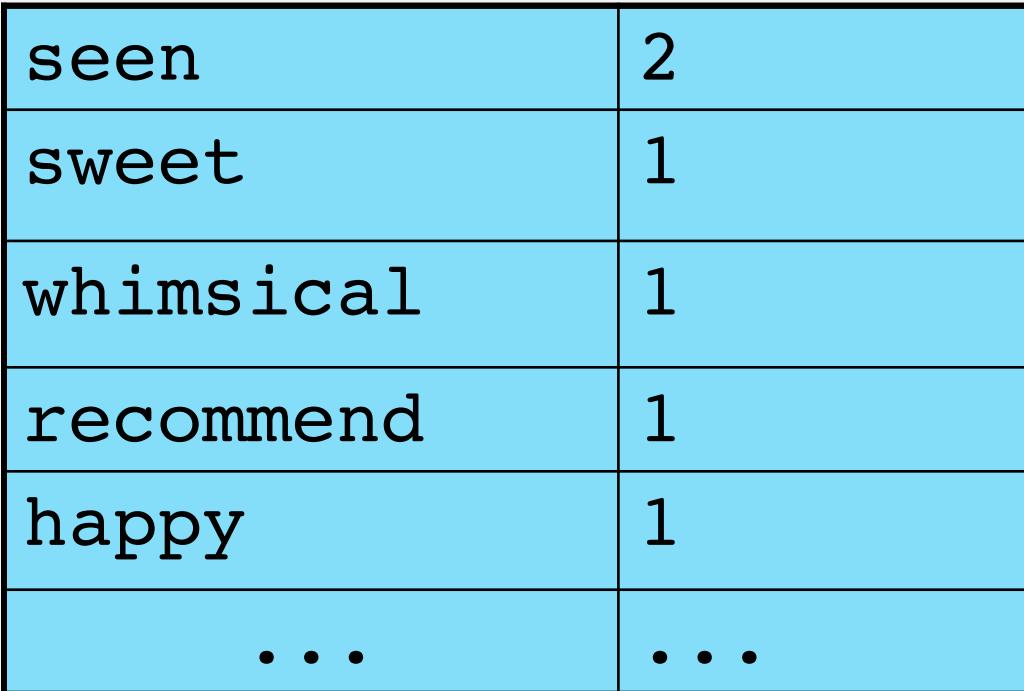
词袋表达

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!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

词袋表达

Y() = C

seen	2
sweet	1
whimsical	1
recommend	1
happy	1
...	...



在文档和类别中使用贝叶斯规则

- 文档 d and 类别 c

$$P(c | d) = \frac{P(d | c)P(c)}{P(d)}$$

朴素贝叶斯分类器(一)

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(c | d)$$

MAP is “maximum a posteriori” = most likely class

$$= \operatorname{argmax}_{c \in C} \frac{P(d | c)P(c)}{P(d)}$$

Bayes Rule

$$= \operatorname{argmax}_{c \in C} P(d | c)P(c)$$

Dropping the denominator

朴素贝叶斯分类器(二)

"Likelihood" "Prior"

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(d | c)P(c)$$

$$= \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c)P(c)$$

Document d
represented as
features x_{1..n}

朴素贝叶斯分类器(三)

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c)P(c)$$

Could only be estimated if a very, very large number of training examples was available.

How often does this class occur?

We can just count the relative frequencies in a corpus

多项式朴素贝叶斯独立假设

$$P(x_1, x_2, \dots, x_n | c)$$

词袋假设：假设不考虑单词的位置

条件独立：在给定类型c时，假设特征概率 $P(x_i|c_j)$ 是独立的

$$P(x_1, \dots, x_n | c) = P(x_1 | c) \bullet P(x_2 | c) \bullet P(x_3 | c) \bullet \dots \bullet P(x_n | c)$$

多项式朴素贝叶斯分类器

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c)P(c)$$

$$c_{NB} = \operatorname{argmax}_{c \in C} P(c_j) \prod_{x \in X} P(x | c)$$

使用多项式朴素贝叶斯分类器进行文本分类

`positions ← all word positions in test document`

$$c_{NB} = \operatorname{argmax}_{c_j \in C} P(c_j) \prod_{i \in positions} P(x_i | c_j)$$

