

法直接看到病人的病情，只能观察病人表现出的症状和各种化验检测数据来推断病情，这时医生就好比一个分类器，而这个医生诊断的准确率，与他当初受到的教育方式（构造方法）、病人的症状是否突出（待分类数据的特性）以及医生的经验多少（训练样本数量）都有密切关系。

1.3、贝叶斯分类的基础——贝叶斯定理

每次提到贝叶斯定理，我心中的崇敬之情都油然而生，倒不是因为这个定理多高深，而是因为它特别有用。这个定理解决了现实生活里经常遇到的问题：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)。这里先解释什么是条件概率：

$P(A|B)$ 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：

贝叶斯定理之所以有用，是因为我们在生活中经常遇到这种情况：我们可以很容易直接得出P(A|B)，P(B|A)则很难直接得出，但我们更关心P(B|A)，贝叶斯定理就为我们打通从P(A|B)获得P(B|A)的道路。

下面不加证明地直接给出贝叶斯定理：

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

1.4、朴素贝叶斯分类

1.4.1、朴素贝叶斯分类的原理与流程

朴素贝叶斯分类是一种十分简单的分类算法，叫它朴素贝叶斯分类是因为这种方法的思想真的很朴素，朴素贝叶斯的思想基础是这样的：对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，哪个最大，则认为此待分类项属于哪个类别。通俗来说，就好比这么个道理，你在街上看到一个黑人，我问你你猜这哥们哪里来的，你十有八九猜非洲。为什么呢？因为黑人中非洲人的比率最高，当然人家也可能是美洲人或亚洲人，但在没有其它可用信息下，我们会选择条件概率最大的类别，这就是朴素贝叶斯的思想基础。

朴素贝叶斯分类的正式定义如下：

- 1、设 $x = \{a_1, a_2, ..., a_m\}$ 为一个待分类项，而每个 a 为 x 的一个特征属性。
- 2、有类别集合 $C = \{y_1, y_2, ..., y_n\}$ 。
- 3、计算 $P(y_1|x), P(y_2|x), ..., P(y_n|x)$ 。
- 4、如果 $P(y_k|x) = \max\{P(y_1|x), P(y_2|x), ..., P(y_n|x)\}$ ，则 $x \in y_k$ 。

那么现在的关键就是如何计算第3步中的各个条件概率。我们可以这么做：

- 1、找到一个已知分类的待分类项集合，这个集合叫做训练样本集。
- 2、统计得到在各类别下各个特征属性的条件概率估计。即 $P(a_1|y_1), P(a_2|y_1), ..., P(a_m|y_1); P(a_1|y_2), P(a_2|y_2), ..., P(a_m|y_2); ...; P(a_1|y_n), P(a_2|y_n), ..., P(a_m|y_n)$ 。
- 3、如果各个特征属性是条件独立的，则根据贝叶斯定理有如下推导：

$$P(y_i|x) = \frac{P(x|y_i)P(y_i)}{P(x)}$$

因为分母对于所有类别为常数，因为我们只要将分子最大化皆可。又因为各特征属性是条件独立的，所以有：

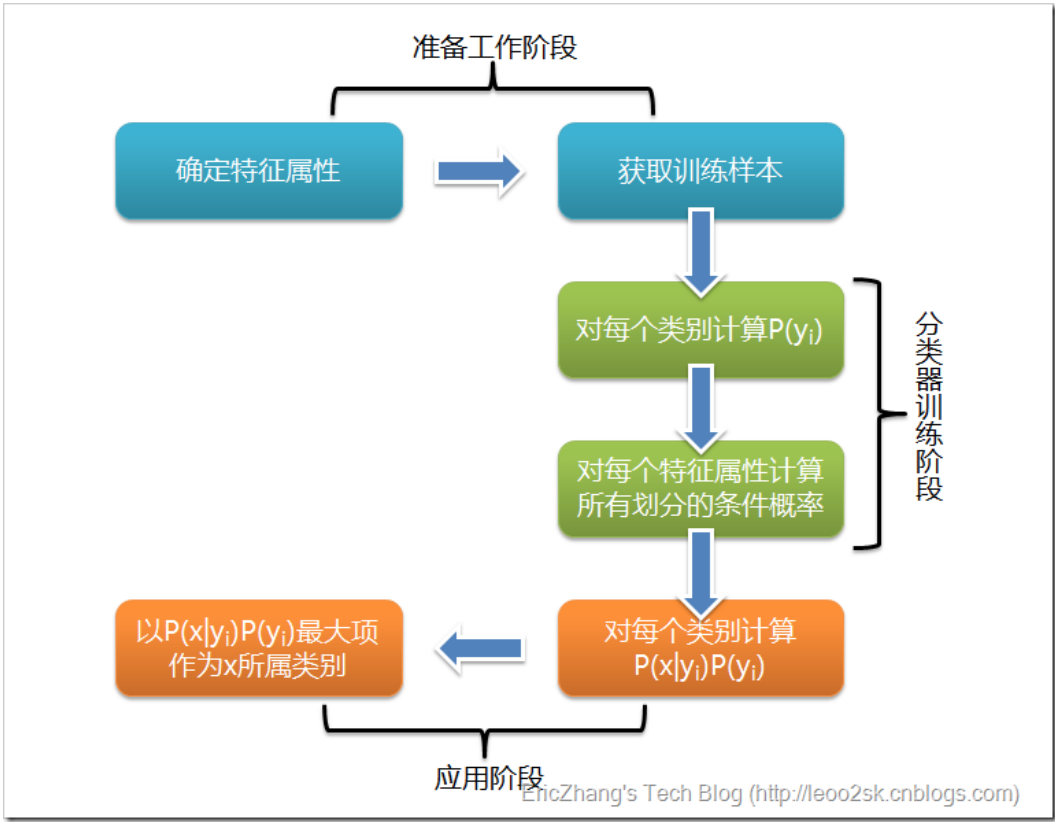
4. 算法杂货铺——分类算法之朴素贝叶斯分类(Naive Bayesian classification)(58)	2009年2月(2)
5. 细说业务逻辑（前篇）(55)	2008年12月(6)
	2008年11月(7)
	2008年10月(3)
	2008年9月(2)
	2008年8月(1)
	2008年7月(9)
	2008年6月(7)
	2008年4月(3)

