MLAPP-C3

3.离散数据的生成式模型

3.1 引言

3.2 贝叶斯概念学习

3.2.1 似然函数

3.2.2 先验分布

3.2.3 后验分布

3.2.4 后验预测分布

3.2.5 一种更复杂的先验分布

3.3 贝塔——二项式模型

3.3.1 似然函数

3.3.2 先验分布

3.3.3 后验分布

3.3.4 后验预测分布

3.4 狄利克雷——多项式模型

3.4.1 似然函数

3.4.2 先验分布

3.4.3 后验分布

3.4.4 后验预测分布

3.5 朴素贝叶斯分类器

3.5.1 NBC模型训练

3.5.2 使用模型进行预测

3.5.3 log-sum-exp技巧

3.5.4 使用互信息进行特征选择

3.5.5 使用词袋法对文本进行分类（**注：本节更多技术细节请读者参考其他文献**）

3.离散数据的生成式模型

3.1 引言

在章节2.2.3.2，我们讨论了如何将贝叶斯准则应用到生成式分类器，从而实现对一个特征向量**x**的分类：

 （3.1）

使用上述模型的关键之处在于为类条件密度指定一个合适的形式，这将决定我们在每个类别中观察到什么类型的数据。本章，我们将集中讨论观察到的数据类型属于离散型的情况。同时，我们将讨论在这类模型中如何推断未知参数的值。

3.2 贝叶斯概念学习

考虑一个问题：一个小孩如何学习理解一个单词的含义，比如单词“dog”。我们进行如下推测：小孩的父母指向关于这个概念（即单词“dog”）的一些正确的样例（正例），然后说“look at the cute dog!”或者“mind the doggy”等之类的话。然而，很少出现父母指着一个错误的样例，然后说道“look at that non-dog.”当然，错误的样例在主动学习的过程中可以会用到——小孩说“look at the dog”，然后父母纠正道：“that’s a cat, dear, not a dog.”——然而心理学研究表明人们可以只从正确的样例中去理解概念。

我们可以将学习一个单词的含义等价为**概念学习**（concept learning），后者又等价于二元分类。为了说明后者，我们定义如果*x*是关于概念*C*的一个正例，则令，否则，令。我们的目标是通过学习得到这个指示函数*f*，该函数定义了哪些样例是属于这个概念*C*的。通过考虑函数*f*的定义（或者说概念*C*中的元素）存在一定的不确定度，我们可以模拟**模糊集合论**（fuzzy set theory），但是是使用标准的概率计算。值得注意的是标准的二元分类问题需要正例和负例的同时存在。相反，我们将设计一种只从正例中进行学习的方式。

为了教学的目的，我们考虑一个简单的关于概念学习的例子，叫做**数字游戏**（number game），这个游戏源自Josh Tenenbaum的博士论文。游戏的过程如下：我首先选择一些简单的数学概念*C*，比如“素数”或者“一个在1到10之间的数”。然后我将给你一系列从概念*C*中随机抽取的正例，然后问你一些新的测试样例是否属于概念*C*。

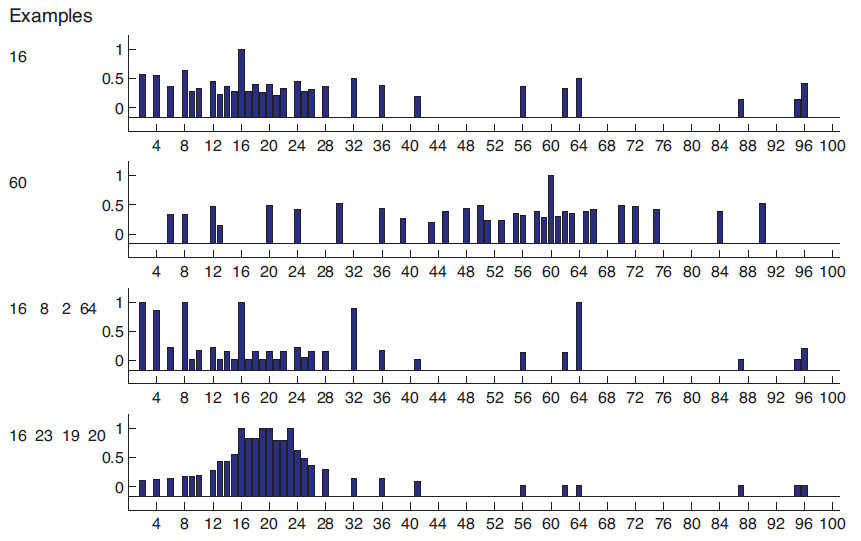


图3.1 基于在数字游戏中参与的8个人的预测结果的分布。前两行代表看到数据和后的统计结果。这两个图说明了分布比较分散的相似性。第3行：看到数据之后的预测结果。该图说明了基于规则的行为（2的幂次方）。第4行：看到数据后的统计结果。这说明了聚集的相似度（接近20的数字）

为了简单起见，我们假设所有的数字都是在1和100之间的整数。现在我告诉你数字“16”是概念中的正例。那么你会觉得哪些数还会是正例？17？6？32？99？显然你很难判断哪个才是正确的，所以你的预测将会十分含糊。那些在某些方面与数字“16”更相似的数字更有可能是正确的，但这里所说的“相似”又该如何定义呢？17与16是相似的，因为它离16近。6与16是相似的，因为它们有一位数完全相同。32与16是相似的，因为它们都是偶数且都是2的幂次方，但数字99好像不相似。所以存在一些数字与16的相似度要更高。我们对上述问题进行形式化表达，引入概率分布，代表对任意的数字，在给定数据的情况下，的概率。被称为**后验预测分布**（posterior predictive distribution）。图3.1（top）展示了参与这个实验的志愿者的预测分布。不难发现，人们基于不同的相似度标准，将与16相似的数作为正确的输出。

现在假设我告诉你8,2和64也是正确的样例。现在你可能会猜测这个隐藏的数学概念是“2的幂次方”。这其实就是**归纳法**（induction）的过程。基于这个假设，预测分布将会更加明确，如图3.1（第三行）所示，大部分的概率质量都分配到了那些2的幂次方上。那如果我给你的数据集，你可能会得到一个不同的**泛化梯度**（generalization gradient）（泛化梯度是指相似性程度不同的刺激引起的不同强度的反应的一种直观表征。它表明了泛化的水平，是泛化反应强度变化的指标），如图3.1（bottom）所示。

我们如何解释这个行为并且在机器学习中去模仿它？归纳的经典方法是假设我们已经有一个关于正确概念的**假设空间**（hypothesis space），比如：奇数，偶数，基于1到100的所有数，2的幂次方，所有以数字*j*（0≤*j*≤9）结尾的数。在假设空间中，所有与数据保持一致的假设构成了**版本空间**（version space）。随着我们看到的例子越来越多，版本空间的大小将会越来越小，我们对那个背后的概念也更加清晰。

然而，版本空间并不是故事的全部。当我们观察到时，有许多满足要求的规则；如何将这些规则进行组合去预测是否成立呢？同样，当我们看到时，为什么你会选择规则“2的幂次方”，而不是选择“所有的偶数”，或者说“除了32以外的所有2的幂次方”呢，所有这些规则都与我们现有的证据（即观察到的数据）保持一致？现在，我们将针对这个问题给出贝叶斯观点的解释。