多个概率相乘时会出现的问题

$$c_{NB} = \operatorname{argmax}_{c_j \in C} P(c_j) \prod_{i \in positions} P(x_i | c_j)$$

当有许多概率相乘时，结果会无限趋近于零(浮点下溢)

有什么方法？

使用对数函数 \log , 因为 $\log(ab) = \log(a) + \log(b)$

也就是将概率相乘转化为概率相加

对数空间

原分类器

$$c_{NB} = \operatorname{argmax}_{c_j \in C} P(c_j) \prod_{i \in positions} P(x_i | c_j)$$

使用对数函数后的分类器

$$c_{NB} = \operatorname{argmax}_{c_j \in C} \left[\log P(c_j) + \sum_{i \in positions} \log P(x_i | c_j) \right]$$

注意：

1): 使用对数函数后，不会改变原有的类别排序

- 有最高概率的类别依旧具有最高对数概率的性质

2): 线性模型

- 因为将所有权重进行了相加操作，因此朴素贝叶斯模型是线性的



Part.02

训练朴素贝叶斯分类器

训练朴素贝叶斯分类器

最大似然估计

- 使用数据中的频率统计

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}}$$

$$\hat{P}(w_i | c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}$$

参数估计

$$\hat{P}(w_i \mid c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}$$

在有关话题 c_j 的文档中，
单词 w_i 出现次数与所有单词的比例

整合所有主题为 j 的文档为一个大型文档库

- 使用单词 w 在文档库中的频率

最大似然估计的问题

假设某个训练文档没有出现单词 `fantastic`, 但被分类为 `positive` 话题, 该怎么办?

$$\hat{P}("fantastic" \mid \text{positive}) = \frac{\text{count}("fantastic", \text{positive})}{\sum_{w \in V} \text{count}(w, \text{positive})} = 0$$

无论其他证据如何, 都不能以零概率为条件!

$$c_{MAP} = \operatorname{argmax}_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

朴素贝叶斯+拉普拉斯平滑

$$\begin{aligned}\hat{P}(w_i \mid c) &= \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} (\text{count}(w, c) + 1)} \\ &= \frac{\text{count}(w_i, c) + 1}{\left(\sum_{w \in V} \text{count}(w, c) \right) + |V|}\end{aligned}$$

多项式朴素贝叶斯训练

从训练语料库中， 提取词汇表(*Vocabulary*)

- Calculate $P(c_j)$ terms
 - For each c_j in C do
$$docs_j \leftarrow \text{all docs with class } = c_j$$
- Calculate $P(w_k | c_j)$ terms
 - $Text_j \leftarrow \text{single doc containing all } docs_j$
 - For each word w_k in *Vocabulary*
$$n_k \leftarrow \# \text{ of occurrences of } w_k \text{ in } Text_j$$

$$P(w_k | c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$

未知单词(unknown words)

什么是未知单词？

- 没有出现在训练集和收集的词汇表中
- 但出现在测试集中

如何解决？

- 忽视它们？如何做？
 - 从测试集中删除
 - 假设不存在
 - 不分配概率给这些单词
- 建立一个支持未知单词的模型？
 - 没有帮助，即使知道哪个类有更多未知单词

停用词(stop words)

很多模型都会忽略停顿词

- 停顿词非常频繁地出现，比如 the 和 a
- 如何去除停顿词？
 - 按照单词频率从高到低排序词汇表
 - 将前10或前50最高频率的单词列入停顿词表
 - 从训练集和测试集中删除这些单词

然而去除停顿词实际上没用

- 朴素贝叶斯模型使用所有单词，并不会忽视停顿词



Part.03

朴素贝叶斯实例

二项朴素贝叶斯情感分类

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

二项朴素贝叶斯情感分类

	Cat	Documents
Training	-	just plain boring entirely predictable and lacks energy no surprises and very few laughs + very powerful + the most fun film of the summer
Test	?	predictable with no fun

