



中科院计算培训中心

Part 5 大数据分类技术及其应用

训练与测试

- 杨文川



- 1) 分类的定义
- 2) 分类主要算法
- 3) Mahout分类过程
- 4) 评估指标以及评测
- 5) 贝叶斯算法新闻分类实例
- 6) MLlib分类过程及其应用



分类的定义

-以文本分类为例

- 定义：给定分类体系，将数据(文本)分到某个或几个类别中。
- 分类模式：
 - 两类问题(binary) 一篇文本属于或不属于某一类；
 - 多类问题(multi-class)，一篇文本属于多个类别中的其中一个类别，多类问题可拆分成两类问题；
 - 一个文本可以属于多类(multi-label)。
- 很多分类体系：
 - Reuters分类体系、中图分类

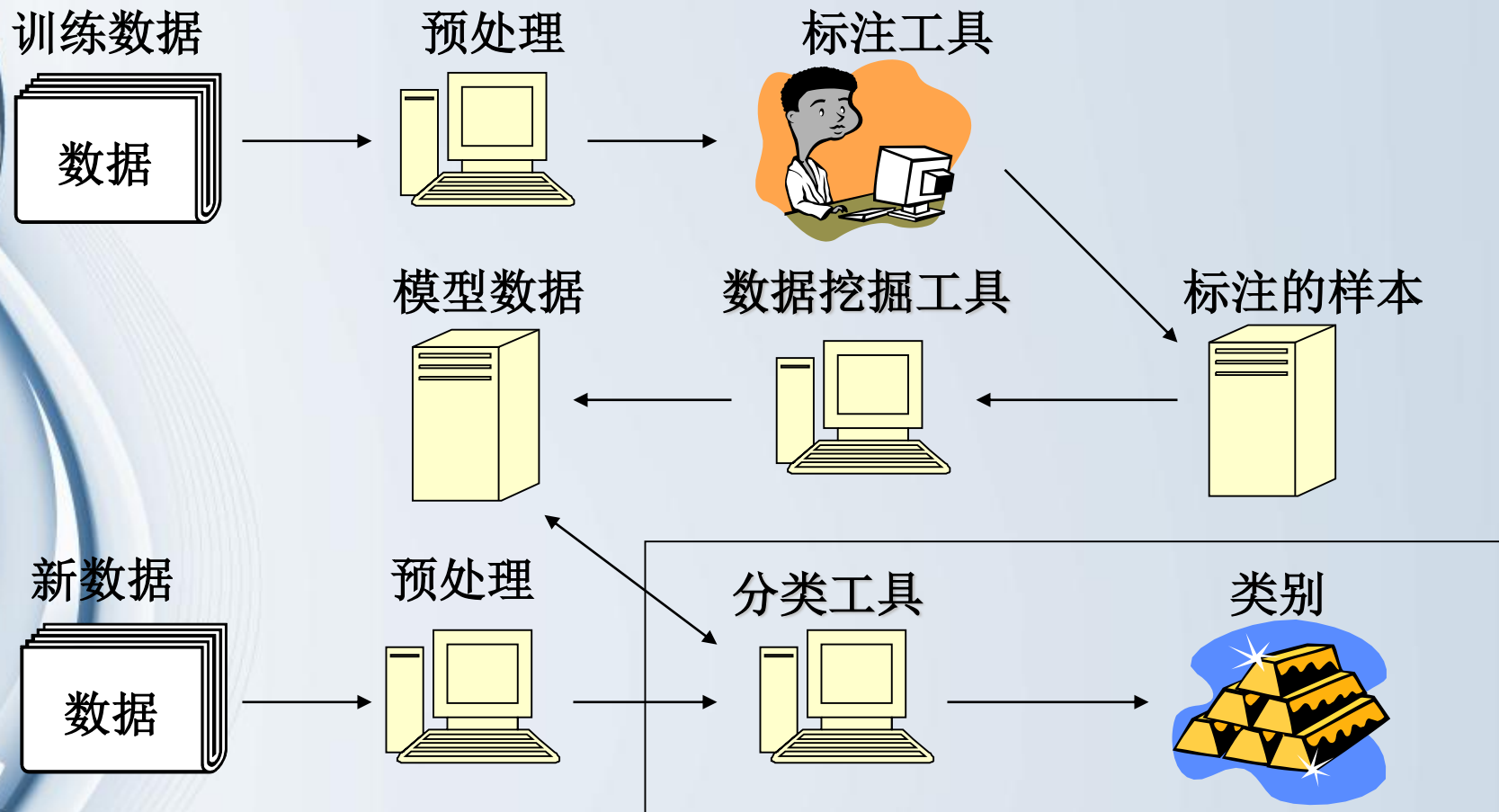


应用

- 垃圾邮件的判定(spam or not spam)
 - 类别 {spam, not-spam}
- 新闻出版按照栏目分类
 - 类别 {政治,体育,军事,...}
- 词性标注
 - 类别 {名词,动词,形容词,...}
- 词义排歧
 - 类别 {词义1,词义2,...}