3.2.1 似然函数

现在我们必须解释，在看到数据时，为什么我们会选择假设，而不是。这里直观的解释在于我们希望避免**可疑的巧合**（suspicious coincidences）：如果真实的概念是偶数，那么我们为什么刚好只看到那些“2的幂次方”呢？

为了形式化上述表达，让我们假设样例是从概念的**延展**（extension）中均匀随机采样得到的。（所谓的概念的延展是指属于这个概念的数字的集合，比如说的延展就是；概念“以9结尾的数”的延展是）。Tenenbaum称这种采样为**强采样假设**（strong sampling assumption），从假设*h*中独立采样*N*个样本（有放回的）概率为：

 （3.2）

这个重要的公式体现了Tenenbaum所说的**尺度原则**（size principle），意思是说模型更倾向于与数据保持一致的最简单的假设。这个原则通常被称为**奥卡姆剃刀**（Occam’s razor）。

为了说明上述原则是如何奏效的，令，我们有，因为在小于100的整数中，只有6个2的幂次方，但，因为有50个偶数。所以的似然大于的似然。在观察到4个样例后，的似然为(1/6)4=7.7×10-4，然而的似然为(1/50)4=1.6×10-7。两个假设之间的**似然比**（likelihood ratio）接近5000:1。似然比量化了我们当初的直觉，即数据如果是从假设中产生的话，那会是一个非常可疑的巧合。

3.2.2 先验分布

假设，基于这个数据集，概念=“除了32以外的2的幂次方”比概念*h*=“2的幂次方”更有可能，因为不需要解释为什么32恰好没出现的巧合。（意思是说如果我们猜测正确的是概念*h*，那么我们需要解释为什么偏偏32没出现在中）。

然而，假设在概念上好像并“不自然”，为了表达这种不自然，我们可以对于这种不自然的概念赋予一个比较低的先验概率。当然，你的先验概率可能与我的不同。这种**主观性**（subjective）是贝叶斯推理饱受争议的原因所在，因为它意味着，一个小孩和一个数学教授会得到不同的答案。事实上，小孩和教授可能不仅有不同的先验概率，而且还会有不同的假设空间。然而，我们做出一些巧妙处理，定义小孩和教授的假设空间是一样的，然后对于那些比较“高级的”概念（即小孩根本不可能接触到的数学概念），将小孩的先验权重设置为0。因此在先验和假设空间之间没有明显的区别（只是两个人的权重值不同罢了）。

尽管先验概率存在争议性，但它却十分有用。如果你被告知服从某些数学规

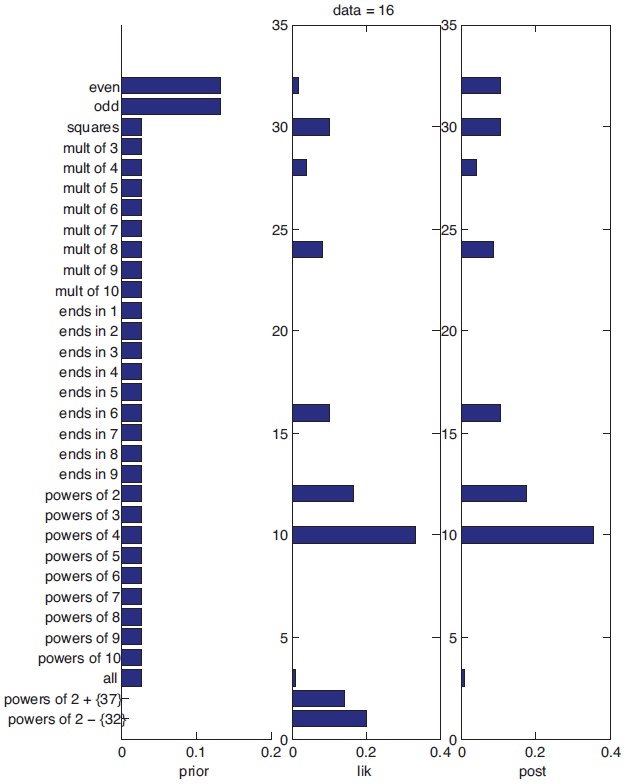


图3.2 情况下的先验分布，似然函数和后验分布。

则的一些数字，比如1200,1500,900和1400，你可能会认为400有可能也服从这个规则，但1183不可能。但是如果你被告知这些数字是一个健康人的胆固醇水平，那你很有可能认为400是不可能的，但1183是有可能的。所以我们可以发现先验分布形成了一个机制，在这个机制下，我们将影响某个问题的一些背景知识考虑进来。如果没有这些背景知识，快速的学习（比如从小的样例中进行学习）是不可能的。

那么我们又该如何使用先验概率呢？出于演示的目的，我们使用一个简单的先验分布，在该先验分布中，30个简单的数学概念服从均匀分布，比如说：“偶数”，“奇数”，“素数”，“以9结束的数”等等。为了让事情更加有趣，我们赋予“奇偶”两个概念更多的先验权重，同时，我们包含两个“不自然”的概念，比如说“2的幂次方以及37”和“2的幂次方除了32”，但是赋予这两个概念较低的先验权重。图3.2（左图）绘制了该先验分布。我们将在后面的内容中讨论更加复杂的先验分布。

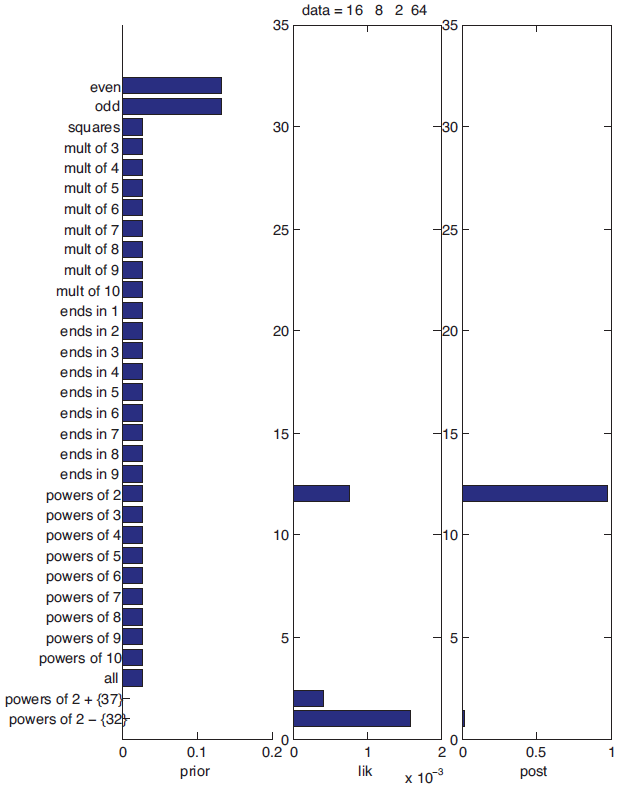


图3.3 情况下的先验分布，似然函数和后验分布

3.2.3 后验分布

后验分布只是似然函数与先验分布的乘积并进行归一化后的结果。在当前的数字游戏的背景下，我们有：

 （3.3）

其中=1的充要条件是所有的数据都在假设*h*的延展中。图3.2绘制了时，对应的先验分布，似然函数和后验分布。不难发现后验分布是先验分布和似然函数的组合。对于大部分的概念，其先验分布都是均匀的，所以后验分布正比于似然函数。然而，那些不自然的概念“2的幂次方以及37”和“2的幂次方除了32”由于先验权重很低导致最终的后验分布权重很低，尽管它们有较高的似然。相反，奇数和偶数的先验权重较高，但由于其似然函数很低，所以最终的后验权重较低。