3. Likelihoods from training:

$$p(w_i|c) = \frac{\text{count}(w_i, c) + 1}{(\sum_{w \in V} \text{count}(w, c)) + |V|}$$

$$P(\text{"predictable"}|-) = \frac{1+1}{14+20} \quad P(\text{"predictable"}|+) = \frac{0+1}{9+20}$$

$$P(\text{"no"}|-) = \frac{1+1}{14+20} \quad P(\text{"no"}|+) = \frac{0+1}{9+20}$$

$$P(\text{"fun"}|-) = \frac{0+1}{14+20} \quad P(\text{"fun"}|+) = \frac{1+1}{9+20}$$

1. Prior from training:

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}} \quad \begin{aligned} P(-) &= 3/5 \\ P(+) &= 2/5 \end{aligned}$$

2. Drop "with"

4. Scoring the test set:

$$P(-)P(S|-) = \frac{3}{5} \times \frac{2 \times 2 \times 1}{34^3} = 6.1 \times 10^{-5}$$

$$P(+)P(S|+) = \frac{2}{5} \times \frac{1 \times 1 \times 2}{29^3} = 3.2 \times 10^{-5}$$

优化情感分类

在情感分析任务中，单词的出现比单词的频率更加重要

- 单词 `fantastic` 出现一次即可以提供信息。
- 单词 `fact` 出现四次没有提供太多信息。

二项式情感分类

- 将单词计数限制为1
- 不同于伯努利朴素贝叶斯

- Calculate $P(c_j)$ terms

- For each c_j in C do

$docs_j \leftarrow$ all docs with class = c_j

$$P(c_j) \leftarrow \frac{|docs_j|}{\text{total \# documents|}}$$

- Calculate $P(w_k | c_j)$ terms

- ~~Remove singletons containing all $docs_j$~~

- ~~For each word type w_k in $Vocabulary$~~

$n_k \leftarrow$ # of occurrences of w_k in $Text_j$

$$P(w_k | c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$

训练二项式朴素贝叶斯

从训练语料库中，提取词汇表(*Vocabulary*)

- Calculate $P(c_j)$ terms

- For each c_j in C do

$docs_j \leftarrow$ all docs with class = c_j

$$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$

- Calculate $P(w_k | c_j)$ terms

- Remove singletons containing all $docs_j$

- For each word type w_k in *Vocabulary*

$n_k \leftarrow$ # of occurrences of w_k in $Text_j$

$$P(w_k | c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$

应用二项式朴素贝叶斯到一个测试集d中

首先对d中所有单词做去重

然后使用下面的公式计算NB：

$$c_{NB} = \operatorname{argmax}_{c_j \in C} P(c_j) \prod_{i \in positions} P(w_i | c_j)$$

二项式朴素贝叶斯

Four original documents:

- it was pathetic the worst part was the boxing scenes
- no plot twists or great scenes
- + and satire and great plot twists
- + great scenes great film

After per-document binarization:

- it was pathetic the worst part boxing scenes
- no plot twists or great scenes
- + and satire great plot twists
- + great scenes film

	NB Counts		Binary Counts	
	+	-	+	-
and	2	0	1	0
boxing	0	1	0	1
film	1	0	1	0
great	3	1	2	1
it	0	1	0	1
no	0	1	0	1
or	0	1	0	1
part	0	1	0	1
pathetic	0	1	0	1
plot	1	1	1	1
satire	1	0	1	0
scenes	1	2	1	2
the	0	2	0	1
twists	1	1	1	1
was	0	2	0	1
worst	0	1	0	1

Counts can still be 2! Binarization is within-doc!



情感分类：应对否定

I really like this movie

I really **don't** like this movie

加入 **don't** 使得原本正面的like成为了负面的

同时加入否定，也可以使得负面成为正面

- **Don't** dismiss this film
- **Doesn't** let us get bored

情感分类：应对否定

Das, Sanjiv and Mike Chen. 2001. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In Proceedings of the Asia Pacific Finance Association Annual Conference (APFA).

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. EMNLP-2002, 79—86.

Simple baseline method:

Add NOT_ to every word between negation and following punctuation:

didn't like this movie , but I



didn't NOT_like NOT_this NOT_movie but I



情感分类：词典(Lexicons)

很多时候没有充足的训练数据集

在这种情况下，可以使用预先建立的词列表-Lexicons

有很多公开发布的词典

MPQA 主观意向词典

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. Proc. of HLT-EMNLP-2005.

