# 朴素贝叶斯算法详解

贝叶斯分类算法是统计学的一种分类方法，它是一类利用概率统计知识进行分类的算法。贝叶斯算法可分为朴素贝叶斯算法,贝叶斯信念网络,树增强型朴素贝叶斯算法(TAN)等等.而其中运用最广的是朴素贝叶斯算法,朴素贝叶斯分类算法(Naive Bayes)可以与决策树和神经网络分类算法相媲美，该算法能运用到大型数据库中，而且方法简单、分类准确率高、速度快。

它的应用场景非常广,我们可以通过已知的训练数据来为未知的数据分类,训练数据越大,结果越接近真实值,我们可以用中医看病来分析这个过程,医生看到病人不能立马就得出结果,而是通过观察病人的各种情况然后推断出病因,这种过程其实是把病人的特征和自己记忆中病例的特征进行比对,和贝叶斯算法对训练数据比对有异曲同工之妙.

贝叶斯定理简介:

贝叶斯定理是关于随机事件A和B的条件概率的一则定理。已知某条件概率，如何得到两个事件交换后的概率，也就是在已知P(A|B)的情况下如何求得P(B|A)。在之前我们先了解下条件概率.

条件概率为: 表示事件B已经发生的前提下，事件A发生的概率，叫做事件B发生下事件A的条件概率。其基本求解公式为：

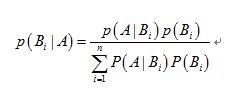
P(A|B)=P(AB)/P(B)

而贝叶斯定理公式为

P(B|A)=P(A|B)P(B)/P(A)

假设B是由相互独立的事件组成的概率空间{B1,B2，...Bn}。则P(A)可以用全概率公式展开：P(A)=P（A|B1)P(B1)+P（A|B2)P(B2)+..P（A|Bi)P(Bi)。

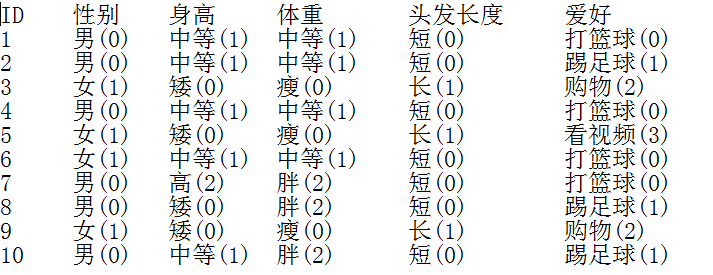
贝叶斯公式表示成：P(Bi|A)=P(A|Bi)P(Bi)/(P(A|B1)P(B1)+P(A|B2)P(B2)+..P(A|Bn)P(Bn))；常常把P(Bi|A)称作后验概率，而P（A|Bn)P(Bn)为先验概率。而P(Bi)又叫做基础概率。即：



看到这些概率论的知识是不是特别头痛,这里只是阐述了一下贝叶斯定理,而朴素贝叶斯算法的理论推导更加负责,这里就不做推论了,有兴趣的朋友可以去了解一下,接下来我将以一个实际的例子来详细描述下其中的过程.

场景描述为:通过每个人的特征情况来判断人的性别.

测试数据为:



其中数据用括号中的数字作为标签特征

现在假如ID为1的人的性别不知道,我们现在按照朴素贝叶斯的思想来计算他的性别:

我们首先假设1号为男性

那么他为男性的概率

P(男)=5/9 //用数据中男生人数/数据总人数

他为男性且身高中等的概率为

P(身高中等|男)=3/5 //用男生中身高中等的人数/男生总人数

他为男性且体重中等的概率为

P(体重中等|男)=2/5 //用男生中体重中等的人数/男生总人数

他为男性且头发短的概率为

P(头发短|男)=5/5 //用男生中短头发的人数/男生总人数

他为男性且爱好打篮球的概率为

P(打篮球|男)=2/5 //用男生中打篮球的人数/男生总人数

利用贝叶斯定理,现在我们可以算出他在身高中等,体重中等,头发短,打篮球的情况下为男生的概率为

a:P(男|身高中等,体重中等,头发短,打篮球)=P(男)\*P(身高中等|男)\*P(体重中等|男)\*P(头发短|男)\*P(打篮球|男)=0.0533333