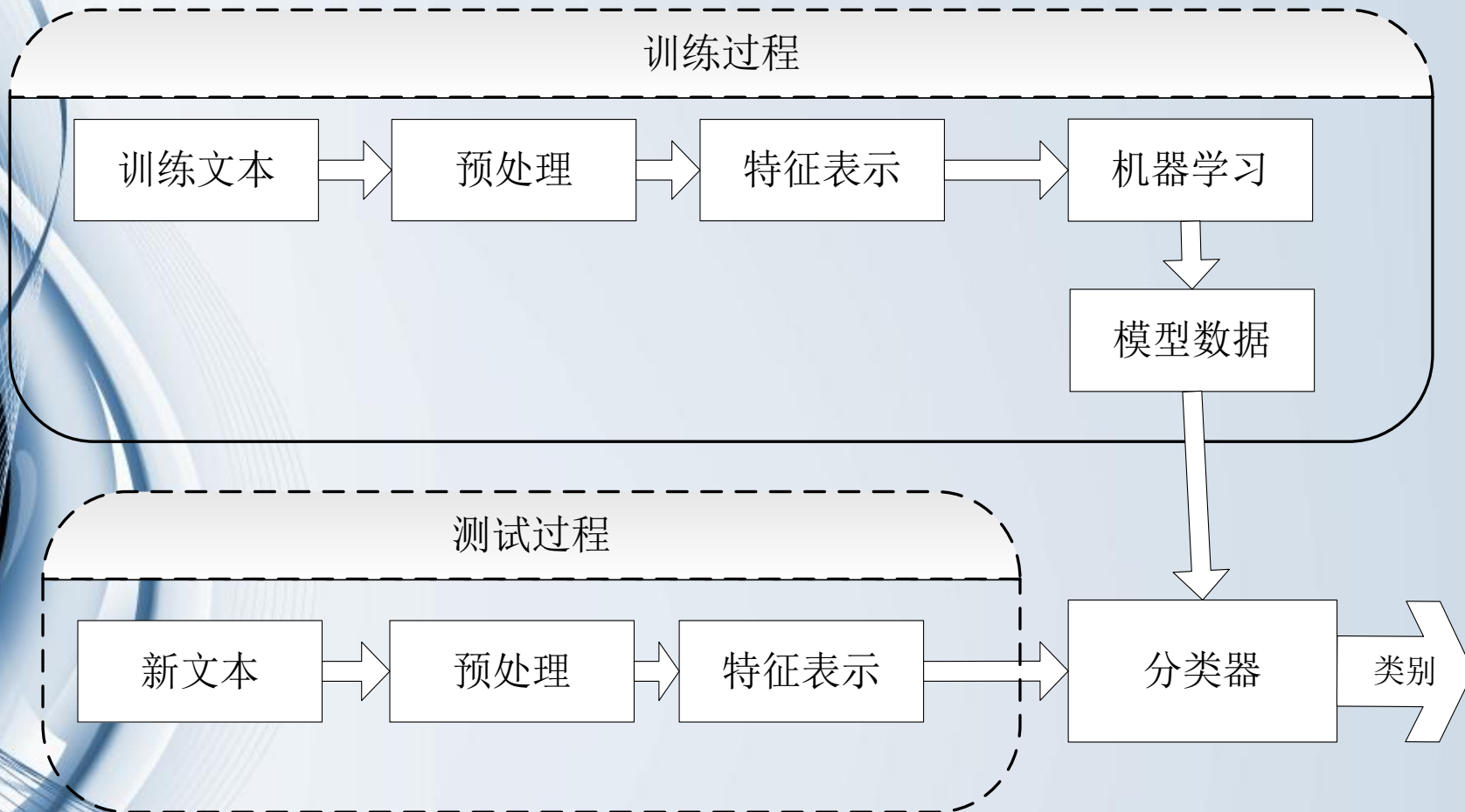


分类过程





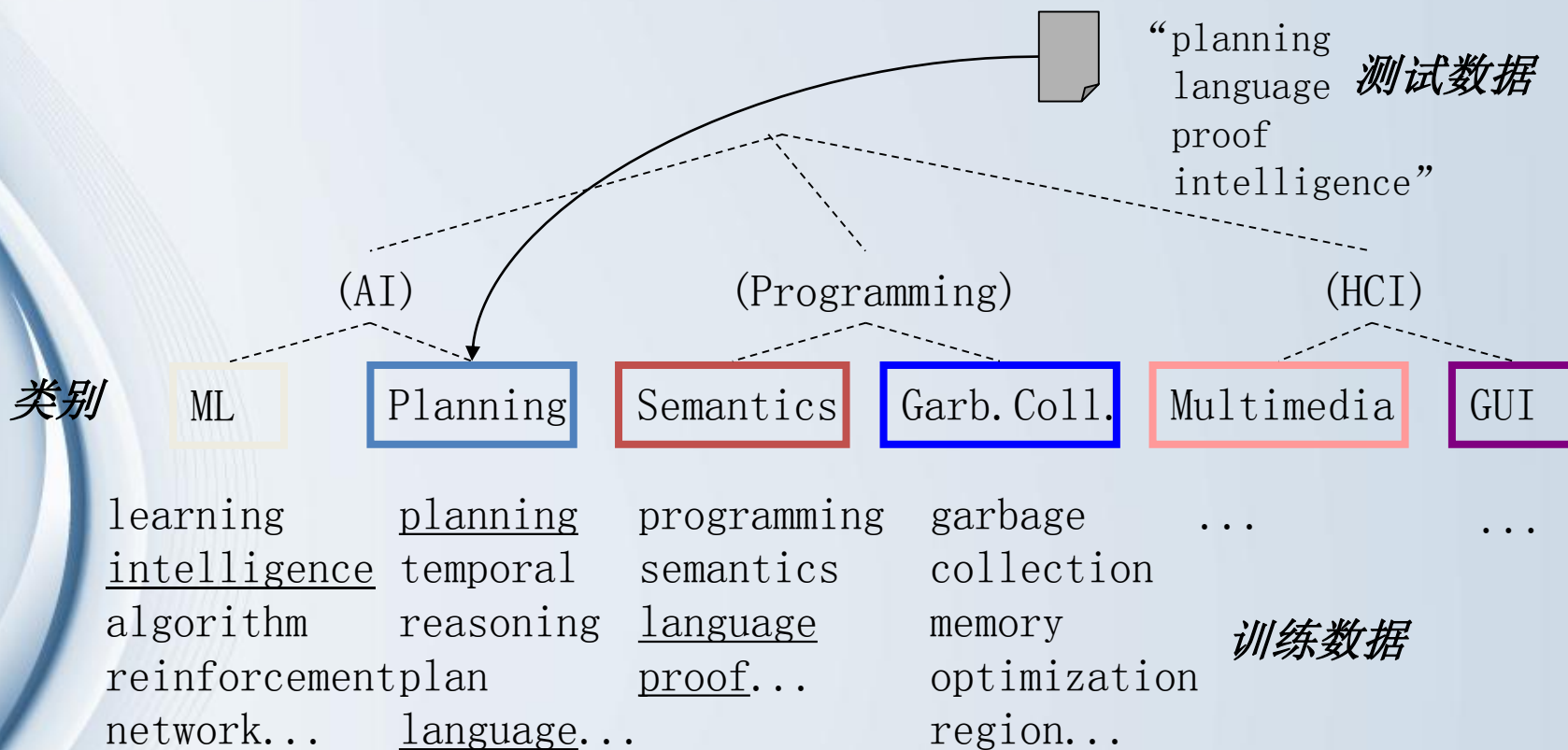
分类过程





分类示例

--文本分类





特征抽取

- 预处理
 - 去掉html一些tag标记
 - 对训练集合全部文档进行分词，并去除停用词和所有的单字词；
 - 统计每一类内出现词语的总词频，并取其中的前100个频率最高的词语作为这一类别的特征词集；
 - 去除在每一类别中都出现的词，然后合并所有类别的特征词集，形成总特征词集。
- 降维技术
 - 特征选择...



