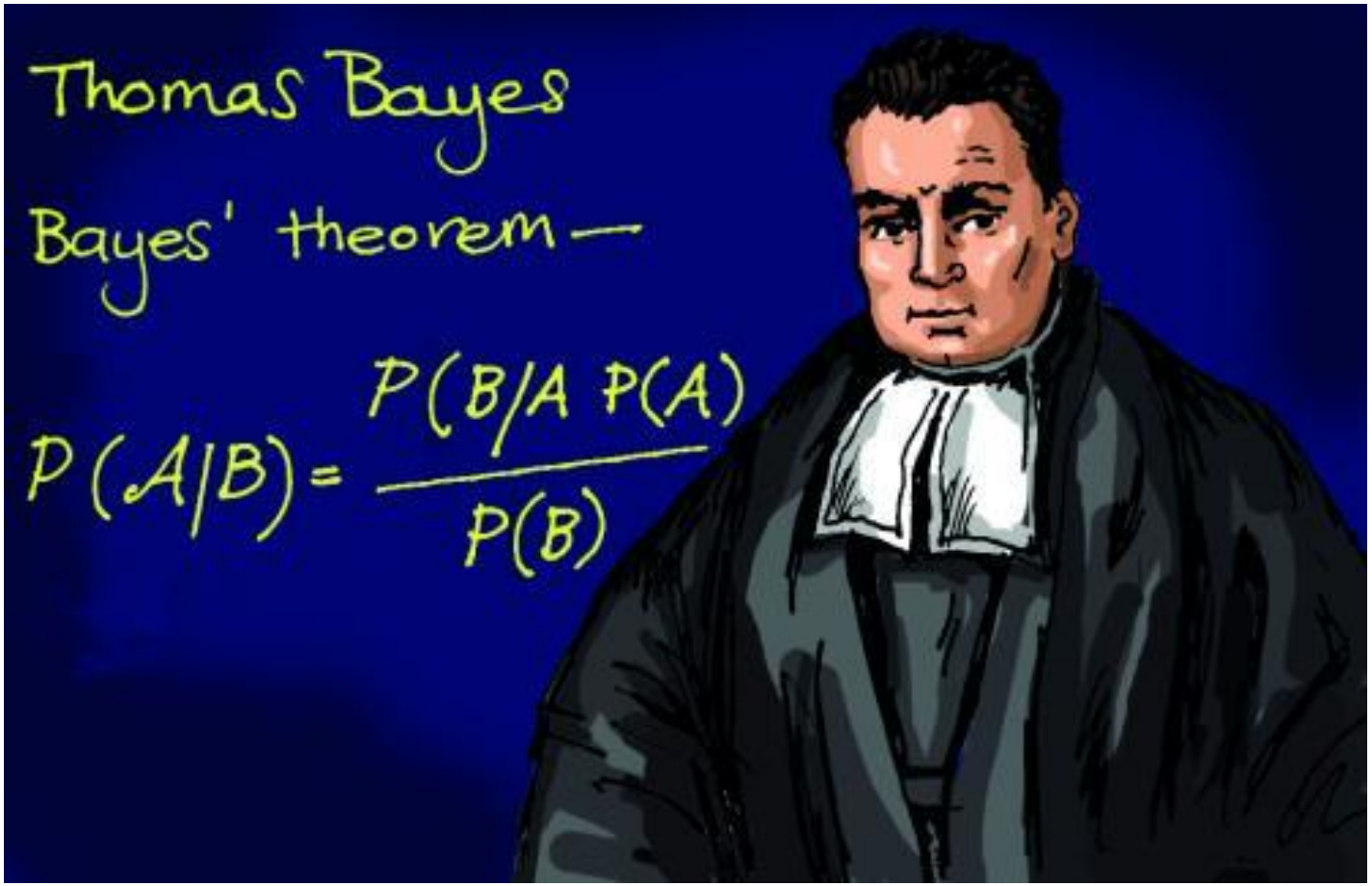
贝叶斯理论



内容提要

- 1、理论的基本内容
- 2、相关论文中的应用
- 3、应用领域综述
- 4、几个实验实验
- 5、总结分析

About Courses

1、Naive Bayesian

2、The Expression of Bayesian Networks

Markov Model

3、D-separation

The Three Models of Conditional Independence

Markov Blanket

4、Networks & MSWT

Bayes公式与定理

贝叶斯定理便是基于下述贝叶斯公式：

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

上述公式的推导其实非常简单，就是从条件概率推出。

根据条件概率的定义，在事件B发生的条件下事件A发生的概率是

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

同样地，在事件A发生的条件下事件B发生的概率

$$P(B|A) = \frac{P(A \cap B)}{P(A)}.$$

整理与合并上述两个方程式，便可以得到：

$$P(A|B)\,P(B) = P(A \cap B) = P(B|A)\,P(A).$$

接着，上式两边同除以P(B)，若P(B)是非零的，我们便可以得到**贝叶斯定理**的公式表达式：

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}.$$

http://blog.csdn.net/v_july_v/article/details/40984699

贝叶斯思维

- 贝叶斯推断只是简单地在考虑了新的证据后，更新你的信念。