阅读排行榜
1. 算法杂货铺——分类算法之朴素贝叶斯分类(Naive Bayesian classification)(291995)
2. 算法杂货铺——分类算法之决策树(Decision tree)(175260)
3. ASP.NET MVC案例教程（基于ASP.NET MVC beta）——第一篇：准备工作(140845)
4. 算法杂货铺——分类算法之贝叶斯网络(Bayesian networks)(98400)
5. 使用SeaJS实现模块化JavaScript开发(89894)

分享到：

$$P(x|y_i)P(y_i) = P(a_1|y_i)P(a_2|y_i)...P(a_m|y_i)P(y_i) = P(y_i) \prod_{j=1}^m P(a_j|y_i)$$

根据上述分析，朴素贝叶斯分类的流程可以由下图表示（暂时不考虑验证）：



可以看到，整个朴素贝叶斯分类分为三个阶段：

第一阶段——准备工作阶段，这个阶段的任务是为朴素贝叶斯分类做必要的准备，主要工作是根据具体情况确定特征属性，并对每个特征属性进行适当划分，然后由人工对一部分待分类项进行分类，形成训练样本集合。这一阶段的输入是所有待分类数据，输出是特征属性和训练样本。这一阶段是整个朴素贝叶斯分类中唯一需要人工完成的阶段，其质量对整个过程将有重要影响，分类器的质量很大程度上由特征属性、特征属性划分及训练样本质量决定。

第二阶段——分类器训练阶段，这个阶段的任务就是生成分类器，主要工作是计算每个类别在训练样本中的出现频率及每个特征属性划分对每个类别的条件概率估计，并将结果记录。其输入是特征属性和训练样本，输出是分类器。这一阶段是机械性阶段，根据前面讨论的公式可以由程序自动计算完成。

第三阶段——应用阶段。这个阶段的任务是使用分类器对待分类项进行分类，其输入是分类器和待分类项，输出是待分类项与类别的映射关系。这一阶段也是机械性阶段，由程序完成。

1.4.2、估计类别下特征属性划分的条件概率及Laplace校准

这一节讨论 $P(a|y)$ 的估计。

由上文看出，计算各个划分的条件概率 $P(a|y)$ 是朴素贝叶斯分类的关键性步骤，当特征属性为离散值时，只要很方便的统计训练样本中各个划分在每个类别中出现的频率即可用来估计 $P(a|y)$ ，下面重点讨论特征属性是连续值的情况。

当特征属性为连续值时，通常假定其值服从高斯分布（也称正态分布）。即：

$$g(x, \eta, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\eta)^2}{2\sigma^2}}$$

$$\text{而 } P(a_k|y_i) = g(a_k, \eta_{y_i}, \sigma_{y_i})$$

因此只要计算出训练样本中各个类别中此特征项划分的各均值和标准差，代入上述公式即可得到需要的估计值。均值与标准差的计算在此不再赘述。

另一个需要讨论的问题就是当 $P(a|y)=0$ 怎么办，当某个类别下某个特征项划分没有出现时，就是产生这种现象，这会令分类器质量大大降低。为了解决这个问题，我们引入Laplace校准，它的思想非常简单，就是对没类别下所有划分的计数加1，这样如果训练样本集数量充分大时，并不会对结果产生影响，并且解决了上述频率为0的尴尬局面。

1.4.3、朴素贝叶斯分类实例：检测SNS社区中不真实账号

下面讨论一个使用朴素贝叶斯分类解决实际问题的例子，为了简单起见，对例子中的数据做了适当的简化。

这个问题是这样的，对于SNS社区来说，不真实账号（使用虚假身份或用户的小号）是一个普遍存在的问题，作为SNS社区的运营商，希望可以检测出这些不真实账号，从而在一些运营分析报告中避免这些账号的干扰，亦可以加强对SNS社区的了解与监管。