数据的表示

- 特征权重计算,是文本向量化表示的一个关键环节
 - TF.IDF
 -
- 权重计算大多是基于特征与单文档之间的关系,孤立了特征与类的联系,因此可能出现大量的兼类现象



典型分类项目的工作流

- 构建并运行分类系统，都需要遵循一系列非常标准的步骤。
- 分类项目的典型工作流是：
 - 训练模型，评估并调整模型以达到可接受水平，然后在生产环境中运行系统。
 - 用合成数据阐述工作流程中的一些步骤。
 - 最后，用Mahout完成分类项目。



目标变量是否能够当做特征

- 作为任何项目的先决条件，需要确定要寻找的信息(目标变量)是否能够当做特征
 - 以适当形式存储在一条记录中，并且符合总体目标的要求，比如识别欺诈性的金融交易。
 - 有时候，需要根据一些现实因素调整目标变量的选择，例如训练代价、隐私问题等。
 - 也需要知道有哪些数据可用于训练，系统运行时又会遇到什么新数据，这样才能设计出有用的分类器。



分类项目的一般开发步骤

- 如表所示

阶段	步骤
(1)训练模型	定义目标变量 搜集历史数据 定义预测变量 选择学习算法 使用学习算法训练模型
(2)评估模型	在测试数据上运行 调整输入(改变预测变量、改变算法，或二者一起改变)
(3)在生产中使用模型	输入新样本，估计未知的目标变量值 必要时重新训练模型



不存在单一的正确选择

- 系统每个阶段的准确性，和输出结果的有效性，反映了用作预测变量的原始特征选择，和目标变量的类别选择的好坏。
 - 对每种选择而言，并不存在单一的正确选择：有很多可取的方法和调整手段，要构建一个高效、健壮的系统，需要尝试一些不同的方法
 - 训练和评估模型的过程中，每一步的特征选择，都需要综合考虑经济和时间上，是否负担得起，以及模型估计值的精度是否可以接受。



第一阶段工作流：训练分类模型

- 在分类项目的第一阶段，需要有一个目标变量
 - 这有利于选择合适的历史数据，用于训练，以及选择有效的学习算法。这些决策之间都是密切相关的。
 - 在下面的讨论中，将考察特征选择方法，对Mahout学习算法的具体影响方式，确定哪个Mahout算法适合当前的分类器



定义目标变量的类别

- 定义目标变量涉及目标变量类别的定义。
 - 目标变量不能在一个开放集合中取值。选择的类别，反过来也会影响对学习算法的选择，因为某些算法仅适用于二值目标变量的情况。
 - 目标变量的类别数越少越好，如果可以把类别数降到两个的话，那么会有更多的学习算法可供选择。
 - 如果目标变量不适合降低到简单形式，可能需要构建多个分类系统，各自处理预期目标变量的某个方面。
 - 在最终的系统中，可以把这些分类系统的输出组合起来，以提供所需的决策或预测。



搜集历史数据

- 所选择的历史数据源，部分依赖于搜集带标注历史数据的具体需求。
 - 在某些问题中，确定目标变量的值是很困难的。
 - 要确保历史数据中，目标变量值的准确性。
 - 在一个有缺陷的目标变量，或数据搜集过程基础上建立的模型，效果很差



定义预测变量

- 选择了一个有效的目标变量，并定义好其可选值集合之后，需要定义预测变量。
 - 这些变量是从训练和测试样本中，提取的特征的具体编码。
 - 需要再一次检测样本源，确保它们的特征有效，且它们的值能以适当的格式存放在记录中。



目标泄漏

- 定义预测变量的一个重要考虑因素，是避免导致目标泄漏。
- 目标泄漏，是指在选择预测变量时，无意中引入了目标变量的信息。
 - 目标泄漏会严重影响分类系统的精度。当告诉分类器目标变量是一个预测变量时，这个问题会相当明显。
 - 这是一个显而易见的问题，但它经常发生。目标泄漏可能是相当隐晦的，很难找到。
- 建议：对结果太理想的模型持怀疑态度



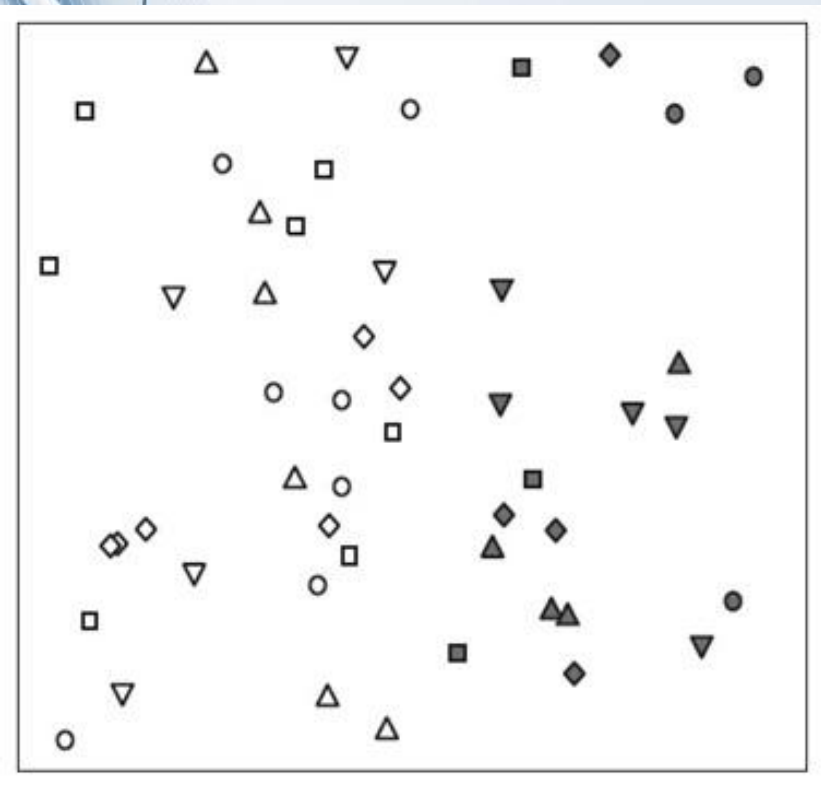
例子

- 假设需要构建一个垃圾邮件检测器，将垃圾邮件，和非垃圾邮件样本放入不同的文件，并按顺序给样本标上序号。
 - 同一文件中的样本，会有连续序号(在某个范围内)，而这一点可以用来区分垃圾和非垃圾邮件。
 - 如果存在这种类型的目标泄漏，很多分类算法都会迅速发现序号的范围，与垃圾邮件的对应关系，并仅依赖该特征解决问题，因为它(在训练数据中)看起来是完全可靠的。
- 这在实际工作中就会出问题