图3.3绘制了的情况下的先验分布，似然函数和后验分布。此时，似然函数更加集中在概念“2的幂次方”周围，并且最终支配了后验分布。这样我们最终可以找出正确的那个概念。（此时，我们发现对那些“不自然”概念赋予低的先验权重的必要性，否则我们将在现有数据上出现过拟合——选择概念“2的幂次方除了32”）。

通常情况下，当我们拥有足够多的数据时，后验将会集中分布于一个单独的概念，这个单独的概念被称为**最大后验概率**（maximum a posteriori，MAP）估计：

 （3.4）

其中为后验分布的众数，为**狄利克雷测度**（Dirac measure），定义为：

 （3.5）

值得注意的是，MAP估计可以写成：

 （3.6）

因为似然函数项与*N*呈指数幂关系，先验分布保持不变。当数据量越来越多时，MAP估计将收敛于**最大似然估计**（maximum likelihood estimate, MLE）：

 （3.7）

换句话说，如果我们有足够多的数据，我们会发现**数据压倒先验**（data overwhelms the prior）的情况。在这种情况下，MAP估计将收敛于MLE。

如果真实的假设就在起初的假设空间中，那么MAP/ML估计将收敛于这个真实假设。因此我们说贝叶斯推断（最大似然估计）是一致性估计（6.4.1将讨论相关细节）。我们同样称假设空间是**极限状态下可辨识的**（identifiable in the limit），意味着我们可以在数据无穷多的极限情况下发现“真相”。如果我们的假设空间不包含“真相”，那么我们将收敛于一个无限接近“真相”的假设。然而，对这里的“接近”进行形式化的表达已经超出了本章的范围。

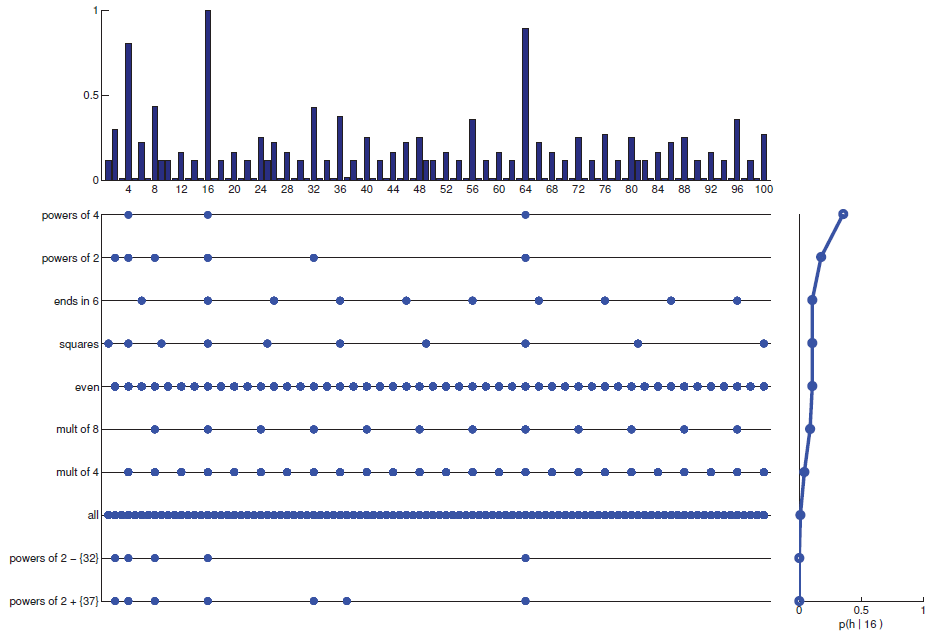


图3.4 在看到数据的情况下，所有假设的后验分布（图右）和相应的预测分布。图中的点说明该点与相应的假设是一致的。右侧的图表示每个假设h的权重。通过对每个点进行加权求和，我们得到上图顶端的

3.2.4 后验预测分布

后验分布是我们关于这个世界本质的**信念状态**（belief state），检验这种信念是否合理的方式是使用它们去预测客观世界中具备可观察性的量（这是检验一个方法是否具有科学性的基础，译者注：**说白了就是你猜测的概念好不好，要看你预测新的样本时到底准不准**）。特别的，在本章中后验预测分布定义为：

 （3.8）

上式只是基于每一个假设所作出的预测的加权平均，称为**贝叶斯模型平均**（Bayes model averaging）。图3.4说明了这一点。图中下方的点展示了每个假设的预测结果；图形右边的垂直曲线显示了每个假设的权重值。如果我们将每一行乘上相应的权重并相加，将得到顶部的分布。

当我们有一个小的或者模糊的数据集时，其后验分布也会是含糊不清的，从而导致概率质量分布较广。然而，我们一旦“将问题解决了”（译者注：即得到了那个隐藏的概念），后验分布将退化成脉冲函数，且以MAP估计值为中心。在这种情况下，预测分布将变成：

 （3.9）

上式被称为预测概率密度的**点估计**（**plug-in approximation）**，因其简单性而被广泛使用。然而，通常情况下，这种方式会低估我们的不确定度，与使用贝叶斯模型平均方法相比，我们的预测结果将不会很平滑。在本书后面的内容中，我们将看到关于这一点的更多例子。

尽管基于最大后验估计的学习很简单，它却不能解释从基于相似度的推理（不确定的后验）到基于规则的推理（确定的后验）的逐渐转变。举例来说，假设我们首先看到了数据集为，那么基于之前的简单的先验分布，最小的一致性假设显然是“所有4的幂次方”，当我们使用这个假设去预测时，只有4,16和64三个数的概率值为非零。显然，这种情况下是过拟合的（译者注：因为数据太少了，我们过分了拟合了数字16的规律）。随后，我们会看到更多数据，比如说给定的数据集变成了，在这种情况下，基于最大后验概率估计原则的假设为“所有2的幂次数”。显然，基于这个假设将有更多的数值得到的概率值不为0。也就是说采用点估计的方式，随着看到的数据不断增加，导致后验预测分布的变化趋势**由窄变宽**。相反，如果采用贝叶斯方法（即贝叶斯模型平均），随着看到的数据量增加，后验预测分布的变化趋势是**由宽变窄**的，而这更加符合人们直观的感受。特别的，当我们的数据集为时，我们的后验分布中有许多假设的概率值都不为0，所以相应的预测分布也比较宽。然而，当我们看到的数据集变成时，后验分布的概率质量将主要集中在一个假设上，所以导致预测分布变窄。所以，通过上面的一个小的案例，我们发现，点估计与贝叶斯模型平均方法在最终的预测分布上存在很大的不同，尽管随着数据量的增加，两者会收敛于同一个结果。