- 贝叶斯主义者很少对于一个结果很肯定，但是他们可以对某件事有一定的信心。

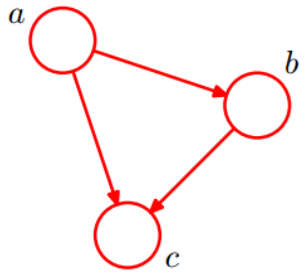
- 就像调BUG，

朴素贝叶斯的假设

- 一个特征出现的概率，与其他特征(条件)独立(特征独立性)
  - 其实是：对于给定分类的条件下，特征独立
- 每个特征同等重要(特征均衡性)

Bayesian network

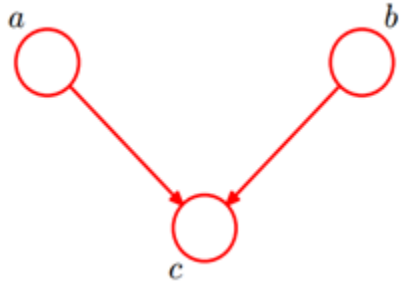
- 贝叶斯网络(Bayesian network)，又称信念网络(Belief Network)，或有向无环图模型(directed acyclic graphical model)，是一种概率图模型，于1985年由Judea Pearl首先提出。它是一种模拟人类推理过程中因果关系的不确定性处理模型，其网络拓朴结构是一个有向无环图(DAG)。



$$p(a,b,c) = p(c|a,b)p(b|a)p(a)$$

贝叶斯网络三种形式

- **形式1：head-to-head**

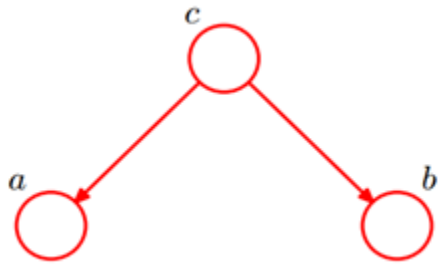


$$\sum_{c} P(a,b,c) = \sum_{c} P(a) * P(b) * P(c \mid a,b)$$

$$\Rightarrow P(a,b) = P(a) * P(b)$$

即在**c未知**的条件下，**a、b被阻断(blocked)，是独立的**

- **形式2：tail-to-tail**

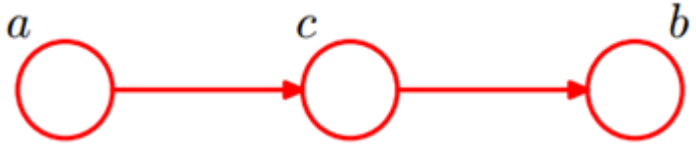


考虑c未知，跟c已知这两种情况：
1、在c未知的时候，有：P(a,b,c)=P(c)*P(a|c)*P(b|c)，此时，没法得出P(a,b) = P(a)P(b)，即c未知时，a、b不独立。