Riloff and Wiebe (2003). Learning extraction patterns for subjective expressions. EMNLP-2003.

- 主页: https://mpqa.cs.pitt.edu/lexicons/subj_lexicon/
- 6885 words from 8221 lemmas, 并且标记有强度(strong/weak)
 - 2718 positive 正面的
 - 4912 negative 负面的
- + : *admirable, beautiful, confident, dazzling, ecstatic, favor, glee, great*
- - : *awful, bad, bias, catastrophe, cheat, deny, envious, foul, harsh, hate*

The General Inquirer

Philip J. Stone, Dexter C Dunphy, Marshall S. Smith, Daniel M. Ogilvie. 1966. The General Inquirer: A Computer Approach to Content Analysis. MIT Press

- 主页: <http://www.wjh.harvard.edu/~inquirer>
- 类型列表: <http://www.wjh.harvard.edu/~inquirer/homecat.htm>
- 表格(Spreadsheet): <http://www.wjh.harvard.edu/~inquirer/inquirerbasic.xls>
- 类型包括:
 - Positive (1915 words) and Negative (2291 words)
 - Strong vs Weak, Active vs Passive, Overstated versus Understated
 - Pleasure, Pain, Virtue, Vice, Motivation, Cognitive Orientation, etc
- 对研究用途的使用是免费的

情感分类中使用词典Lexicons

添加一项属性，即当一个单词出现在词典中那么获取一个计数
例如，该单词出现在正面词典或者负面词典时，给予该单词这一属性。

那么所有正面或者负面的单词都获得了这一属性。

使用1-2属性不如使用所有单词

- 但是当训练数据稀疏或不代表测试集时，密集的词典特征可以提供帮助

朴素贝叶斯：垃圾邮件过滤

垃圾邮件的特点：

- 会提到 millions of (dollar) ((dollar) NN,NNN,NNN.NN)
- 邮件来源用很多数字开头
- 主题都是大写
- HTML中的文字图片比很小
- 会提到“百分之百保证”
- 声明可以从列表中删除