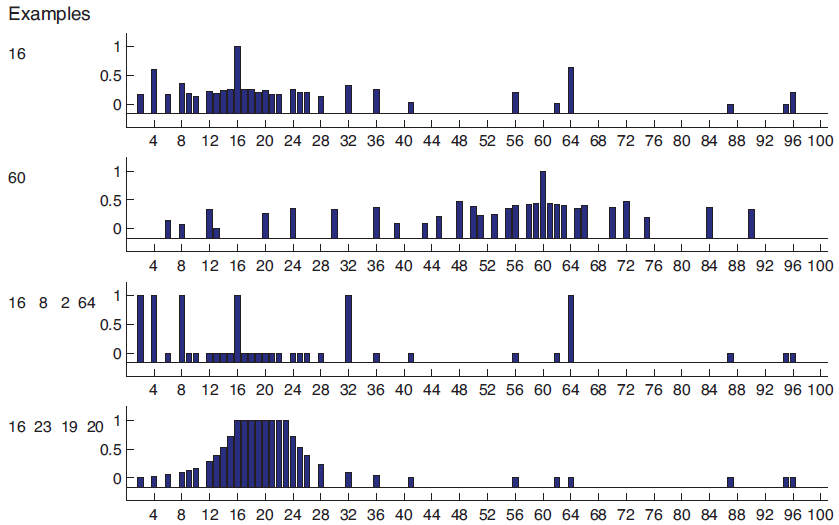


图3.5 基于全假设空间得到的预测分布。对比图3.1，基于贝叶斯模型平均得到的预测结果只针对那些人类数据可用的；这也是为什么最上面的分布较图3.4比较稀疏。

3.2.5 一种更复杂的先验分布

为了模拟人类的行为，Tenenbaum分析了一些实验数据，这些数据反映了人们是如何衡量数字之间的相似度的，基于这些分析，使用了一个稍微复杂的先验概率分布。在这个先验分布中，不仅包含与上文相似的一些数学概念，还包括在*n*和*m*之间的所有区间（1≤*n*，*m*≤100）。（值得注意的是，这些假设并不是互斥的）。因此最终的先验分布是两种先验的混合，一种基于数学规则，一种基于区间：

 （3.10）

在上述模型中唯一的自由参数就是相对权重。只要（即更多的人相信最终的假设应该是一个数学规则），最终的结果对的大小并不会太敏感。基于这个更大的假设空间，模型的预测分布如图3.5所示。它与如图3.1所示的人类的预测分布惊人的相似，尽管它并不适应人类预测的数据（除了假设空间的选择）。

3.3 贝塔——二项式模型

在前面章节中我们介绍了一个数字游戏，在这个游戏中，给定了一系列的离散观察值，我们需要根据给定的观测值，在有限的假设空间中推理出一个分布。在这个过程中，我们的计算十分简单：我们只需要加法、乘法和除法的一些操作。然而，在很多应用中，未知参数是连续的，也就是说假设空间的大小为*K*，或者是它的一个子集。其中*K*是参数的数量。这使得数值计算变得复杂，因为我们需要将求和的操作变成积分。然而，其基本的思想是一致的。

为了说明这个问题，我们将首先考虑抛硬币的实验，在给定一系列试验结果的前提下，推断出硬币朝上的概率。尽管这个看起来很普通，但是结果证明这个模型是本书后面将介绍的许多方法的基础，包括朴素贝叶斯分类器，马尔科夫模型等等。同时这个试验在历史上也是很重要的，它是贝叶斯在1763年的原始论文中所分析的案例。

我们将遵循我们目前所熟悉的方法：**已知先验分布和似然函数，推理出后验分布和后验预测分布。**

3.3.1 似然函数

假设（伯努利分布），其中代表硬币的正面朝上，代表反面朝上。为速率参数（即正面朝上的概率）。如果试验结果服从独立同分布，则似然函数的形式为：

 （3.11）

其中表示正面朝上的次数，表示反面朝上的次数。这两个统计量被称为数据的**充分统计量**（sufficient statistics），因为只需要知道这两个统计量就可以从数据中推断出参数（当然和也可以作为充分统计量）。

更加正式的表达，我们称如果，则称**s**()为数据的充分统计量。如果我们采用一个均匀的先验分布，那么上式等价于。因此如果我们拥有两个数据集且具备相同的充分统计量，那么我们将推理得到一样的参数值。

现在假设数据由在*N*=*N*1+*N*0（*N*值大小固定）次试验中观察到的正面朝上的次数*N*1组成。在这种情况下，,其中Bin代表二项式分布，其概率质量函数定义为：

 （3.12）

因为二项式系数与参数无关，所以二项式分布与伯努利分布的似然函数是一样的（差一个系数）。所以无论我们观察到的是统计量=(*N*1,*N*)（对应二项式分布）还是*N*次试验的序列结果={*x*1,…,*xN*}（对应伯努利分布），我们关于参数的推理结果都是一样的。

3.3.2 先验分布

我们需要一个先验分布，其定义域为区间[0,1]。为了让数学计算更加简单，使先验分布与似然函数的形式保持一致是一件很方便的事情，比如说我们的先验分布具备如下形式：

 （3.13）

上式包含一些先验参数和。在这种情况下，我们可以很容易的计算出后验分布：

 （3.14）

如果先验分布与后验分布有相同的形式，我们称先验分布为相应的似然函数的**共轭先验**（conjugate prior）。共轭先验因其便于计算且容易解释而被广泛使用，我们将在后面看到这一点。

在伯努利分布中，其共轭先验为我们在2.4.6节介绍的贝塔分布：

 （3.15）

先验分布中的参数*a*,*b*被称为**超参数**（hyper-parameters）。通过设置超参数我们可以在模型中融入我们的一些背景知识。比如说，根据经验，我们相信参数的期望值为0.7，标准差为0.2，那么我们就可以设置*a*=2.975，*b*=1.275。或者说我们相信的期望值为0.15，且位于区间（0.05,0.30）的概率值为0.95，则设置*a*=4.5,*b*=25.5。

如果我们对参数没有一点背景知识，只知道它位于区间[0,1]，那么我们就可以使用一个均匀的先验分布，它是一种不包含任何信息的先验分布（章节5.4.2给出更多细节）。在贝塔分布中，我们可以设置*a*=1,*b*=1，从而实现均匀先验分布。

3.3.3 后验分布

似然函数与贝塔先验分布相乘将得到后验分布，我们有：

 （3.16）

特别的，后验分布是通过将先验分布中的超参数与经验计数（**注意此处的经验计数为我们实验过程中的计数**，即*N*1和*N*0）相加得到的。出于这个原因，我们称超参数为**伪计数**（pseudo counts）。先验分布的强度，又被称为**等价样本尺寸**（equivalent sample size），等于伪计数的和，即；这与经验数据的尺寸大小*N*1+*N*0=*N*扮演相同的角色。

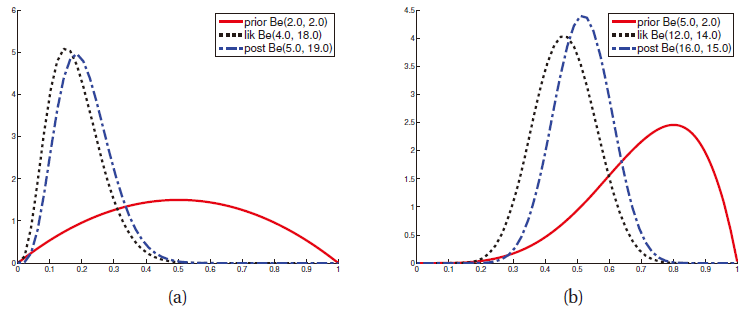


图3.6 (a)基于充分统计量N1=3,N0=17的二项式分布更新先验分布Beta(2,2)，形成后验分布Beta(5,19)。(b)基于充分统计量N1=11,N0=13的二项式分布更新先验分布Beta(5,2)，形成后验分布Beta(16,15)。