例子1：将位置用作预测变量

- 用一个使用合成数据的简单例子，演示如何选择预测变量，使Mahout的模型能够准确地预测期望的目标变量。
 - 图是一个历史数据集。假设在搜索颜色填充的形状：颜色填充是目标变量。
- 特征可以视为包含形状，和位置的预测变量。
 - 位置貌似适合用作预测变量：水平(x)坐标可能就足够了。
 - 形状似乎并不重要





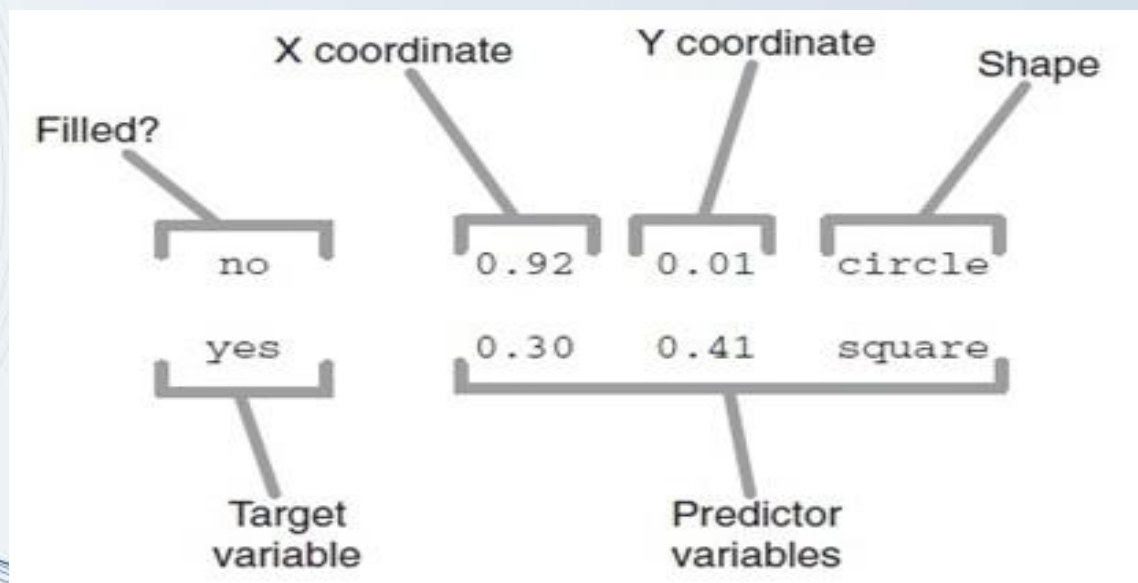
确定目标变量的类别

- 对于分类问题，必须确定目标变量的类别，在这个例子里就是颜色填充。
- 很明显，颜色填充有两种可能取值，即填充或未填充。
 - 现在需要选择用作预测变量的特征。哪些特征是可以正确表述的呢？
 - 首先排除颜色填充(它是目标变量)，可以将位置或形状用作变量。
 - 可以用x和y坐标来描述位置。基于一个数据表，可以为每个样本创建一条记录，包含目标变量和正在考虑的预测变量的字段。



两个训练样本的记录示例

- 图中数据的记录，包含存储目标变量值的字段，以及存储预测变量值的字段。
 - 在这种情况下，与位置(x、y坐标)和形状相关的值包含在两个样本中



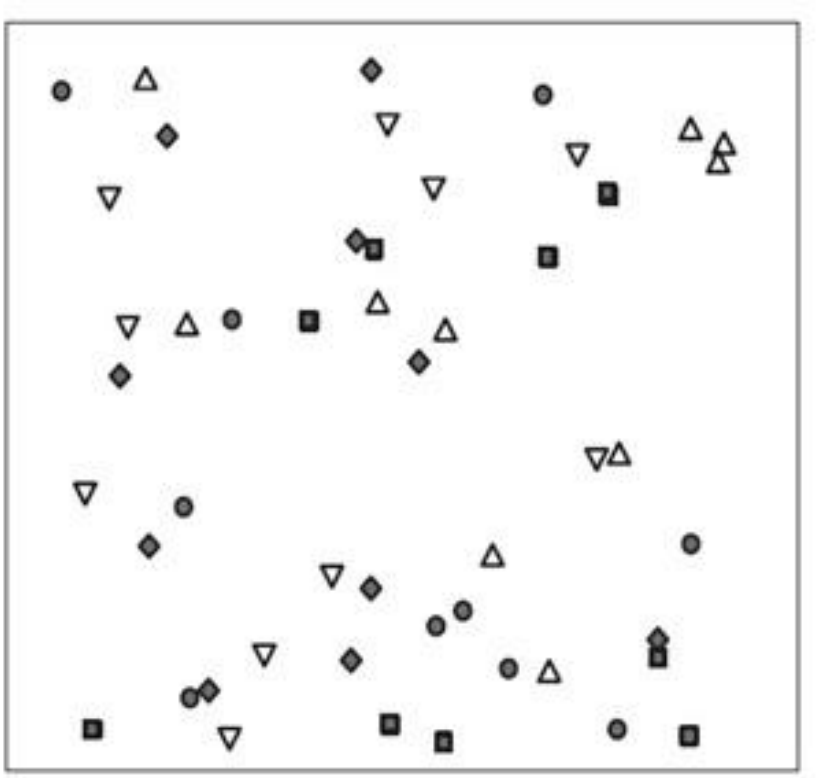


减少冗余

- 在设计分类系统时，需要根据经验，选择最可能有效的特征，模型的准确性也会反映出选择是否正确。
 - 引入的冗余或无关特征越少，分类器提供准确结果的可能性就越大。
 - 例如前例的数据，可能最好忽略y坐标和形状，仅使用x坐标训练模型。



例子2：不同数据需要不同预测变量



- 看看另一组历史数据，它们与之前的数据有着同样的特征。
 - 但在这种情况下，无论x还是y坐标，对预测符号是否填充颜色，没有作用。
 - 位置不再有用，但现在形状成为了一个有用的特征。
- 选作预测变量的特征(形状)具有3种值(圆形、三角形、正方形)。
 - 可引入朝向，来区分这些形状(朝上的三角形和朝下的三角形)



选择一个学习算法来训练模型

- 在选择所要使用的算法时，要考虑一些参数
 - 例如训练数据的规模
 - 预测变量的特性
 - 目标变量的类别数等。



Mahout实现的分类算法

- Mahout实现的分类算法有：
 - 随机梯度下降 (SGD)
 - 贝叶斯分类(Bayes)
 - 在线学习算法(Online Passive Aggressive)
 - 隐马尔科夫模型 (HMM)
 - 决策森林(随机森林,DF)
 -