2、在c已知的时候，有：P(a,b|c)=P(a,b,c)/P(c)，然后将P(a,b,c)=P(c)*P(a|c)*P(b|c)带入式子中，得到：P(a,b|c)=P(a,b,c)/P(c) = P(c)*P(a|c)*P(b|c) / P(c) = P(a|c)*P(b|c)，即c已知时，a、b独立。

在c给定的条件下，a，b被阻断(blocked)，是独立的，

- **形式3：head-to-tail**



1、c未知时，有：P(a,b,c)=P(a)*P(c|a)*P(b|c)，但无法推出P(a,b) = P(a)P(b)，即c未知时，a、b不独立。
2、c已知时，有：P(a,b|c)=P(a,b,c)/P(c)，且根据P(a,c) = P(a)*P(c|a) = P(c)*P(a|c)，可化简得到：

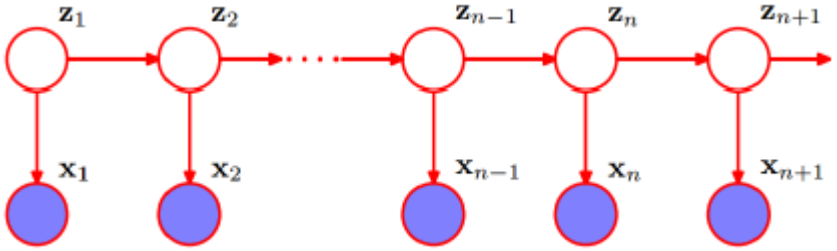
$$
\begin{aligned}
& P(a, b \mid c) \\
= \ & P(a, b, c) / P(c) \\
= \ & P(a) * P(c \mid a) * P(b \mid c) \ / \ P(c) \\
= \ & P(a, c) * \ P(b \mid c) \ / \ P(c) \\
= \ & P(a \mid c) \ * \ P(b \mid c)
\end{aligned}
$$

在c给定的条件下，a，b被阻断(blocked)，是独立的，

Markov chain

- 当前状态只跟上一状态有关，跟上上或上上之前的状态无关。这种顺次演变的随机过程，就叫做马尔科夫链（Markov chain）。

HMM



Bayes优缺点

优点

1.对待预测样本进行预测，过程简单速度快(想想邮件分类的问题，预测就是分词后进行概率乘积，在log域直接做加法更快)。

2.对于多分类问题也同样很有效，复杂度也不会有大程度上升。

3.在分布独立这个假设成立的情况下，贝叶斯分类器效果奇好，会略胜于逻辑回归，同时我们需要的样本量也更少一点。

4.对于类别类的输入特征变量，效果非常好。对于数值型变量特征，我们是默认它符合正态分布的。

缺点

- 1.对于测试集中的一个类别变量特征，如果在训练集里没见过，直接算的话概率就是0了，预测功能就失效了。当然，我们前面的文章提过我们有一种技术叫做『平滑』操作，可以缓解这个问题，最常见的平滑技术是拉普拉斯估测。
- 2. 朴素贝叶斯算出的概率结果，比较大小还凑合，实际物理含义...恩，别太当真。
- 3.朴素贝叶斯有分布独立的假设前提，而现实生活中这些predictor很难是完全独立的。