图3.6(a)展示了一个例子，我们通过一个尖的似然函数（即有大量的采样样本）更新一个较弱的Beta(2,2)先验分布，不难发现后验分布与似然函数几乎一样，因为采样得到的经验数据“碾压”了先验分布。图3.6(b)给出了另一个例子，我们通过一个较尖的似然函数更新一个较强的Beta(5,2)先验分布，现在我们发现后验分布处在了先验分布与似然函数之间。

值得注意的是，序列化地更新后验分布等价于在一个单独的批次中进行更新。为了明白这一点，假设我们手上有两个数据集****和，且其充分统计量分别为,和。令和为联合数据集的充分统计量。在批次更新模式中，我们有：

 （3.17）

在序列更新模式中，我们得到的最终结果为：

 （3.18）

 （3.19）

 （3.20）

这使得贝叶斯推理特别适合用在**在线学习**（online learning）的相关领域，我们将在8.5.5节看到更多相关细节。

3.3.3.1 后验分布的期望和众数

根据式2.56，我们知道最大后验概率估计（即Beta分布的众数）为：

 （3.21）

如果我们使用均匀先验分布，最大后验估计（MAP）将退化为最大似然估计（MLE），也就是正面朝上的次数的经验占比：

 （3.22）

上式的结果很直观，当然我们也可以通过最大化似然函数（3.11）得到上式最大似然解（**读者可以证明之**）。

相比之下，后验分布的期望为：

 （3.23）

众数和期望之间的差别很重要，这一点将在后文中证明。

现在我们将展示的是后验分布的期望是先验分布期望与最大似然估计MLE的凸组合，这就契合了一个概念：在我们开始的信念（先验分布）与实验数据所告诉我们的事实（极大似然）之间，后验分布在这两者之间做出妥协。

令为先验分布的等价样本尺寸，它控制着先验分布的强度，令先验分布的期望为，则后验分布的期望为：

 （3.24）

其中为先验分布的等价样本尺寸与后验分布的等价样本尺寸之间的比例。越小，先验分布的强度越弱。所以当，**后验分布的期望值将趋向于最大似然解**。读者也可以自己探索如下结论：后验分布的众数是先验分布的众数与最大似然解之间的凸组合，所以**最大后验估计同样收敛于最大似然估计**。

3.3.3.2 后验分布的方差

期望和众数都是关于参数的点估计，知道我们在多大程度上信任这个估计值是十分有用的。后验分布的方差就是为了衡量这种信任程度的。Beta后验分布的方差为：

 （3.25）

当时，我们可以对上面公式进行简化：

 （3.26）

所以，估计值的“误差条”（即后验分布的标准差）为：

 （3.27）

不难发现不确定度下降的速率为。值得注意的是，当时，不确定度最大，当接近0或者1时，不确定度最小。**这就意味着“确定一个硬币是不均匀的”要比“确定一个硬币是均匀的”更加容易。**

3.3.4 后验预测分布

截止目前，我们将注意力主要集中在对未知参数的推理上。现在让我们将目光转移到对未来可观察数据的预测上。

考虑在服从Beta(*a*,*b*)的后验分布的情况下，预测下一次抛掷硬币试验中，正面朝上的概率。我们有：

 （3.28）

 （3.29）

所以不难发现，后验预测分布的期望等价于（在当前情况下）基于后验分布期望的点估计，即：。

3.3.4.1 过拟合与黑天鹅悖论

假设我们使用最大似然估计进行点估计，即。不幸的是，这种近似的方式在样本很少的情况下性能十分糟糕。举例来说，我们在连续的3次硬币试验中看到3次反面朝上，那么根据最大似然估计的原则，我们有。然而，如果使用这个参数估计值，我们将认为硬币正面朝上是不可能发生的事情。这被称为**零计数问题**（zero count problem）或者**稀疏数据问题**（sparse data problem），这种问题在样本数量很小的情况下经常出现。或许我们会认为在大数据时代，这种情况基本上不会发生，但是一旦我们基于某种准则对数据进行拆分时——比如某个特定的人已经从事某个特定活动的次数——样本的尺寸将变得很小。这个问题会在诸如在网页上进行个性化推荐的任务中出现。因此贝叶斯方法依然是有用的，哪怕是在大数据时代。

零计数问题与哲学中的**黑天鹅悖论**（black swan paradox）十分相似。这是基于古代西方人的基本观念：天鹅都是白色的。在当时的那种环境下，黑天鹅一般是对那些不可能发生的事的隐喻（黑天鹅于17世纪被欧洲探险家在澳大利亚发现）。术语“黑天鹅悖论”一词最早由著名的科学哲学家Karl Popper创造。这个悖论被用于说明在归纳（induction）中出现的问题，所谓归纳法就是如何基于历史上出现的特定的观测值得到关于未来的一般性结论。

让我们针对上述问题推导出一个简单的贝叶斯解决方案。我们将使用一个均匀先验分布，即*a*=*b*=1，在这种情况下，基于后验分布期望进行点估计，这个方法符合**拉普拉斯继承法则**（Laplace’s rule of succession）：

 （3.30）

上述方法向我们展示了一种常规操作：在经验计数上加1，基于归一化的结果进行点估计，该技术被称为**加一平滑**（add-one smoothing。）（值得注意的是加一平滑是针对最大似然解作出的调整，使用最大后验估计参数进行点估计不具备平滑效应，因为最大后验估计参数为，如果*a*=*b*=1，它将退化成最大似然估计。）

3.3.4.2 预测未来多次试验的结果

假设我们对未来*M*次试验中正面朝上的次数*x*感兴趣。定义为：

 （3.31）

（3.32）

不难发现，上式中的积分项就是分布Beta(*a*+*x*, *M*-*x*+*b*)的归一化常数。因此：

 （3.33）

所以后验预测分布由下式给出，即所谓的**贝塔——二项式**（beta-binomial）分布：

 （3.34）

该分布的期望和方差为：

 （3.35）

如果*M*=1，则*x*∈{0,1}，不难发现此时期望退化成，与式3.29保持一致。

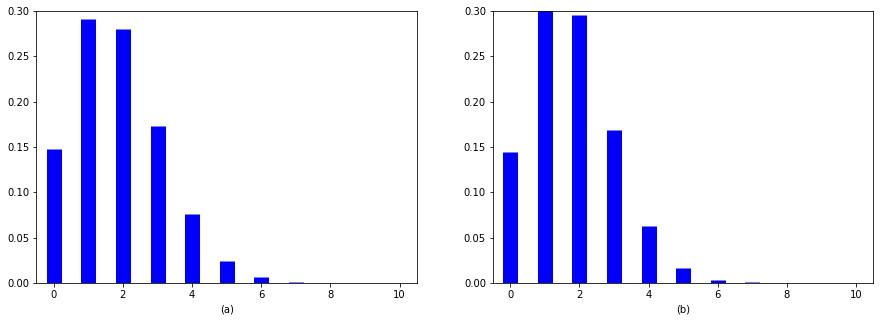


图3.7. (a)看到*N*1=3,*N*0=17后的后验预测分布；（b）基于MAP的点估计。图形由程序**betaBinomPostPredDemo**生成

这个过程如图3.7(a)所示。我们以Beta(2,2)作为先验分布，当看到*N*1=3和*N*0=17时，绘制出后验预测密度。图3.7(b)绘制了基于MAP估计值作出的预测结果。不难发现，通过贝叶斯方法进行的预测具备较长的尾域，其概率质量分布得更广，因此不太倾向于过拟合和黑天鹅悖论。