朴素贝叶斯：不同语言

确定一段文字是用哪种语言写的。

基于字符N-grams的特征可以很好的完成识别任务

对多类型的一些语言进行训练是很重要的

- 例如美式英语包括非裔美式英语，英语还有很多类型比如印式英语

总结：朴素贝叶斯不那么朴素

速度快，存储消耗少

在小样本数据集上的效果很好

对无关属性具有鲁棒性

无关的属性会彼此抵消而不会影响结果

在一些属性重要性程度相当的领域中，表现很好

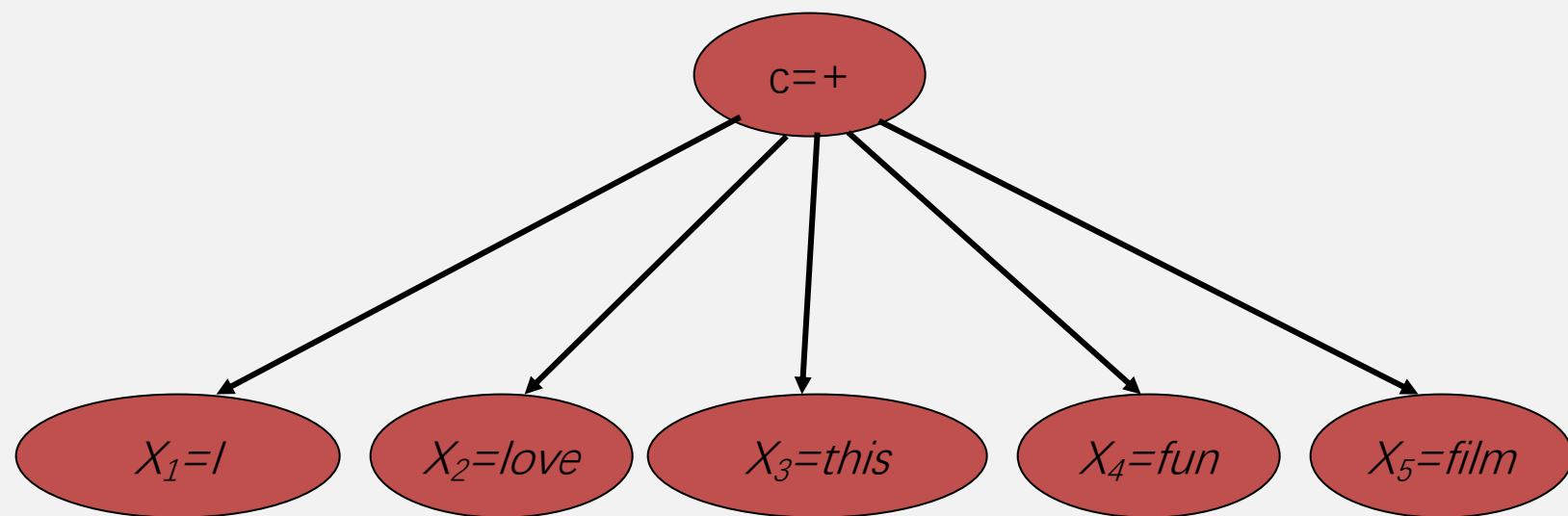
在一些情况下，尤其在数据很少的情况下，决策树会遭遇碎片化

若独立性假设成立，则为最优

可用做文本分类任务实验对比时的一个基准

- 其他分类器可以输出更好的结果

多项式朴素贝叶斯生成模型



朴素贝叶斯和语言模型

朴素贝叶斯分类器可以使用很多特征：

- URL, 邮箱地址, 字典, 基于网络的属性特征

之前仅使用了文本中的所有单词作为特征

朴素贝叶斯和语言模型有一个很重要的相似点

一个类 == 一个一元语言模型

Sec.13.2.1

- 给每个单词分配一个概率: $P(\text{word} | c)$
- 给每个句子分配一个概率: $P(s|c) = \prod P(\text{word}|c)$

Class *pos*

0.1	I		I	love	this	fun	film
0.1	love		0.1	0.1	.05	0.01	0.1
0.01	this						
0.05	fun						
0.1	film						

$P(s | pos) = 0.0000005$

...

使用朴素贝叶斯作为一个语言模型

- 哪一类可以赋予 s 以更大的概率？

Model pos		Model neg					
0.1	I	0.2	I	<u>I</u>	<u>love</u>	<u>this</u>	<u>fun</u>
0.1	love	0.001	love	0.1	0.1	0.01	0.05
0.01	this	0.01	this	0.2	0.001	0.01	0.005
0.05	fun	0.005	fun				0.1
0.1	film	0.1	film				

$P(s|pos) > P(s|neg)$



Part.04

二分类评估

评估

考虑一个二分类任务

一个甜品销售经理，想知道人们对甜品的评价

发布一个有关该甜品的说说或者微博

- 正面 有关该甜品积极的评价
- 负面 所有其他的评价

2X2混合矩阵

		<i>gold standard labels</i>		
		gold positive	gold negative	
<i>system output labels</i>	system positive	true positive	false positive	$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}$
	system negative	false negative	true negative	
		$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}$		
				$\text{accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{fp} + \text{tn} + \text{fn}}$

评估：准确率Accuracy

为什么不使用Accuracy作为评估标准？

假设有100万条微博评论

- 100条是有关对该甜品正面的评价
- 99.9万条是其他的(包括负面的， 中性的， 无关的等)

若我们对所有标记为“无关该甜品”类别的评论建立分类器，

- 那么准确率Accuracy是99.99%
- 因此我们要使用精度(Precision)和召回率(recall)

评估：精度Precision

所有"正确被检索的结果(TP)"占所有"实际被检索到的(TP+FP)"的比例.

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

评估：召回率Recall

所有"正确被检索的结果(TP)"占所有"应该检索到的结果(TP+FN)"的比例.