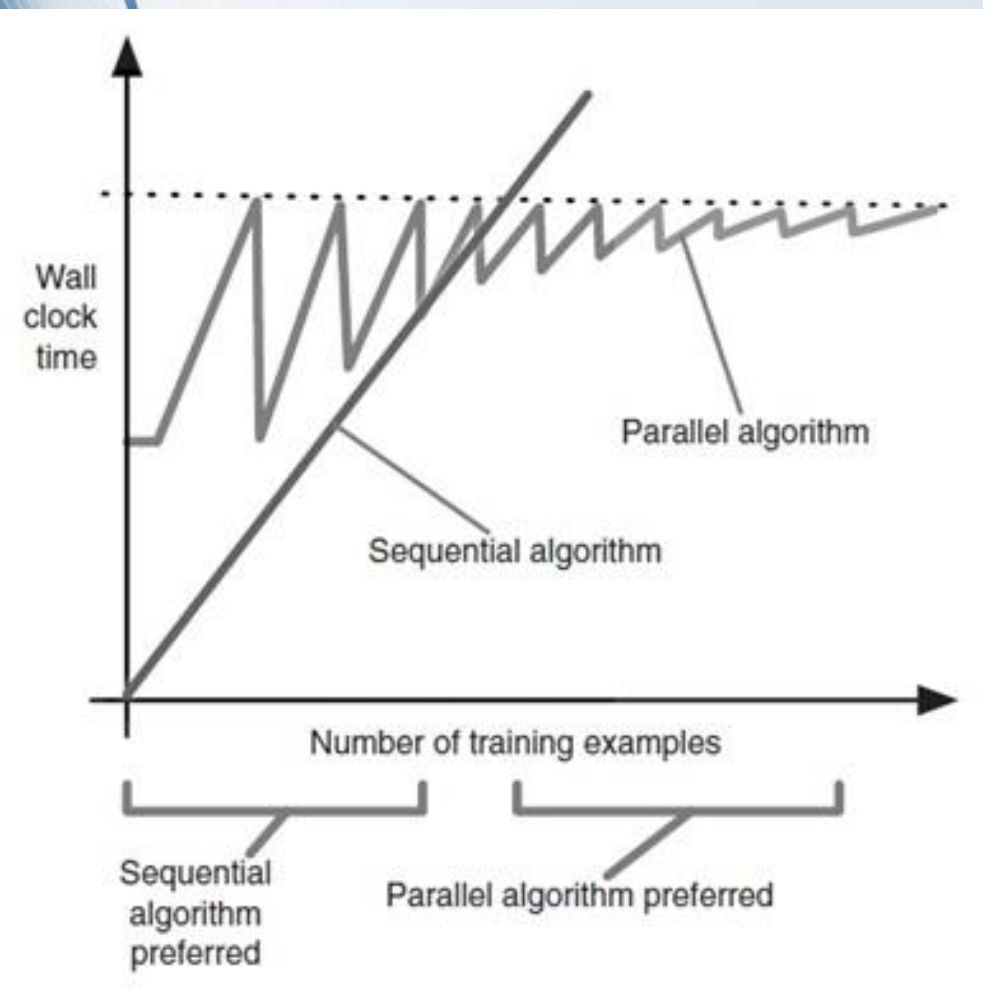


不同的算法各有优势

- 不同的算法也各有优势，以前面的例子为证
 - 在例1中，训练算法应该使用x坐标位置来判定颜色填充。在例2中，形状更有用。
- 注意到，一个点的x坐标点位置是连续变量，需要算法可以使用连续变量。
 - 在Mahout中，SGD和随机森林法，可以使用连续变量。
 - 朴素贝叶斯和补充朴素贝叶斯算法，则无法使用连续变量。



并行串行算法的权衡



并行算法有相当大的额外开销，开始处理样本之前，需要花一些时间去设置计算环境。

对于某些中等规模的数据集，串行算法可能不仅仅是够用，而往往是首选的。

这种权衡如图所示，其中比较了假设的串行，和并行可扩展算法的运行时间

锯齿形状的下落部分，是由于添加了新机器



使用学习算法训练模型

- 确定好合适的目标变量和预测变量集，并选好学习算法之后，训练的下一步，就是运行训练算法来生成模型。
- 这个模型会捕获学习算法得到的预测变量，与目标变量之间关系的本质。
- 对于例1，模型可能是类似下面的伪代码：
 - if ($x > 0.5$) return FILLED;
 - else return UNFILLED;



第二阶段 workflow: 评估分类模型

- 在生产中使用分类系统之前，有一个必要步骤，即确定它到底能有多好的表现。
 - 要做到这一点，必须评估该模型的准确性，并在真正开始分类之前做出或大或小的调整。
- 评估训练好的模型通常并不简单。在准备将系统投入生产之前，要对模型进行评估和调优。
- 后面有一个循序渐进的donut例子，将初步了解如何评估。



第三阶段 workflows: 生产中使用模型

- 一旦模型输出的准确性达到可接受的范围，就可以对新数据分类了。
 - 输入数据的质量，是决定生产中分类系统性能的最重要因素。
- 对模型进行周期性的复检是很有用的，有必要重新训练模型。
 - 如果需要分析的新数据预测变量的值不准确
 - 或者新数据与训练数据不一致
 - 或外部条件随着时间发生了改变
 - 分类模型输出的质量都会下降。



进行复检

- 要进行复检，需要搜集新的样本，并验证或生成目标变量值。
- 然后可以像第二阶段一样，将这些新样本作为测试数据，比较这些结果和原始数据中用于测试的数据集的结果。
 - 如果新的结果明显恶化，这就意味着某些因素发生了变化，需要考虑重新训练模型。
 - 需要重新训练模型的一个常见原因：将模型集成到的系统时，会改变系统。



银行欺诈检测系统的例子

- 如果一家大银行部署了一个非常有效的欺诈检测系统，会怎样？
 - 在数月或数年之内，骗子会慢慢适应，并开始采用原始模型无法检测的新技术。
 - 在这些新的欺诈方法造成损失之前，银行的建模人员监测到精度下降，并更新模型是很重要的



使用新的训练数据重新训练模型

- 一个分类系统的性能有所降低时，并不一定就要放弃这种方法。
 - 使用新的、更合适的训练数据，重新训练模型也许就足够了。
 - 另外，对训练算法做出某些调整可能会有用。
 - 实时评估也是重新训练的一部分，经常研究模型的更新



贝叶斯算法分类实例

20Newsgroups数据集

- Mahout提供的20Newsgroups数据集是收集大约2万个新闻组文档，均匀的分布在20个不同的集合。
 - 这20个新闻组集合，采集最近流行的数据，集合到文本程序中作为实验，采用不同的挖掘技术，例如文本分类，文本聚集。
 - 将使用Mahout的Bayes Classifier创建一个模型，它将一个新文档，分类到这20Newsgroups数据集