**附：关于上述问题的个人解释**

在图3.7中我们分别基于贝叶斯模型平均法对未来10次试验中（投掷硬币）正面朝上的次数的分布（图(a)）和基于最大后验概率估计进行的点估计。不难发现，当我们看到经验计数*N*1=3和*N*0=17时（看到这个数据后，我们有理由相信这个硬币在未来的10次投掷中，正面朝上的次数很多的概率很低），基于贝叶斯模型平均法进行的预测有较高的尾域概率（即正面朝上的次数较多的概率相较于图（b）还是比较高的），而基于MAP估计进行的点估计显然很符合我们在看到经验计数后的直观感受，即所谓的过拟合。

3.4 狄利克雷——多项式模型

在上一章节中，我们讨论了如何推理一个硬币正面朝上的概率。本节，我们将泛化这些结果，去推理一个具有*K*个面的骰子面*k*朝上的概率。正如我们在后面将会看到的，这个方法被广泛地应用于文本数据，生物序列数据等的分析。

3.4.1 似然函数

假设我们抛掷骰子*N*次，观察到的结果为={*x*1,…,*xN*}，其中*x*i∈{1,…,*K*}。假设所有的数据独立同分布，则似然函数的形式为：

 （3.36）

其中为面*k*朝上的总次数（这是该模型的充分统计量）。多项式分布的似然函数与上述模型的似然函数形式上是一致的（相差一个与参数无关的常数项）。

3.4.2 先验分布

因为参数向量位于一个*K*维空间中，它是一个概率单纯形（详见2.5.4）。我们需要一个先验分布，其定义域也是一个概率单纯形。理想情况下，我们还希望这是一个共轭先验。幸运的是，狄利克雷分布（详见2.5.4）满足这两个标准。所以我们使用如下所示的先验分布：

 （3.37）

3.4.3 后验分布

将先验分布与似然函数相乘，得到的后验分布同样服从狄利克雷分布：

 （3.38）

 （3.39）

 （3.40）

我们发现后验分布通过将先验分布中的超参数（伪计数）与经验计数*N*k相加即可得到。

我们可以通过微积分的方式计算出后验分布的众数（即最大后验概率估计），然而，前提是必须满足约束条件（读者可以思考一下为什么我们不需要显示地表达约束）。我们使用**拉格朗日乘子法**（Lagrange multiplier）来解决这个问题。含约束的目标函数，或者**拉格朗日算符**（Lagrangian）的形式为对数似然函数加上对数先验分布和约束条件：

 （3.41）

为了简化符号书写，我们定义。对上式关于参数求偏导，得到：

 （3.42）

关于参数求偏导，得到：

 （3.43）

 （3.44）

我们对式（3.44）两边求和，得到：

 （3.45）

 （3.46）

其中为先验分布的等价样本尺寸。所以，最大后验概率估计为：

 （3.47）

该式与式2.73保持一致。如果我们使用一个均匀先验分布，即，我们得到最大似然估计：

 （3.48）

这是面*k*朝上次数的经验占比。

3.4.4 后验预测分布

对于一个单独的multinoulli试验（即一次试验有多次结果），其后验预测分布为：

 （3.49）

 （3.50）

 （3.51）

其中表示出了以外的所有分量。

上述表达式避免了我们在3.3.4.1节看到的零计数问题。事实上，这种形式的贝叶斯平滑对于multinomial情况更加重要，因为当我们将数据分到多个类别（大于2）中时，数据的稀疏性更容易出现。

3.4.4.1 工作案例：使用词袋法的语言模型

使用狄利克雷——多项式模型实现贝叶斯平滑的一个应用是**语言模型**（language modeling）。在语言模型中，我们预测在一个序列中下一个可能出现的单词是什么。此处我们将采用一个非常简单的方法，假设第*i*个单词,下一个单词的出现与否与其他所有单词彼此独立，但同时服从分布。这被称为词袋法模型。给定一系列历史单词，我们该如何预测下一个出现的单词是哪一个呢？

举例来说，假设我们观察到了下面的序列（童谣的一部分）：

Mary had a little lamb, little lamb, little lamb,

Mary had a little lamb, its fleece as white as snow

进一步的，假设我们的语料库由如下单词组成：

Mary lamb little big fleece white black snow rain unk

1 2 3 4 5 6 7 8 9 10

其中**unk**表示unknown（未知），代表所有在语料库中没有出现的单词。为了对童谣的每一行进行编码，我们首先去掉所有的标点符号，去掉那些**停用词**（stop words），比如说“a”，“as”，“the”等。我们也可以进行**词干提取**（stemming），意味着将单词转换为它最基本的形式，比如将复数单词后的“s”去掉，或者将动词后面的“ing”去掉（比如raining变成rain）。在上面的例子中，没有单词需要进行词干提取。最后，我们将每个单词替换为它们在语料库中的编号：

1 10 3 2 3 2 3 2

1 10 3 2 10 5 6 8

现在我们忽略单词之间的顺序，计算每个单词出现的频次，形成一个关于单词频次的统计直方图：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 单词 | mary | lamb | little | big | fleece | white | black | snow | rain | unk |
| 计数 | 2 | 4 | 4 | 0 | 1 | 1 | 0 | 1 | 0 | 3 |

利用*Nj*表示单词*j*出现的次数。如果我们使用表示关于参数的先验分布，其后验预测分布为：

 （3.52）

如果我们设置，将会得到：

（3.53）

预测模型的众数为*X*=2（“lamb”）和*X*=3（“little”）。值得注意的是单词“big”，“black”和“rain”在历史序列中并没有出现，但模型预测的概率值并非等于0，说明这些单词在未来还是有可能出现的，只是其概率值比较低。我们将会在后面的内容中看到更复杂的语言模型。

3.5 朴素贝叶斯分类器

本节，我们将讨论如何对一个具有离散特征值的向量进行分类，其中*K*表示特征可取的数值数量，*D*表示特征的数量。我们将使用一个**生成式模型**方法。这就要求我们指定类条件概率分布。最简单的方式是假设在给定类别的前提下特征之间是**条件独立**（conditionally independent）的。这将允许我们将类条件密度写成一维概率密度的乘积：

 （3.54）

相应的模型被称为**朴素贝叶斯分类器**（naïve Bayes classifier, NBC）。

我们称这个模型是“朴素”的，是因为在实际使用中我们并不要求特征之间是彼此独立的，哪怕是基于类别标签的条件独立。然而，尽管朴素贝叶斯假设并不正确，但这种分类器的工作效果往往很好。一个理由是因为这个模型十分简单，其参数数量的数量级只有*O*(*CD*)（*C*为类别数量，*D*为特征数量），因此它不太容易过拟合。