如果通过纯人工检测，需要耗费大量的人力，效率也十分低下，如能引入自动检测机制，必将大大提升工作效率。这个问题说白了，就是要将社区中所有账号在真实账号和不真实账号两个类别上进行分类，下面我们一步一步实现这个过程。

首先设 $C=0$ 表示真实账号， $C=1$ 表示不真实账号。

1、确定特征属性及划分

这一步要找出可以帮助我们区分真实账号与不真实账号的特征属性，在实际应用中，特征属性的数量是很多的，划分也会比较细致，但这里为了简单起见，我们用少量的特征属性以及较粗的划分，并对数据做了修改。

我们选择三个特征属性： a_1 ：日志数量/注册天数， a_2 ：好友数量/注册天数， a_3 ：是否使用真实头像。在SNS社区中这三项都是可以直接从数据库里得到或计算出来的。

下面给出划分： a_1 ：{ $a \leq 0.05$, $0.05 < a < 0.2$, $a \geq 0.2$ }}, a_2 ：{ $a \leq 0.1$, $0.1 < a < 0.8$, $a \geq 0.8$ }}, a_3 ：{ $a=0$ （不是）， $a=1$ （是）}。

2、获取训练样本

这里使用运维人员曾经人工检测过的1万个账号作为训练样本。

3、计算训练样本中每个类别的频率

用训练样本中真实账号和不真实账号数量分别除以一万，得到：

$$P(C=0) = 8900/100000 = 0.89$$

$$P(C=1) = 110/100000 = 0.11$$

4、计算每个类别条件下各个特征属性划分的频率

$$P(a_1 \leq 0.05 | C=0) = 0.3$$

$$P(0.05 < a_1 < 0.2 | C=0) = 0.5$$

$$P(a_1 > 0.2 | C=0) = 0.2$$

$$P(a_1 \leq 0.05 | C=1) = 0.8$$

$$P(0.05 < a_1 < 0.2 | C=1) = 0.1$$

$$P(a_1 > 0.2 | C=1) = 0.1$$

$$P(a_2 \leq 0.1 | C=0) = 0.1$$

$$P(0.1 < a_2 < 0.8|C = 0) = 0.7$$

$$P(a_2 > 0.8|C = 0) = 0.2$$

$$P(a_2 \leq 0.1|C = 1) = 0.7$$

$$P(0.1 < a_2 < 0.8|C = 1) = 0.2$$

$$P(a_2 > 0.2|C = 1) = 0.1$$

$$P(a_3 = 0|C = 0) = 0.2$$

$$P(a_3 = 1|C = 0) = 0.8$$

$$P(a_3 = 0|C = 1) = 0.9$$

$$P(a_3 = 1|C = 1) = 0.1$$

5、使用分类器进行鉴别

下面我们使用上面训练得到的分类器鉴别一个账号，这个账号使用非真实头像，日志数量与注册天数的比率为0.1，好友数与注册天数的比率为0.2。

$$P(C = 0)P(x|C = 0) = P(C = 0)P(0.05 < a_1 < 0.2|C = 0)P(0.1 < a_2 < 0.8|C = 0)P(a_3 = 0|C = 0) = 0.89 * 0.5 * 0.7 * 0.2 = 0.0623$$

$$P(C = 1)P(x|C = 1) = P(C = 1)P(0.05 < a_1 < 0.2|C = 1)P(0.1 < a_2 < 0.8|C = 1)P(a_3 = 0|C = 1) = 0.11 * 0.1 * 0.2 * 0.9 = 0.00198$$

可以看到，虽然这个用户没有使用真实头像，但是通过分类器的鉴别，更倾向于将此账号归入真实账号类别。这个例子也展示了当特征属性充分多时，朴素贝叶斯分类对个别属性的抗干扰性。

1.5、分类器的评价

虽然后续还会提到其它分类算法，不过这里我想先提一下如何评价分类器的质量。

首先要定义，分类器的正确率指分类器正确分类的项目占有被分类项目的比率。

通常使用回归测试来评估分类器的准确率，最简单的方法是用构造完成的分类器对训练数据进行分类，然后根据结果给出正确率评估。但这不是一个好方法，因为使用训练数据作为检测数据有可能因为过分拟合而导致结果过于乐观，所以一种更好的方法是在构造初期将训练数据一分为二，用一部分构造分类器，然后用另一部分检测分类器的准确率。



本文基于[署名-非商业性使用 3.0](#)许可协议发布，欢迎转载，演绎，但是必须保留本文的署名[张洋](#)（包含链接），且不得用于商业目的。如您有任何疑问或者授权方面的协商，请[与我联系](#)。

好文要顶

关注我

收藏该文



T2噬菌体

关注 - 14

粉丝 - 2866

58

1

荣誉：推荐博客

+加关注

« 上一篇：利用简单的一元线性回归分析估计软件项目开发时间

» 下一篇：算法杂货铺——分类算法之贝叶斯网络(Bayesian networks)

分类：[\[11\]数据挖掘](#)

标签：[算法](#), [数据挖掘](#), [贝叶斯](#), [分类](#)

分享到...