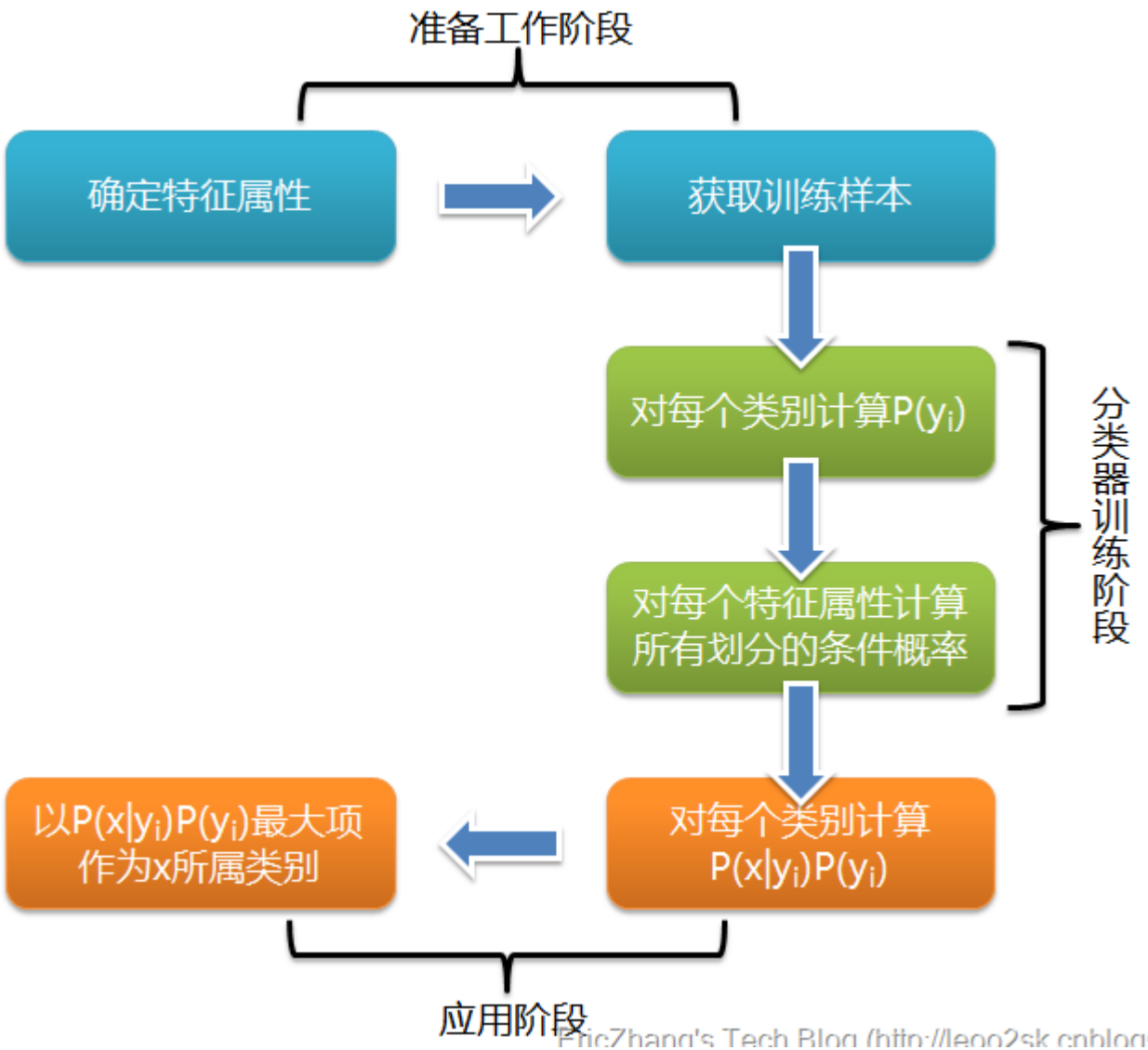
Bayes in papers

- 《基于贝叶斯网络的Android恶意行为检测方法》张国印
- 首先根据关联规则挖掘特征之间的依赖关系，然后根据依赖关系构建贝叶斯分类器的网络结构。在改进的关联规则算法中，我们获得了特征之间存在依赖关系的边的集合，通过最好局部优先搜索的策略，迚行贝叶斯网络结构的学习。（根据MWST）

- 《基于粒子群的加权朴素贝叶斯入侵检测模型》任晓奎
- 模型首先用粗糙集理论对样本属性特征集进行约简，再利用改进的粒子群算法优化加权朴素贝叶斯算法的属性权值，获得属性权值的最优解，用获得的最优解构造贝叶斯分类器完成检测。

- 粒子群优化加权朴素贝叶斯算法就是利用粒子群优化算法来寻找属性样本的最优权值。

分析应用的流程


准备工作阶段
确定特征属性
获取训练样本
对每个类别计算P(yi)
对每个特征属性计算
所有划分的条件概率
分类器训练阶段
对每个类别计算
P(x|yi)P(yi)
以P(x|yi)P(yi)最大项
作为x所属类别
应用阶段