类条件概率密度的形式与每个特征的类型有直接关系。我们给出一些不同的概率形式：

* 对于实数域的特征，我们可以使用高斯分布：，其中表示特征*j*在类别*c*中的期望，表示相应的方差。
* 对于二元特征*xj*∈{0,1}，我们可以使用伯努利分布：，其中表示在类*c*中特征*j*发生的概率。这通常又被称为**多变量伯努利朴素贝叶斯**（multivariate Bernoulli naïve Bayes）模型。我们将会在下文给出一个应用案例。
* 对于类别特征*xj*∈{1,…,*K*}，我们可以使用多项式分布：，其中表示在类*c*中*xj*可能的*K*个取值的统计直方图（即取不同值的概率）。

显然，我们可以处理不同类型的特征，或者使用不同的概率分布假设。当然，我们也可以很容易将不同类型的概率分布混合起来使用。

3.5.1 NBC模型训练

现在我们讨论如何训练朴素贝叶斯分类器。所谓训练一个模型，往往是意味着求取关于模型中待定参数的MAP或者ML估计。然而，我们也会讨论如何求解关于参数的整个后验分布。

3.5.1.1 NBC的MLE

对于单个数据而言，其概率值为：

 （3.55）

对于*N*个独立同分布数据构成的数据集，我们有：

****

对上式取对数，得到（**附**：**关于式3.56的最后一步推导，读者可以尝试通过先展开再合并的方式完成。**）：

 （3.56）

不难发现对数似然函数分解成不同的项，其中一项包含参数（包含*C*个分量，分别表示每个类的先验概率），包含*DC*个项目的参数。因此我们可以对不同的参数进行单独的优化。

根据式3.48，不难发现类的先验概率的MLE为：

 （3.57）

其中为所有样本中属于类*c*的数量。

参数的MLE与特征服从什么样的类条件概率分布有关。为了简单起见，不妨假设所有特征都是二元特征，即。在这种情况下，MLE为：

 （3.58）

其中为在类*c*中特征*j*出现的样本的数量。

|  |
| --- |
| **算法 3.1**：对于一个具备二元特征的样本进行NBC训练 |
| 1. *Nc*= 0, *Njc*= 0 ;  2. **for** *i* = 1: *N* **do**  3. *c* = *yi* //第*i*个样例的标签  4. *N*c := *N*c + 1 ;  5. **for** *j* = 1: *D* **do**  6. **if** *xij* = 1 **then**  7. *Njc*:= *Njc*+ 1  8. |

实现上述模型的训练十分简单：算法3.1给出了程序的伪代码。这个算法的时间复杂度为*O*(*ND*)。这种方法也可以泛化到那些具有混合类型的特征上。这种简单性也是该方法被广泛使用的理由之一。

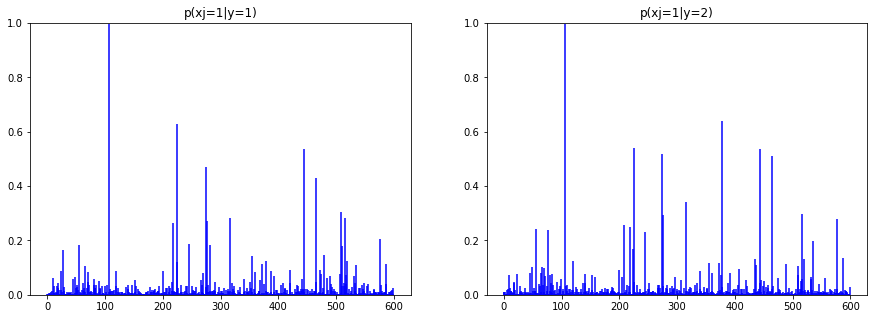


图3.8 类条件密度，两个类别分别对应“X windows”和“MS windows”。图形由程序**naiveBayesBowDemo**生成。

图3.8给出了一个例子，其中数据中拥有2个类别，600个二元特征，分别表示在词袋法模型中某个单词是否出现。图形对在两个类别的情况下向量进行了可视化展示。在索引值为107的地方出现的大的峰值对应单词“subject”，它在两个类别中出现的概率都为1。（我们将在3.5.4节讨论如何过滤掉这些不提供有价值信息的特征。）（注：所谓没有价值就是指这个单词在两个类中是肯定出现的，那么它对我们的分类任务就起不到作用了）

3.5.1.2 贝叶斯方法下的朴素贝叶斯

最大似然解的麻烦之处在于它容易过拟合。举例来说，考虑图3.8中所展示的例子：对应单词“subject”的特征（称其为特征*j*）在两个类中都出现了，所以我们估计。如果我们遇到一封新的邮件但其中并没有这个单词，那么将会发生什么情况呢？我们的算法将会失效，因为我们将发现对于任何一个类，即新的邮件不属于任何一个类。这是我们在章节3.3.4.1中所提及的黑天鹅悖论的另一种体现。

**附：为了方便读者直观的了解上述内容，我们给出了如下的案例：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **label** | **Word\_1** | **Word\_2** | **…** | **Subject( *j* )** | **Word\_n-1** | **Word\_n** |
| C1 | 1 | 0 | … | 1 | 1 | 1 |
| 0 | 1 | … | 1 | 0 | 0 |
| C2 | 0 | 0 | … | 1 | 0 | 1 |
| 1 | 1 | … | 1 | 0 | 0 |
| 0 | 1 | … | 1 | 1 | 0 |

表格中为模型的训练样本，共有两类数据C1和C2，每一类文本中单词“subject”始终存在，那么根据3.5.1.1介绍的模型训练方法，我们有，那么当我们遇到一个新的文本**x**，其中没有单词“subject”时，我们将遇到如下情况：



同样我们也可以得到。

针对过拟合的一个简单解决方案是使用贝叶斯方法。为简单起见，我们使用一个因子分解形式的先验分布：

 （3.59）

针对参数我们使用先验分布，针对每个参数，我们使用先验分布。一般情况下，我们取，，此时对应于前文所提及的加一平滑或者拉普拉斯平滑。

将式3.56的似然函数与式3.59的先验分布组合，得到因子分解形式的后验分布：

 （3.60）

 （3.61）

 （3.62）

换句话说，为了计算后验分布，我们只需要利用似然函数中的经验计数去更新先验计数。基于算法3.1进行必要的调整以使用当前的方法也是很直接的。

3.5.2 使用模型进行预测

在测试阶段，我们的目的是计算

 （3.63）

在贝叶斯模型平均的方法中，我们需要对所有未知参数进行积分：

 （3.64）

 （3.65）

幸运的是，计算上式是比较简单的，尤其是当后验分布服从狄利克雷分布。根据式3.51，我们知道后验预测分布的加权平均可以简单的基于后验分布的期望值进行点估计。所以：

 （3.66）

 （3.67）

 （3.68）

其中。

如果我们基于一个单独的点作为对后验分布的估计，即，其中可能是ML或者MAP估计值，则后验预测分布就可以直接基于这些参数进行计算，从而形成一个几乎相同的准则：

 （3.69）

区别在于我们只是将后验分布的期望替换成后验分布的众数（即MAP）或者MLE 。然而，这一点小小的区别在实际过程中却十分重要，因为利用后验分布的期望进行预测所导致的过拟合更小（见3.3.4.1）。

3.5.3 log-sum-exp技巧

现在我们讨论一个重要的实践细节，这个细节在任何一种生成式分类器中都会遇到。我们可以使用式2.11计算样本所属类别的后验分布，前提是使用合适的类条件密度（比如点估计）。不幸的是，公式2.11在实现过程中可能会因为**数值下溢**（numerical underflow）而失败。问题在于通常情况下是非常小的数，尤其当是一个高维向量时，因为我们要求，所以我们看到任何一个特定的高维向量的概率是很小的。该问题的解决方案是当我们使用贝叶斯公式时对上式取对数：