条件概率和贝叶斯公式

- 条件概率定义：
 - 设A、B是两个事件，且 $P(A) > 0$ ，称为在事件A发生的条件下，事件B发生的条件概率
 - $P(B|A) = P(AB)/P(A)$



设实验E为样本空间，A为E的事件， B_1, B_2, \dots, B_n 为 Ω 的一个分割，且 $P(B_i) > 0, i=1, 2, \dots, n$ ，则由：

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

有：

$$P(B_i | A) = \frac{P(B_i)P(A | B_i)}{\sum_{i=1}^n P(B_i)P(A | B_i)}$$

上式称为贝叶斯公式



例子

某电子设备厂所用的元件是由三家元件厂提供的，三厂家次品率为0.02, 0.01, 0.03, 提供元件厂的份额分别为0.15, 0.8, 0.05, 设这三个厂家的产品在仓库是均匀混合的，且无区别的标志。

问题1：在仓库中随机地取一个元件，求它是次品的概率。

问题2：在仓库中随机地取一个元件，若已知它是次品，为分析此次品出自何厂，需求出此元件由三个厂家分别生产的概率是多少？



【解】 设A取到的元件是次品， B_i 标识取到的元件是由第*i*个厂家生产的，则

$$P(B_1)=0.15, P(B_2)=0.8, P(B_3)=0.05$$

对于问题1，由全概率公式：

$$P(A) = \sum_{i=1}^3 P(B_i)P(A | B_i)$$

$$=0.15 \times 0.02 + 0.80 \times 0.01 + 0.05 \times 0.03 = 0.0125$$

对于问题2，由贝叶斯公式：

$$P(B_1 | A) = \frac{P(B_1)P(A | B_1)}{P(A)} = \frac{0.15 \times 0.02}{0.0125} = 0.24$$

$$P(B_2 | A) = \frac{P(B_2)P(A | B_2)}{P(A)} = \frac{0.80 \times 0.01}{0.0125} = 0.64$$

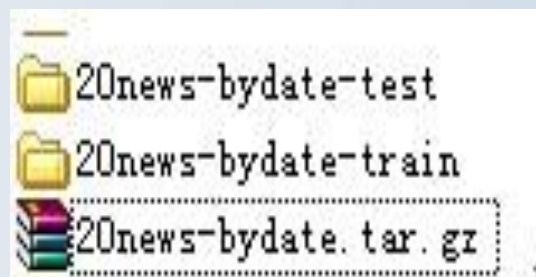
$$P(B_3 | A) = \frac{P(B_3)P(A | B_3)}{P(A)} = \frac{0.05 \times 0.03}{0.0125} = 0.12$$

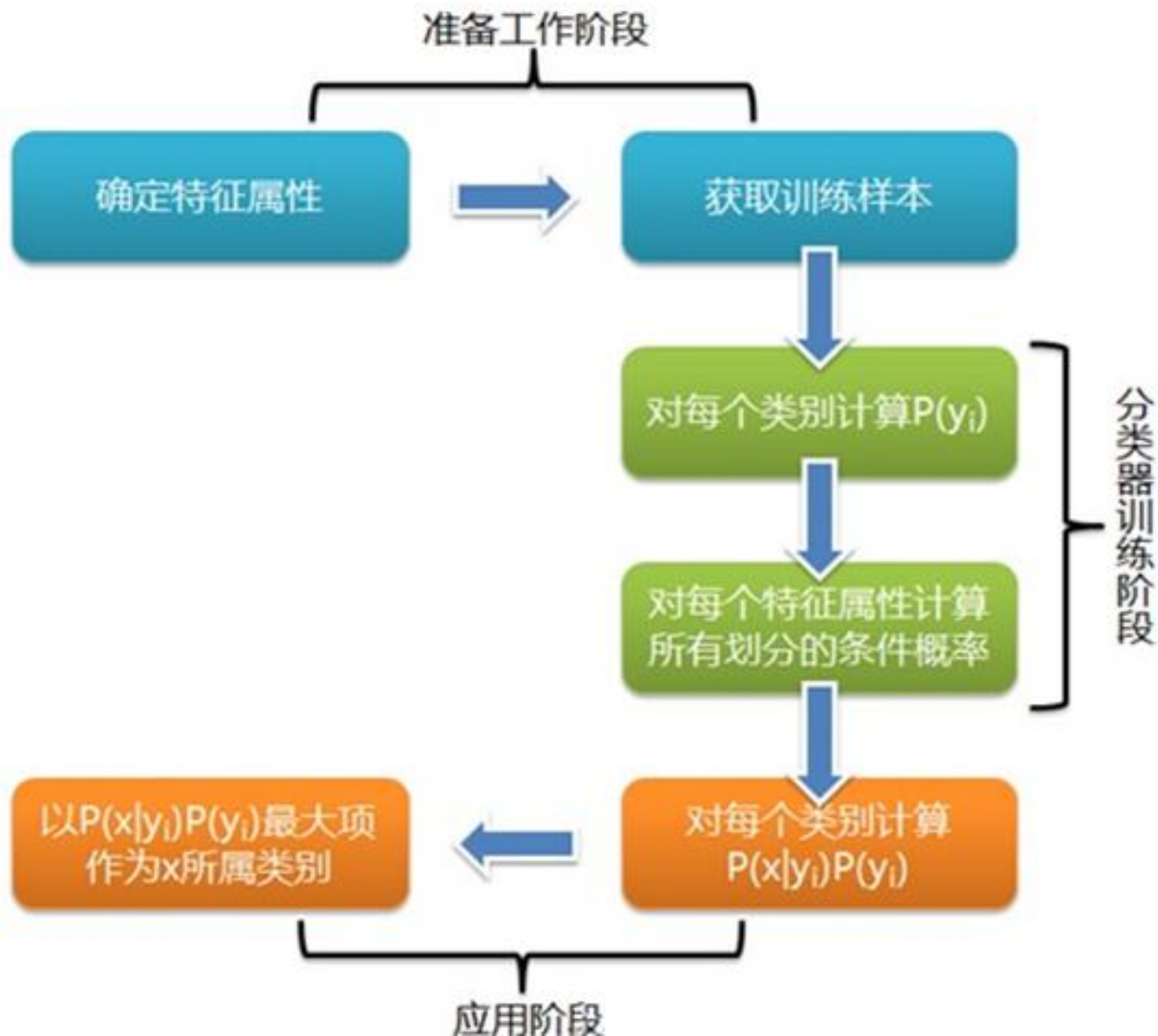
结果表明，这个次品来自第2家工厂的可能性最大，来自第1家工厂的概率次之，来自第3家工厂的概率最小。