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

为什么要使用精度和召回率

避免先前accuracy = 99.99% 情况的出现

强调true positives

- 也就是找到想要找到的items

评估：F-measure

F-measure 是精度和召回率的调和均值

$$F_{\beta} = \frac{(\beta^2 + 1)PR}{\beta^2P + R}$$

通常使用F1-score作为度量标准，即当 $\beta = 1$ 时

$$F_1 = \frac{2PR}{P + R}$$

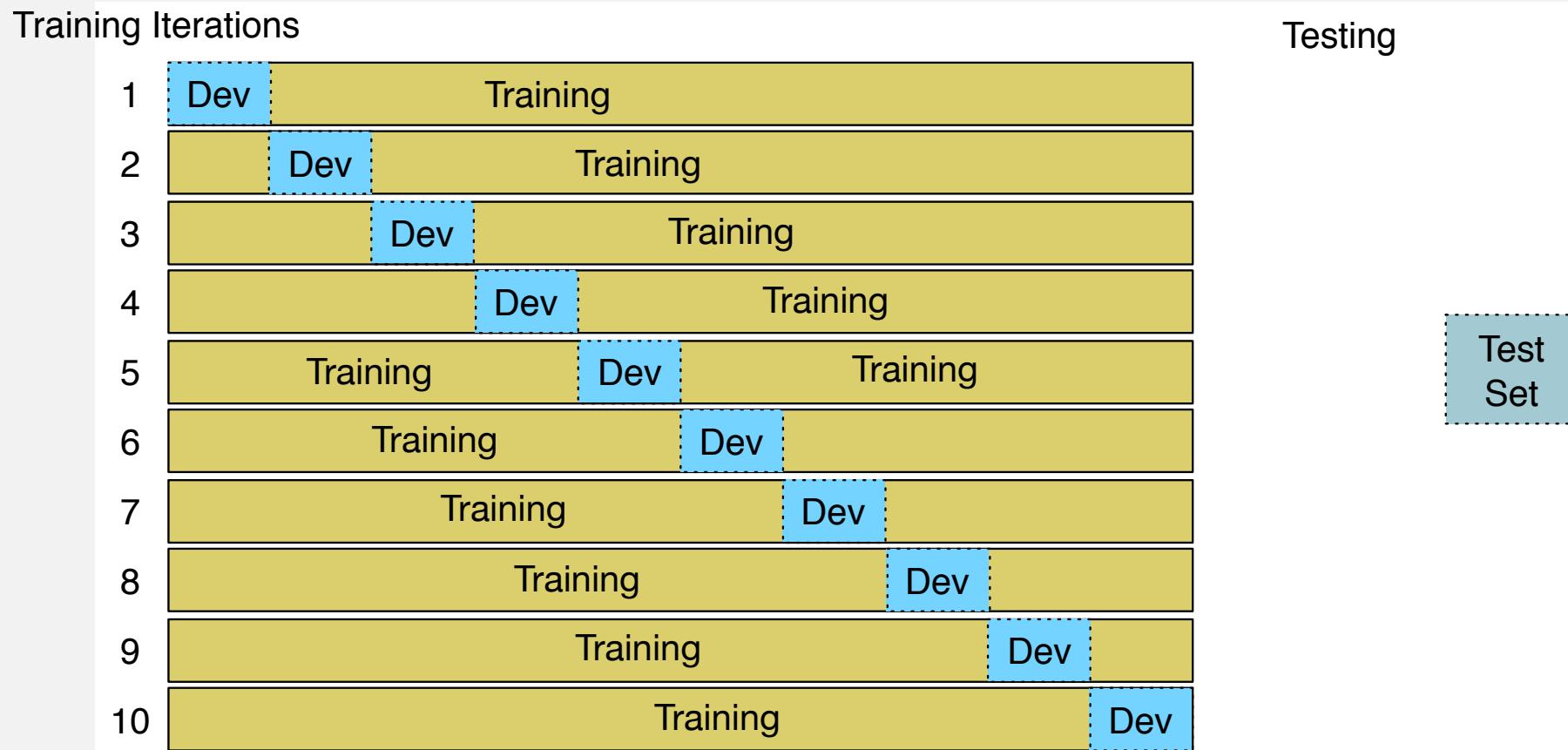
开发集(Devset)和交叉验证(cross validation)



在训练集上学习， 在开发集上调整， 在测试集上验证

- 帮助避免过拟合
- 更保守的性能估计
- 如何划分训练集和开发集？

交叉验证：多次划分





Part.04

多分类评估

三分类混合矩阵

		gold labels			
		urgent	normal	spam	
system output	urgent	8	10	1	$\text{precision}_u = \frac{8}{8+10+1}$
	normal	5	60	50	$\text{precision}_n = \frac{60}{5+60+50}$
	spam	3	30	200	$\text{precision}_s = \frac{200}{3+30+200}$
		$\text{recall}_u = \frac{8}{8+5+3}$	$\text{recall}_n = \frac{60}{10+60+30}$	$\text{recall}_s = \frac{200}{1+50+200}$	