 （3.70）

 （3.71）

然而，上式要求我们计算下式：

 （3.72）

可是我们并不能对对数进行求和（即）。幸运的是，我们可以将真数（log*N*中*N*为真数）中的最大项提取出来。比如说：

 （3.73）

一般情况下，我们有：

 （3.74）

其中。这被称为log-sum-exp技巧，被广泛使用。

该技巧在算法3.2中被使用，该算法给出了利用NBC计算的伪代码。值得注意的是，如果我们只是为了计算，我们并不需要使用这种技巧。因为我们只需要求解使最大的。

|  |
| --- |
| **算法 3.2**：对于一个具备二元特征的样本利用NBC进行预测 |
| 1 **for** i = 1 : *N* **do**  2 **for** c = 1 : *C* **do**  3 *L*ic = log ；  4  **for** *j* = 1 : *D* **do**  5  **if** *xij* = 1 **then** *Lic* := *Lic* + log ；  6 **else** *L*ic := *L*ic + log ；  7 *pic*=exp(*L*ic-logsumexp(*Li*,:))；  8 =argmaxc*p*ic ； |

3.5.4 使用互信息进行特征选择

考虑到NBC模型对潜在的很多特征的联合概率进行建模，它可能遇到过拟合的问题。除此以外，它的时间复杂度为*O*(*ND*)，对于某些应用来说可能是比较高的。

一种解决上述两个问题的常见方法是进行**特征选择**（feature selection），将那些对分类问题帮助并不大的无关特征去除。特征选择的最简单方法是对每个特征的相关性进行单独的评估，然后选择前*K*个相关度最高的特征，其中*K*的选择是基于对精度和复杂度的一种权衡。这种方式被称为变量**排名**（ranking），**过滤**（filtering）或者**筛选**（screening）。

一种衡量相关性的方式是使用特征*Xj*和类标签*Y*之间的互信息：

 （3.75）

互信息可以看作是由于观察到了特征*j*而导致的标签分布的熵的减少。如果特征是二元的，很容易得到互信息的形式为（**读者可以证明下式**）：

 （3.76）

其中，所有这些量都可以在我们训练NBC模型过程中顺便得到。

表3.1 展示了将上述方法应用在图3.8所示的二元词袋法数据集的结果，不难发现，拥有最高互信息的特征与那些发生概率最高的特征相比，更具备可判别性。比方说，出现频率最高的单词“subject”在两个类别中都有出现，它之所以总是出现是因为这个数据集是一个新闻类的数据，而每个新闻都有一个主题（“subject”）。但显然这个单词不具备可判别性。与类别之间拥有最高互信息的单词分别是“windows”，“microsoft”，“dos”和“motif”，这是在情理之中的，因为实际数据中的两个类别分别对应Microsoft Windows和X Windows。

|  | **class1** | **prob** | **class2** | **prob** | **highest MI** | **MI** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | subject | 0.997788 | subject | 0.997788 | windows | 0.149046 |
| **1** | this | 0.639381 | windows | 0.639381 | microsoft | 0.066171 |
| **2** | with | 0.539823 | this | 0.539823 | dos | 0.063858 |
| **3** | but | 0.537611 | with | 0.537611 | motif | 0.054201 |
| **4** | you | 0.517699 | but | 0.517699 | window | 0.046677 |

表3.1 在类1（X windows）和类2（MS windows）中最有可能出现的5个单词。以及与类标签的互信息最高的5个单词。表格由程序**naiveBayesBowDemo**生成。

3.5.5 使用词袋法对文本进行分类（**注：本节更多技术细节请读者参考其他文献**）

**文本分类**（Document classification）是指将文本分类到不同的类别中。一种简单的方式是将每个文本表示为二元向量，向量中的分量记录着每个单词在文本中是否出现，所以*xij*=1的充要条件是单词*j*在文本*i*中出现，否则*xij*=0。然后我们可以使用下面的类条件密度：

 （3.77）

上式被称为**伯努利乘积模型**（Bernoulli product model）或者叫**二元独立模型**（binary independence model）。

然而，上述模型忽略了每个单词在文本中出现的次数。一个更加精确的方式是将每个单词在文本中出现的次数进行考虑。特别的，令为文本*i*的表示向量，其分量代表每个单词在文本中出现的次数，所以，其中*N*i表示文本*i*中的单词数量（所以）。对于类条件密度，我们可以使用多项式分布：

 （3.78）

上式中我们含蓄地表达了文本*i*的长度*N*i与类别无关。其中表示在类*c*中单词*j*出现的概率；这些参数满足约束。

尽管多项式分类器很容易训练并且在测试时也很简单，但对于文本分类问题效果并不是很好。原因之一在于它并没有考虑单词使用过程中的**突发特性**（burstiness）。所谓突发特性是指：大部分单词在任何给定的文本（比如训练样本）中从未出现，但是一旦它们在新的文本中出现过一次，它们很有可能会出现多次。

多项式模型不能适应上述的突发性现象。为了说明原因，注意式3.78具备（此处的*Nij*相当于式中的*xij*）的形式，因为对于那些罕见单词（如前文所述，在训练样本中这些单词很少出现，导致），如果基于这个参数估计值，那么我们将会相信在类*c*中不可能出现很多罕见单词（这与前文中的突发特性相悖）。对于那些出现频次高的单词（较大），这种衰减的速率不会太快。为了直观上明白这里面的原因，注意那些出现频次特别高的单词往往是那些功能单词，比如“and”，“the”，和“but”等等，这些单词与文本的类别没有关系。单词“and”出现的概率基本上保持不变，无论它之前出现的概率如何，所以独立性假设对于这些常用单词来说更加合理。然而，因为那些罕见单词对于我们的分类目的往往影响更大，所以需要我们更加小心的对待。

各种特别的启发式方法已经应用在多项式文本分类器中以提高性能。接下来，我们介绍一种类条件概率密度，它的性能与那些特别的方式相近，然而更具备概率性解释。

假设我们简单地将多项式类条件概率密度替换为**狄利克雷复合多项式**（Dirichlet Compound Multinomial，DCM）密度，定义如下：

 （3.79）

（上式由式5.24导出。）令人惊讶的是这个简单的变化就可以使模型适应突发特性现象。直观的解释是：当我们发现单词*j*出现过一次时，的后验计数将被更新，使得单词*j*再次出现的可能性增加。相反，如果固定，每个单词的出现是独立的。多项式模型对应于从一个容器中拿出一个球，记录下它的颜色后再放回。相反，DCM模型对应于拿出一个球，记录下颜色，放回的同时再复制一个相同的球，这被称为**Polya urn（译者注：**波利亚坛子模型(Polya's urn scheme)是一个著名的概率模型：假设坛子中装有*N*个球，其中有*N*1个黑球、*N*2个白球。任意取出一个，记下其颜色，并且在下次取球之前把该球连同另外r个与它同色的球一起放人坛中，再从坛中取出一个球，如此以往**）**。

使用DCM作为类条件密度得到的性能比多项式好很多，并且与那些先进的方法相比，其性能并不逊色。DCM模型的唯一缺点是其模型的训练过程更加复杂。