数据的准备

- 下载20news-bydate.tar.gz数据包并解压缩
- <http://people.csail.mit.edu/jrennie/20Newsgroups/20news-bydate.tar.gz>
- 例如：把数据包放在/home/lab466/example下了，所以以下的命令都是在这个目录下的

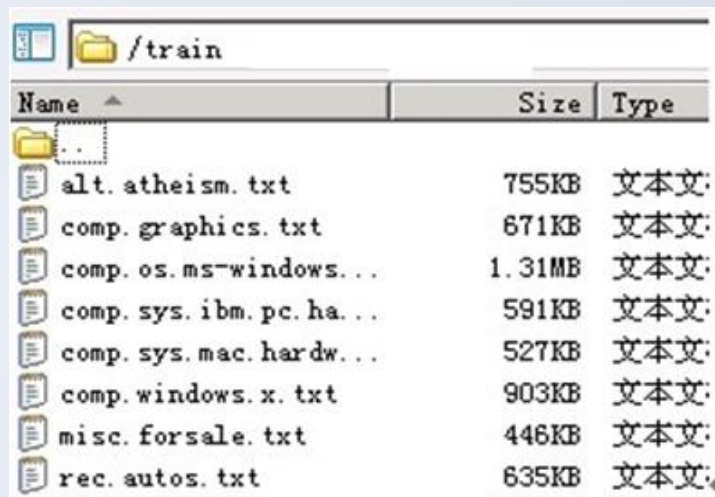






生成input的数据

- `bin/mahout Prepare20newsgroups -p 20news-bydate-train/ -o 20news-train/ -a org.apache.lucene.analysis.standard.StandardAnalyzer -c UTF-8`

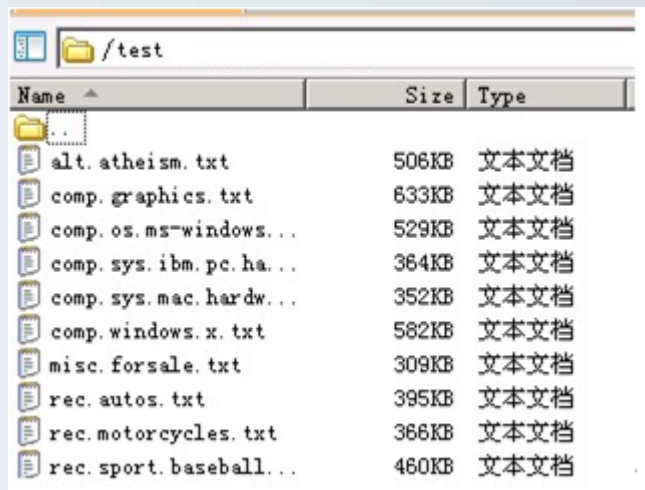


Name	Size	Type
alt.atheism.txt	755KB	文本文
comp.graphics.txt	671KB	文本文
comp.os.ms-windows...	1.31MB	文本文
comp.sys.ibm.pc.ha...	591KB	文本文
comp.sys.mac.hardw...	527KB	文本文
comp.windows.x.txt	903KB	文本文
misc.forsale.txt	446KB	文本文
rec.autos.txt	635KB	文本文



生成test的数据

- `bin/mahout Prepare20newsgroups -p 20news-bydate-test -o 20news-test -a org.apache.lucene.analysis.standard.StandardAnalyzer -c UTF-8`



Name	Size	Type
alt.atheism.txt	506KB	文本文档
comp.graphics.txt	633KB	文本文档
comp.os.ms-windows...	529KB	文本文档
comp.sys.ibm.pc.ha...	364KB	文本文档
comp.sys.mac.hardw...	352KB	文本文档
comp.windows.x.txt	582KB	文本文档
misc.forsale.txt	309KB	文本文档
rec.autos.txt	395KB	文本文档
rec.motorcycles.txt	366KB	文本文档
rec.sport.baseball...	460KB	文本文档



在Hadoop中执行命令与过程

- 上传文件到HDFS
 - `hadoop fs -put /home/lab466/20new-train`
 - `hadoop fs -put /home/lab466/20news-test`





执行命令与过程

- 下面将在hadoop运行map reduce工作，为了Train这个分器,系统将运行一段时间
 - `bin/mahout trainclassifier -i
hdfs://localhost:9000/user/lab466/20news-train -o
hdfs://localhost:9000/user/lab466/20news-model -type
bayes -ng 1 -source hdfs`



结果

Incorrectly Classified Instances : 9 0.1995%

Total Classified Instances : 4512

Confusion Matrix

[illegible]



运行Test分类器

- 由于案例数据较多，在单机测试环境下，将运行近10分钟，新的newmodel大概有300M
 - 可以通过<http://localhost:50030/jobtracker.jsp>来监控job的状态
- 在input目录运行Test分类器
 - `bin/mahout testclassifier -d
hdfs://localhost:9000/user/lab466/20news-test -m
hdfs://localhost:9000/user/lab466/20news-model -type
cbayes -ng 1 -source hdfs -method mapreduce`



输出结果

```
Correctly Classified Instances : 2672 88.4768%  
Incorrectly Classified Instances : 348 11.5232%  
Total Classified Instances : 3020
```

Confusion Matrix

```
a b c d e f g h i j k l m n o p q r s t <--Classified as  
127 0 0 0 0 0 0 1 0 1 0 0 0 0 1 2 1 0 6 0 | 139 a = alt.atheism  
0 136 5 5 4 5 1 0 1 0 0 1 3 0 0 0 0 0 0 0 | 161 b = comp.graphics  
0 9 121 18 3 3 1 1 0 0 0 0 0 0 0 0 0 0 0 0 | 156 c = comp.os.ms-windows.misc  
1 1 5 127 8 0 1 0 0 0 0 1 1 0 1 0 0 0 0 0 | 146 d = comp.sys.ibm.pc.hardware  
0 4 4 7 155 1 1 1 0 0 0 0 4 0 1 0 0 0 0 0 | 178 e = comp.sys.mac.hardware  
0 14 8 1 5 122 1 0 0 1 0 1 0 0 0 0 0 0 0 0 | 153 f = comp.windows.x  
0 4 3 12 3 0 119 5 0 1 0 0 4 0 0 0 0 0 0 0 | 151 g = misc.forsale  
0 0 1 1 0 1 3 134 3 0 0 0 1 0 0 0 0 0 0 1 | 145 h = rec.autos  
0 0 0 0 3 0 0 5 141 0 0 0 0 0 0 0 0 0 0 0 | 149 i = rec.motorcycles  
0 0 0 0 1 0 0 0 0 151 2 0 0 1 0 0 0 0 0 0 | 155 j = rec.sport.baseball  
0 0 0 1 0 1 0 1 0 5 144 0 0 0 0 1 0 0 0 0 | 153 k = rec.sport.hockey  
1 2 0 0 1 0 1 0 0 0 0 156 1 0 0 0 0 2 0 0 | 164 l = sci.crypt  
0 4 1 6 6 1 2 4 0 0 0 2 119 0 0 0 0 0 1 | 146 m = sci.electronics  
0 1 1 1 0 0 3 0 0 0 0 0 5 137 0 0 1 1 0 2 | 152 n = sci.med  
0 6 0 0 0 0 0 1 0 0 0 0 0 1 167 0 0 0 2 0 | 177 o = sci.space  
2 1 1 0 1 0 0 0 0 0 2 1 1 1 0 150 1 0 1 1 | 163 p = soc.religion.christian  
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 2 149 0 0 2 | 154 q = talk.politics.mideast  
0 0 0 1 0 0 0 0 0 0 0 2 0 0 0 0 0 140 1 7 | 151 r = talk.politics.guns  
13 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 3 0 3 70 5 | 96 s = talk.religion.misc  
2 1 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 17 1 107 | 131 t = talk.politics.misc
```