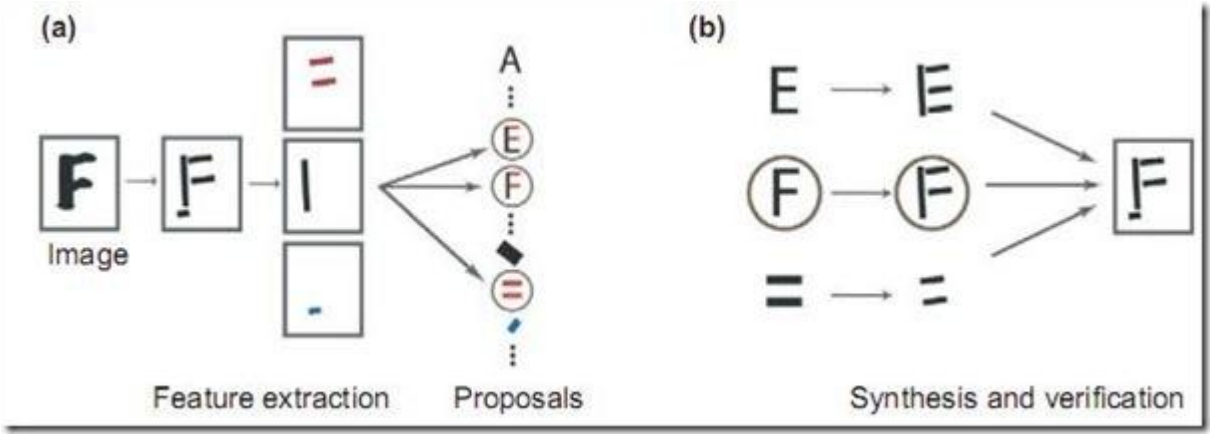
应用

- （一）拼写检查　通过计算编辑距离

P(我们猜测他想输入的单词 | 他实际输入的单词)

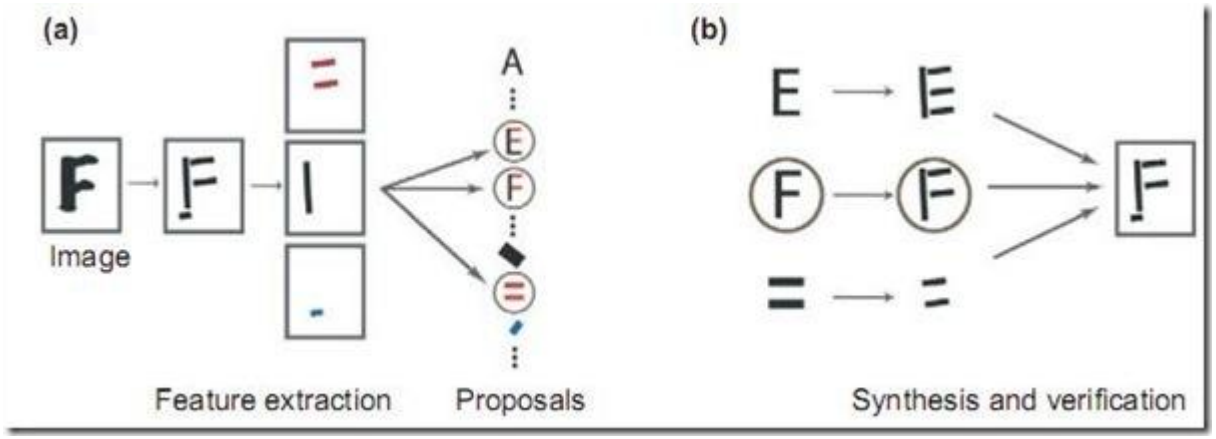
应用

- **（二）贝叶斯图像识别，Analysis by Synthesis**
- 贝叶斯方法是一个非常 general 的推理框架
- 核心理念：Analysis by Synthesis （通过合成来分析）。
- 06 年的认知科学新进展上有一篇 paper 就是讲用贝叶斯推理来解释视觉识别的，