同理,我们假设1号为女性,可以得出

P(女)=4/9

P(身高中等|女)=1/4

P(体重中等|女)=1/4

P(头发短|女)=1/4

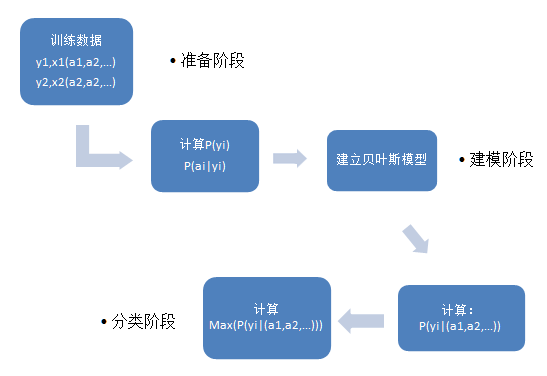
P(打篮球|女)=1/4

可以计算得出他在身高中等,体重中等,头发短,打篮球的情况下为女生的概率为

b:P(女|身高中等,体重中等,头发短,打篮球)=P(女)\*P(身高中等|女)\*P(体重中等|女)\*P(头发短|女)\*P(打篮球|女)=0.001736111

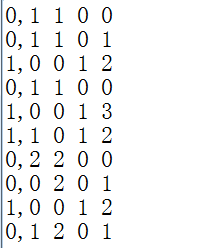
由a,b两式知:1号为男生的概率高于为女生的概率,由此我们判定1号为男生.

我们再看一下流程图:



在理解了朴素贝叶斯算法的计算过程后,我们用Spark mllib中的NaiveBayes算法来测试一下:

在之前我们首先将测试数据装换为数字模型，人物的性别我们用0,1表示，如下图：



核心源码如下:

//读入数据

val data = sc.textFile("/home/kexu/朴素贝叶斯算法测试数据(2)")

val parsedData =data.map { line =>

val parts =line.split(',')

LabeledPoint(parts(0).toInt,Vectors.dense(parts(1).split(' ').map(\_.toDouble)))

}

// 把数据的70%作为训练集，30%作为测试集.

val splits = parsedData.randomSplit(Array(0.7,0.3),seed = 11L)

val training =splits(0)

val test =splits(1)

//获得训练模型,第一个参数为训练数据，第二个参数为平滑参数，默认为1，可自定义

val model =NaiveBayes.train(training,lambda = 1.0)

//对模型进行准确度分析,得出计算精度

val predictionAndLabel= test.map(p => (model.predict(p.features),p.label))

val accuracy =1.0 \*predictionAndLabel.filter(x => x.\_1 == x.\_2).count() / test.count()

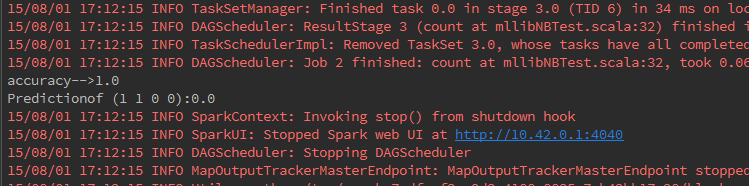
println("accuracy-->"+accuracy)

//输入(0,0,1,2)的人物特征数据来计算他的性别并输出

println("Predictionof (1 1 0 0 ):"+model.predict(Vectors.dense(1,1,0,0)))

我们调用了Spark内置算法NaiveBayes来进行朴素贝叶斯计算

最后我们根据1号的数据,来判断他的性别,现在看看输出结果:



可以看出结果为男生,和我们一开始的计算结果也是一致的,不过由于我们的数据量太少,而且数据是自己撰写的,真实度不高,导致精度为100%,不过在实际情况中一般是很难达到这么高的精度,一般我们都是增加训练数据来提高精度,或者采用多种算法协同计算的方法.

好了朴素贝叶斯算法的Spark实现就讲完了.测试源码即数据均在压缩包中。