数据格式分析

- 就像案例一样：需要输入的内容是文本分类和模型使用
 - 在train中添加特征类型数据
 - 在test中添加测试数据
- Train的输入格式，第一个字符时类标签，剩余的是特征（单词），标签和特征词用table键隔开，如下：

```
alt.atheism howland.reston.ans.net europa.eng.gtefsd.com uunet mcsun germany.eu.net news.dfn.de  
correction have been more elaborate arguments made looks have passed your post filtering i thus inter  
strong motives some other things note he fulfilled loads prophecies psalms isaiah elsewhere 24 hrs  
alied intent we have already looked dictionary define word isn't sufficient we only need ask quest  
one would hope bob beauchaine bobbe vice.ico.tek.com said queens could stay blew bronx away sank m  
on believe wasn't ever man good however most claims extraordinary eg virgin birth virgin sense havin  
patrick c leger writes you know just occurred me today whole christian thing can blamed solely mar  
re.wpd.sgi.com jon livesey writes i hope islamic bank something other than bcci which ripped off si  
ages because performed convenience what happens typically woman willing move her lover without any  
honestly think someone who did would voluntarily come forward confess kind extremism shown your co  
ferent people within religion nothing existing different perspectives within religion perhaps one  
tory islamic history yes supposed have been predominant view turkish caliphate i would like refere
```

- 如在20newsgroup这个例子中的文件中 alt.atheism是类标签，剩下的是特征。其他的类似



结果提取并应用

- 看这条输出：

a	b	c	<--Classified as		
386	3	0	389	a	= comp.graphics
15	304	0	319	b	= alt.atheism

- a列代表数据在a中的符合条数（即可能有386个在a），所以a被归类在comp.graphics的可能性很高
- 可通过这样的数据，来判断数据可能被归类到哪里



主要的文本分类评估指标小结

- 评价方法
 - F 值
 - 查准率(precision) = $a / (a + b)$
 - 召回率(recall) = $a / (a + c)$
 - fallout = $b / (b + d)$
 - 损失函数
 - 其它
 - AUC, Confusion Matrix, Entropy...



MLlib二元分类模型

- MLlib目前支持两个适用于二元分类标准模型：
 - 线性支持向量机(SVM)和逻辑回归，同时也包括分别适用于这两个模型家族的L1和L2正则化变体
 - 训练算法都利用一个底层的，梯度下降基础算法
 - 二元分类算法的输入值，是一个正则项参数(regParam)和多个与梯度下降相关的参数(stepsize, numIterations, miniBatchFraction)。
- 目前可用的二元分类算法有：
 - SVMwithSGD
 - LogisticRegressionWithSGD
- 下面主要讲解SVM



线性支持向量机(SVM)

- 线性支持向量机是一种大规模分类问题的标准方法。线性表示如下

$$L(w; x, y) := \max\{0, 1 - yw^T x\}$$

- 默认使用L2正则化。L1为可选。这样，问题变成了一个线性程序。
- 线性SVM算法输出一个SVM模型。
 - 给定一个新数据点，用 x 表示，模型根据 $w^T x$ 的值做预测。
 - 默认如果 $w^T x \geq 0$ 输出为正值，否则为负值。



评价矩阵

- MLlib支持常用的二分类评价矩阵。
 - 包括准确率，召回率，F值，receiver operating characteristic(ROC)，准确率召回率曲线，和area under the curve(AUC)。
- AUC是一种常用的模型性能比较方法，用于帮助用户通过准确率/召回率/F值来选择预测阈值。



二元分类例子

- 下面是Scala中二元分类例子
ClassifySVM
 - 其中利用了SVMWithSGD
 - 注意观察其中的数据sample_libsvm_data.txt
- 代码段演示了，如何导入一份样本数据集，使用算法对象中的静态方法，在训练集上执行训练算法，在所得的模型上进行预测，并计算训练误差



中科院计算培训中心

MLlib朴素贝叶斯分类

- 朴素贝叶斯分类算法是一种监督学习算法
 - 它在大数据集的应用中，方法简便、准确率高和速度快的优点。
- 朴素贝叶斯分类算法主要有两种模型
 - 多项式模型(multinomial model)和伯努利模型(Bernoulli model)。
 - MLlib使用广泛应用的多项式模型。



Spark实现贝叶斯算法

- 参见例子SparkNBayesTst
- 训练数据
 - 0,1 0 0
 - 0,2 0 0
 - 0,1 0 0.1
 - 0,2 0 0.2
 - 0,1 0.1 0
 -
 - 其中第一列代表类别，训练数据有三类：0、1、2。
 - 第2-4列代表数据的三个维度，可以想象成前文中性别分类算法中的头发长度、服饰和体型这三个要素
- 为保证每个要素的权值相差不大，需要取相对的数值，例如头发长度/最长的头发长度。



示例

- 代码参见SparkNBayesTst中源码
- 结果
 - Accuracy=1.0
 - Prediction of (0.5, 3.0, 0.5):1.0
 - Prediction of (1.5, 0.4, 0.6):0.0
 - Prediction of (0.3, 0.4, 2.6):2.0
- 可看到此次训练的模型精度十分高为100%，即测试数据的类别和用模型预测出来的对于类别完全吻合
- 实际生产环境中是无法达到100%的。后面又预测了3个不在训练数据中的数据，结果相同。



中科院计算培训中心

谢 谢