应用

- 首先是视觉系统提取图形的边角特征，然后使用这些特征自底向上地激活高层的抽象概念（比如是 E 还是 F 还是等号），然后使用一个自顶向下的验证来比较到底哪个概念最佳地解释了观察到的图像。



应用

- （三）中文分词
- 南京市长江大桥

- 南京市　长江大桥

- 南京市　长江　大桥

应用

- （四）统计机器翻译

- 统计机器翻译因为其简单，自动（无需手动添加规则），迅速成为了机器翻译的事实标准。而统计机器翻译的核心算法也是使用的贝叶斯方法。

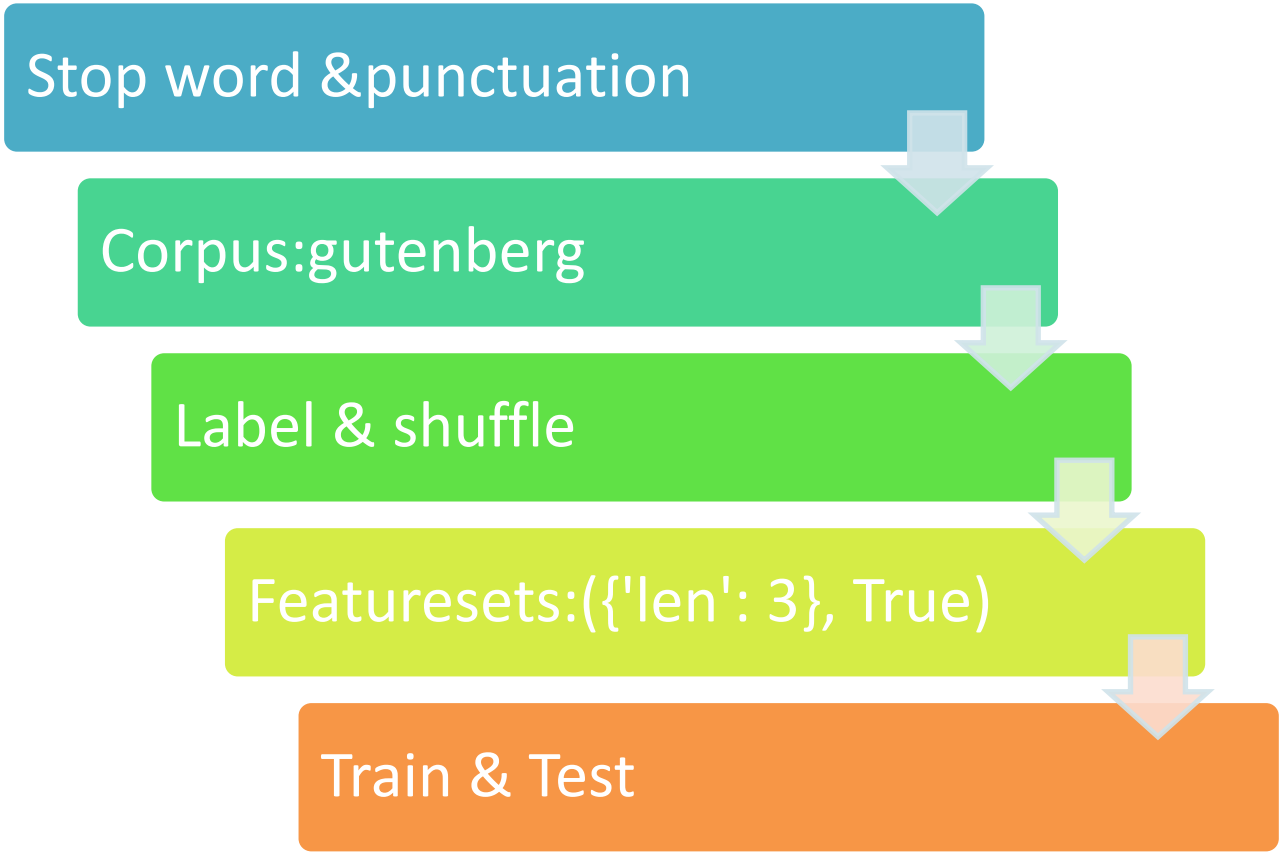
- 假设 e 为：John loves Mary 。我们需要考察的首选 f 是：Jean aime Marie （法文）。我们需要求出 P(e|f) 是多大，为此我们考虑 e 和 f 有多少种对齐的可能性，
- 如：John (Jean) loves (aime) Marie (Mary)