如何在三分类任务中组合P和R为一个评估标准

宏平均(macroaveraging):

- 计算每一类的性能评估
- 取所有类结果的均值

微平均(microaveraging):

- 收集所有类的结果到混合矩阵中
- 从矩阵表中计算P和R

宏平均和微平均

Class 1: Urgent		Class 2: Normal		Class 3: Spam		Pooled						
		true	true	true	true	true	true					
system	urgent	urgent	not	system	normal	not	system	spam	not	system	yes	no
urgent	8	11		60	55		200	33		268	99	
not	8	340		40	212		51	83		99	635	
precision = $\frac{8}{8+11} = .42$		precision = $\frac{60}{60+55} = .52$		precision = $\frac{200}{200+33} = .86$		microaverage precision = $\frac{268}{268+99} = .73$						
macroaverage precision = $\frac{.42+.52+.86}{3} = .60$												

统计上的显著性检验

如何评估那个分类器更好？

假设两个分类器A和B：

- 标准 M ： $M(A,x)$ 代表分类器A在测试集 x 上的性能
- $\delta(x)$ ：代表A和B在 x 上的性能差异
 - $\delta(x) = M(A,x) - M(B,x)$
- 当 $\delta(x) > 0$ 时，A优于B
- $\delta(x)$ 称为效应量(effect size)
- 当 $\delta(x)$ 为正数时，就结束了吗？
 - 也可能是由发生在测试集或者实验环境中的意外导致的

统计上的假设检验

有两个假设：

- 零假设：A没有B好
- A比B好

$$H_0 : \delta(x) \leq 0$$
$$H_1 : \delta(x) > 0$$

如何排除零假设？

- 从测试集数量的范围内设定一个随机变量X
- 如果 H_0 是真，那么我们看到的 $\delta(x) > 0$ 的情况是否会在其他测试集中？这种情况发生的可能性有多大？
 - 形式化定义为P值：

$$P(\delta(X) \geq \delta(x) | H_0 \text{ is true})$$

统计上的假设检验

$$P(\delta(X) \geq \delta(x) | H_0 \text{ is true})$$

在上述例子中，P值表示在假设 H_0 成立的情况下 $\delta(x)$ 发生的概率

- 若 H_0 成立但 $\delta(x)$ 的值很大，说明P值很低

若P值很小(0.05 or 0.01),

- 那么表示在零假设成立的情况下，很难再次观测到 $\delta(x)$ ，可以否定零假设

若观测到 δ 的概率小于阈值，那么说明 ‘A优于B’ 的结果在统计上是显著的

- 因此否定零假设

统计上的假设检验

- 如何计算这个概率？
- 在NLP中，不常用参数检验(如：t检验)
- 而是常用基于抽样样本的非参数检验：
- 人为建立多个版本

- 假设建立无穷多个测试集 x'
 - 在每个测试集上测量 $\delta(x)$ 的值
 - 获得一个分布
 - 设置一个阈值0.01
 - 若在99%的测试集上出现 $\delta(x) > \delta(x')$
 - 得出结论在原始测试集中得出的结果是真的，不是人为假设的。

统计上的假设检验

两种常用方法：

- 近似随机化(approximate randomization)
- 自助抽样检验(bootstrap test)

配对检验：

- 对比两组观察，其中一组中的每一个观察都可以匹配到另一组的某一个观察。
- 例如两个分类器A和B在测试集上的性能匹配，是否有同样的观察？



自助抽样检验(The Bootstrap Test)

Efron and Tibshirani, 1993

定义：使用近似抽样分布来确定统计上的显著性的一种随机化检验。

工作原理：从整体样本集中有放回地重复抽取较小的样本集。

是否可以用于准确率，精度，召回率，F-measure的计算中？



自助抽样检验(The Bootstrap Test)实例

课后学习：从网上找一些实例了解该检验工作原理



谢谢

立自苦取



信息工程学院
(人工智能学院)