- 就是其中的一种（最靠谱的）对齐，为什么要对齐，是因为一旦对齐了之后，就可以容易地计算在这个对齐之下的 P(e|f) 是多大，只需计算：P(John|Jean) * P(loves|aime) * P(Marie|Mary)即可。

Experiment

E1

- 1、以单词长度为特征的停用词判定



E1：Data&Function

```
sw = set(nltk.corpus.stopwords.words('english'))
punctuation = set(string.punctuation)

def word_features(word):
    """以单词的长度作为特征"""
    return {'len':len(word)}

def isStopword(word):
    "判定是否为停用词"
    return word in sw or word in punctuation
```

E1:Process

```
"Get Data:an artical"
gb = nltk.corpus.gutenberg
words = gb.words("shakespeare-caesar.txt")

"(u'greeke', False)"
labeled_words = ([(word.lower(), isStopword(word.lower())) for word in words])

random.seed(42)
random.shuffle(labeled_words)
print labeled_words[:5]

featuresets = [(word_features(n), word) for (n, word) in labeled_words]
"[({'len': 3}, True), ({'len': 6}, False), ({'len': 5}, False), ({'len': 3}, True),

cutoff = int(.9 * len(featuresets))
train_set, test_set = featuresets[:cutoff], featuresets[cutoff:]
classifier = nltk.NaiveBayesClassifier.train(train_set)
print "'behold' class",classifier.classify(word_features('behold'))
print "'the' class", classifier.classify(word_features('the'))

print "Accuracy", nltk.classify.accuracy(classifier, test_set)
print classifier.show_most_informative_features(5)
```

- E1:Result

```
[(u'was', True), (u'greeke', False), (u'cause', False), (u'but', True), (u'house',
'behold' class False
'the' class True
Accuracy 0.846362229102
Most Informative Features
                    len = 7              False : True   =     62.7 : 1.0
                    len = 6              False : True   =     49.1 : 1.0
                    len = 1               True : False  =     12.0 : 1.0
                    len = 2               True : False  =     10.7 : 1.0
                    len = 5              False : True   =     10.4 : 1.0
```

E2:用scikit-learn做文本分类

- Dataset : sklearn.datasets　20newsgroups
- 特征:

  每个元素是特定单词在文本中出现的次数

  （本应用TF-IDF，但会造成训练与测试集的特征数不同，改用单纯词频）
- 维度：11314 documents, 130107 vectors for all categories

E2：code

- Naive Bayes
- from sklearn.naive_bayes import MultinomialNB
- from sklearn import metrics
- newsgroups_test = fetch_20newsgroups(subset = 'test',
- categories = categories);
- fea_test = vectorizer.fit_transform(newsgroups_test.data);
- #create the Multinomial Naive Bayesian Classifier
- clf = MultinomialNB(alpha = 0.01)
- clf.fit(fea_train,newsgroup_train.target);
- pred = clf.predict(fea_test);
- calculate_result(newsgroups_test.target,pred);

E2:Result

	precision	recall	f1-score
Naive Bayes	0.771	0.770	0.769
KNN	0.652	0.645	0.645
SVM	0.819	0.816	0.816
KMeans	0.289	0.313	0.266

模型稳定性高低

$$F1 = \frac{2 ? Precision ? Recall}{Precision + Recall}$$

E3：

- *Kaggle旧金山犯罪类型分类问题*

朴素贝叶斯建模耗时 1.350199 秒
朴素贝叶斯log损失为 2.582355
逻辑回归建模耗时 60.785606 秒
逻辑回归log损失为 2.591964

Think & Idea

- Classifier HMM MCMC
- An evaluation

  -> A natural sentence

  & A reasonable answer

Reference

- 我们来聊一聊机器学习的核心:参数估计以及贝叶斯模型
- 从贝叶斯方法谈到贝叶斯网络


thank
you!